UNIVERSITY OF SOUTHAMPTON

# A Case Study on Modelling and Analysing Machine Breakdowns

by

Shu Pan

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Social and Human Sciences
Mathematical Sciences

March 2018

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF SOCIAL AND HUMAN SCIENCES
MATHEMATICAL SCIENCES

Doctor of Philosophy

**A Case Study on Modelling and Analysing Machine Breakdowns**

by Shu Pan

Most manufacturing models to date have assumed independence of all random variables in the system. In practice, autocorrelation effects are present in production lines time series. In this thesis, we extend this literature by studying autocorrelation in machine times to failure in detail. Our work focuses on the practical aspects of detecting and modelling autocorrelated uptimes, as well as including them in simulations.

We apply a practical procedure to detect autocorrelation in uptimes. The procedure has very mild assumptions and compensates for the number of machines it is applied to, ensuring that the probability of a Type I error is kept low.

We then provide two ways to model autocorrelated times to failures. The first is to use ARMA models including GARCH terms. We also provide a method based on the Markov-Modulated Poisson Process, a special case of the Markov Arrival Process.

For both methods discussed above, we provide diagnostic plots and a quantitative way to select the most appropriate model for a given series of uptimes. This allows us to automatically select an appropriate model.

Finally, to enable Ford to use our methods in simulation, we provide a way to generate simulated uptimes from each of our models.

# Contents

# List of Figures

# List of Tables

# Declaration of Authorship

I, Shu Pan, declare that this thesis entitled "A Case Study on Modelling and Analysing Machine Breakdowns" and the work presented in it are my own. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With this exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- none of this work has been published before submission.

Signed:

Date:

# Acknowledgements

My thanks go to Remco, my partner, for the happiness and support he has brought to me.

I would like to thank my father and mother for their kind and generous support over these years.

I am grateful for all the help I have received from the lecturers in the University of Southampton.

I would acknowledge my friends and other office mates in maths for their company over the years.

# Chapter 1

# Introduction

## 1.1 Motivation

Ford Motor Company builds and operates production lines that manufacture its cars. To build cost-effective production lines, the company needs to test various designs of a proposed production line in simulation. For these simulations to be accurate, a number of aspects of the production line must be reflected in the simulation, for instance:

1. The time a machine can operate before breaking down
2. The time it takes to repair a machine
3. The time it takes a machine to produce a part

Ford have studied each of the aspects above. While their model is mostly satisfactory, one of their primary current concerns is the first point, i.e. accurately modelling the time between breakdowns for each machine in the production line.

Currently, Ford model the times between breakdowns as independent and identically distributed (iid) random variables, by sampling from the empirical distribution. Their main concern is that the independence assumption may not be valid. In this thesis, we discuss how Ford may model the uptimes of their machines without assuming they are iid.

## 1.2 Related work

In this section, we review related work in modelling manufacturing systems, the impact of autocorrelation on manufacturing systems and ways that autocorrelation may be modelled.

### 1.2.1 Modelling Manufacturing Systems

A great number of literature has been generated in the area of modeling manufacturing systems since the early 1950's. A representative work in this area is Buzacott and Hanifin[25], in which the authors introduce the definition of operation dependent failures and time dependent failures. It also distinguishes the concepts between single station failures and total line failures. Moreover, Buzacott and Hanifin[25] provide a simple way to calculate the production rate under both time dependent and operation dependent failures. The production line has no buffer in their study.

Due to the great deal of literature in this area, there are a number of review papers. For instance, Perros[103] focuses on queuing networks with blocking and reviews a large number of relevant papers on this area. Smunt and Perkins[124] are particularly interested in asynchronous flow lines with reliable machines. They provide a review for many simulation papers in this area. A systematic description of the models, performance measures and methods in this field can be found in Guen[53]. Other surveys that emphasise the recent approaches and results include Perros[105], Perros[102], Onvural[96], and Dallery and Gershwin[36].

The models and results in the research of manufacturing systems have different features. Some of the important features that distinguish between models are discussed below:

**Synchronous and asynchronous systems** The major class of models are divided into asynchronous and synchronous models. In an asynchronous model, machines do not have to start or stop processing at the same instant. The presence of buffers between machines allows the machines to start and stop independently. Most real systems are asynchronous systems, and asynchronous models form an important class in the literature. It is usually assumed that the operation times in an asynchronous model are exponentially distributed.

In a synchronous system, machines have deterministic and equal processing times. Moreover, the machines start and stop at the same instant when they are not under repair. For further details about the asynchronous and synchronous models, see Buzacott[24][26].

**Reliable and unreliable machines** The machines in the production line can be either reliable or unreliable. However, in some cases, an unreliable machine can be modelled as a reliable machine. Therefore, methods designed to model reliable machines may also be used to model unreliable machines. Gaver[45] introduces the definition of completion time for an unreliable machine, which is the time difference between the instants of beginning and completion of the processing of a part. As such, the time includes the actual processing time of a part and the repair times corresponding to the failures that have occurred during the processing of the part.

The idea of transforming an unreliable machine into a reliable machine is to replace the completion time of an unreliable machine by the processing time of a reliable machine. The processing time of the reliable machine should be random and identical to the completion time of the unreliable machine. Further discussion about the reliable and unreliable machines can be found in Gaver[45], Bobbio and Trivedi[18], Sastry and Awate[117], Liu and Buzacott[80], and Altiok[3].

**Features of flow lines** The major features of a flow line include blocking, starving, processing, failures, and repairs. Some of the important features are discussed below.

- When the production line have limited buffers, blocking may occur. The definitions for different types of blocking were first introduced by Altiok and Stidham[8]. As discussed in their work[8], two types of blocking mechanisms are of interest in the manufacturing system: blocking-afer-service(BAS) and blocking-before-service(BBS). In the BAS blocking[97][54], a finished part stays on the working station until the downstream buffer has space. In a system with BBS blocking, a machine cannot start processing a part if there is no space available in the downstream buffer. Most production lines operate under the BAS mechanism, and hence that is what most authors assume.
- Processing time is the time required for a machine to complete a part on a working station. It is also referred to as operation time or cycle time in the literature. Usually, processing times at one machine and cross working stations are assumed to be independent. The commonly used distributions in modelling processing times are exponential distributions, geometric distributions, Coxian distributions and phase-type distributions[68][92].
- In a system with unreliable machines, the machines are prone to failures. When a failure happens, the machine is unable for processing parts until it is repaired. Most authors assume the repair process of each machine is independent of each other. Therefore, like the processing times, the time to repair is commonly assumed to be exponential, geometric, Coxian and phase-type distributions.
- We say a machine is operational if it is up; a machine is working if it is operational and not idle (neither blocked nor starved). Buzacott and Hanifin[25] introduce the idea of two major types of failures: operation dependent failures(ODF) and time dependent failures(TDF). ODF can only occur when the machine is working, and it is mainly caused by mechanical faults. TDF can occur at any time, and it is mainly due to failures of electronic systems.

**Exact and inexact methods** Exact methods and approximate methods are used to model the production line to obtain the performance measures. The exact methods are more appropriate to model small production systems, such as a two-station manufacturing line. Most of the results obtained using the exact analysis are based on Markovian analysis. To describe the production line by a Markov process, the distributions are required to be exponential or geometric distributions. The exact solutions for small production lines are available for a wide range of models satisfying these assumptions. The related study can be found in Muth[89][90], Stewart[125], Philippe et al.[106], Rao[109][110][111], Muth and Alkaff[91] and Stewart[126].

For large flow lines with more machines, it is very difficult to obtain the exact solution, even when more powerful computers are available. As Dallery et al.[36] discussed, the attempt to extend the analytic solution of a two-machine flow line model to three machines always ends up with models that are not tractable, or are subject to numerical problems, or too limited to be of interest. Therefore, approximate methods are an alternative to model flow lines with large number of machines. Most approximate methods are based on decomposition, which is to decompose the original system with $K$ working station into a set of $K - 1$ smaller subsystems which are easier to analyze. Each subsystem is associated with a buffer of the original line. The early work of decomposition methods include Sevast'yanov[120], and Hillier and Boling[59], which devise the principles of decomposition methods. Related work using decomposition methods to model flow lines can be found in Altiok and Ranjan[7], Dallery and Frein[35], Altiok[2], Perros and Altiok[104], Pollock et al.[107], Takahashi et al.[128].

Other authors have derived approximate formulas for estimating the production rate of a production line. Knott[70][71] develops a formula to calculate the production rate of a two-machine flow line with identical Erlang distributions. Haydon[56] extends Knott's formula to measure manufacturing lines with any number of machines. Muth and Alkaff[91] develop a formula to measure the efficiency of a flow line with any number of machines and no intermediate buffer. Blumenfield[17] extends Muth and Alkaff's formula to include intermediate storage between working stations. The other related work on measuring the production rate of a multi-stage flow line include: Van Dijk and Lamond[130], Baskett et al.[11], and Shanthikumar and Jafari[122].

Mathematical models that can obtain the exact solutions to answer the questions of interests are suitable for relatively small and simple systems. In real life, however, most manufacturing systems are too complex to allow realistic models to be evaluated mathematically[75]. Therefore, simulation modelling is widely used in modelling large systems. In fact, One of the largest application areas for simulation modeling is that of manufacturing systems[77]. Law[77] addresses some of the specific issues that simulation can deal with in manufacturing systems, such as production scheduling, inventory policies, and reliability analysis. Detailed discussions of how simulation is used to design and analyze manufacturing or warehousing systems can be found in Carson et al.[29] and Law and Kelton[74]. In particular, Law and Kelton[74] study the statistical issues in input simulation, and provide numerical examples to address the importance of correct input modelling for manufacturing systems. Law and Kelton[74] also discuss various sources of randomness in manufacturing systems. For example, inter-arrival times of jobs to a machine, processing times of jobs at a machine, machine working times before breakdown, machine repair times, and the outcomes of inspecting jobs. In later studies, Law and McComas[76] provide a practical discussion of the steps in simulation studies.

The discussions above have shown how broad the literature on modelling manufacturing systems is. For the purposes of our project, we only focus on the work that has similar features to Ford's production line. The features of interests for modelling and analysing a production line are listed below. They are also depicted in Figure 1.1, where only two workstations are presented.

- The flow line consists of parallel-machine workstations.
- The machines are unreliable and each workstation consists of a number of identical machines.
- There is finite intermediate buffer between workstations.
- The type of blocking mechanism is blocking-after-service(BAS). That is, when a machine at the first workstation completes a part, it checks the availability of space in the buffer. If the buffer is not full, it places the finished part onto the buffer and loads a new part immediately from upstream; if the buffer is full, the finished part stays in the working area of the machine of the first workstation and causes blockage of that machine. Similarly, if a machine at the last workstation completes a part, it sends the finished part immediately out of the system and loads a part from the buffer, provided the buffer is not empty. Otherwise, the machine in the last workstation is starved.
- All machines can only fail when processing a part. That is, the machines have operation dependent failures.
- There is unlimited supply of parts upstream of the first workstation; there is unlimited space downstream of the last workstation. In other words, the first workstation is never starved, and the last workstation is never blocked.

The following literature contains some or all of the features of interests. Buzacott[27] studies a two-workstation production line with identical machines and a finite intermediate buffer. In his model, the processing times and repair times are assumed to be exponentially distributed. The probability of failure during each operation is assumed to be constant. Buzacott[27] obtains an exact solution for the production capacity using a continuous time Markov chain model.

Gershwin and Berman[48] study a two-machine tandem flow line with finite buffer size. They make the assumptions that the processing times, times to failure, and times to

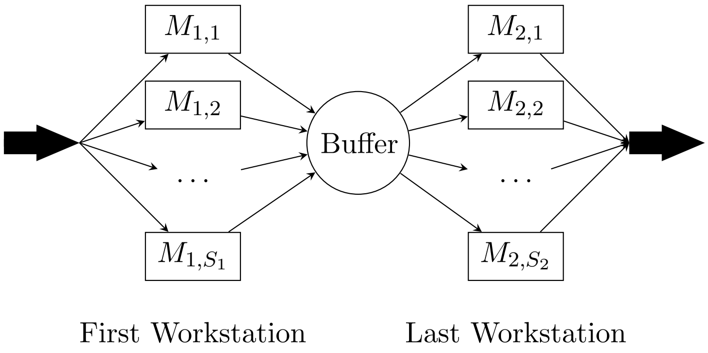


FIGURE 1.1: A two-workstation parallel-machine system with finite buffer

repair are all exponentially distributed. By calculating the steady state probabilities of the machines being in the operational and under repair states, they obtain the efficiencies of the flow line, such as the production rate and the average in-process inventory.

Berman[15] builds his model based on the work of Gershwin and Berman[48]. In his model, the processing time is allowed to have Erlang distributions, whereas the failure and repair times are assumed to be exponentially distributed. Both Berman[15] and Gershwin and Berman[48] assume blocking-before-service(BBS), which is not the case in our project. Altiok and Ranjan[7] analysis a two-node transfer line case where they assume the process completion times are a mixture of generalized Erlang distributions. Other studies related to modelling two-machine transfer lines with limited buffer in between can be found in Sastry[116], Artamonov[9], and Yeralan and Muth[134].

For longer production lines, Buzacott[24] extends his two-machine system to a three-machine production line with geometric times to failure and equal deterministic times to repair. Gershwin and Schick[49] model a buffered flow line with three unreliable machines under the assumptions that the machines have geometrically distributed times between failures and times to repair. They also notice the difficulty in obtaining exact solutions when extend a two-machine system to a three-machine system. Another author who attempts to expand the two-machine system to a three-machine system is Wiley[131]. Dallery and Gershwin[36] conclude that any asynchronous production line can be modelled as a discrete space continuous time Markov process, if all the distributions are given under phase-type forms. Similar assumptions in the manufacturing systems can be found in the work of Gershwin[46], Gershwin and Schor[50], Jafari and Shanthikumar[65], Dallery et al.[34], and Dogan-Sahiner and Altiok[40].

To summarize, most of the results in modelling manufacturing systems are based on the assumptions that all the random variables, such as the processing times, times to failure, and times to repair are independent and exponentially distributed. Therefore, the exact solutions can be obtained by modelling the systems as a Markov chain with continuous time and discrete states. For large production systems, exact solutions are not achievable. In order to tackle this problem, large systems are commonly decomposed into a set of small and easy to analyze systems. Therefore, the results of large systems actually rely on the exact solutions of the individual smaller systems.

In practice, the assumptions of independent and identically distributed variables may not hold. In these cases, the tools discussed above are not applicable and different tools are needed. The next subsection argues that real systems are impacted by autocorrelation.

### 1.2.2 The Impact of Autocorrelation on Manufacturing Systems

A vast majority of manufacturing-related models make independence assumptions on associated stochastic processes. These assumptions often lead to models that are easy to simulate and analytically tractable. For example, job inter-arrival times, service demands, failure times can each be modelled as a renewal process(see Kelly[67]).

The existence of "bursty" behavior, which is mainly underpinned by autocorrelations in random streams[5], has raised the question of whether the uncritical use of independence assumptions in modelling is appropriate. The term burstiness has been used to describe clustering phenomena in a traffic process[44]. For example, the occurrence of machine failures in short inter-arrival times followed by a relatively long one at a particular workstation is a phenomenon of burstiness. Moreover, burstiness of flows in broadband networks is common in the teletraffic community[14][69][78][98][66]. The study of burstiness in the discipline of telecommunication can be also be found in Leland et al.[79], Crovella and Bestavros[33], Salvador et al.[115], and Paxson and Floyd[99].

Various studies have found that models that do not incorporate dependencies among random variables can result in inaccurate performance predictions. Livny et al.[81] present a simulation study to show the influence of autucorrelation on a single-queue single-server system. Their simulation results show that autocorrelation in the inter-arrival and service times may severely affect the performance in an FIFO(First-In, First-Out) queue. They conclude that models use independence assumptions can lead to poor estimates of performance measures. In particular, they found that positive autocorrelations always lead to higher mean waiting times. The effect of negative correlations, however, is not clear in their study due to the complicity of the autocorrelation function.

Altiok and Melamed[5] demonstrate the profound impact that autocorrelations can have on the performance measures of a manufacturing system. Specifically, their study compares a number of performance measures with renewal components to their autocorrealted counterparts. The authors argue that machine failure is an important factor that directly affect the efficiency of producing processes. Although time to failure is traditionally assumed to be exponentially or geometrically distributed (see Altiok[4], Buzacott and Shanthikumar[28], and Gershwin[47]) in performance evaluation models, it is possible that machine failures exhibit autocorrelations.

Altiok and Melamed[5] also illustrate the impact of autocorrelation on the performance measures, such as throughput, system time, and the number of jobs in the system, using simulation case studies. Their results show that ignoring the burstiness can result in unacceptable prediction errors and overly optimistic performance measures. The authors then conclude that it is important to identify random components that posses correlations, and weigh the tradeoff between simplified assumptions and the accuracy of predictions carefully.

Hendricks and McClain[58] use simulation to examine the output processes of a serial production line. They find that negative autocorrelation is present in the output processes. They have also found that negative autocorrelation structure tends to reduce the variability of the output process. Therefore, small buffer sizes are sufficient for the production line with even highly variable machines.

Takahashi and Nakamura[127] study the effect of autocorrelated demands on the performance of a Kanban system. They conclude that the autocorrelated demands is an important factor affecting the performance of the system. Other works showing the existence of autocorrelation in the data from industrial plants can be found in Luxhoj and Shyur[86], Melamed and Hill[87], Mertens et al.[88], Schömig and Mittler[118], and Young and Winistorfer[135].

Nielsen[93] casts doubts on Livny et al.[81] and Altiok and Melamed[5]'s study. His results show that the impact of autocorrelation may not be as profound as the other authors have claimed. The reason is that other effects, such as the circumstances in manufacturing plants may reduce its importance.

Pereira et al.[100] present simulation experiments to show how autocorrelation can affect the performance of manufacturing systems. They conclude that process performance is improved by negative autocorrelation effects, and reduced by positive autocorrelation effects.

Since autocorrelation effects are present in manufacturing systems, we will discuss how to model the processes with autocorrelation in the next subsection.

### 1.2.3 Modelling autocorrelated processes

A number of methods to model autocorrelated processes have been used in the literature. Uhlig et al.[129] provide a recent review of a number of models and tools. We will now briefly discuss the various areas mentioned in [129], and some of their applications.

First, a number of authors have used processes based on time series analysis, such as autoregressive models. Pereira et al.[101] use an autoregressive model for the cycle time of machines in a production line. Autoregressive to anything (ARTA) models are discussed by Nielsen[94], and used in [31]. Underlying both autoregressive and

ARTA models is an autoregressive process, as defined by (1.1). Here, $Y_t$ is the series being modelled, $c$ and the $\alpha_i$ are constants, and $\epsilon_t$ is a normally distributed error term. In an autogressive model, $Y_t$ is observed directly. In the ARTA case, $X_t$, defined by (1.2) is observed. The function $F$ is the cumulative distribution function of the desired distribution, $\Phi$ is the cumulative density of the normal distribution. The idea of (1.2) is to transform $X_i$ to Gaussian, as normality is a very important assumption in many statistical techniques. The advantage of ARTA is that it allows to represent stationary time series with arbitrary marginal distributions. Other techniques to transform data into normality include Box-Cox Transformation[20].

$$Y_t = c + \sum_{i=1}^{p} \alpha_i Y_{t-i} + \epsilon_t \tag{1.1}$$

$$X_t = F^{-1}(\Phi(Y_t)) \tag{1.2}$$

Other authors have considered a variety of arrival processes based on Markov processes. Many of these are based on Markov Arrival Processes(MAP), introduced by Lucantoni et al.[85]. In the context of queuing theory, the MAP is considered, among others, by Lucantoni et al.[85] and Okazaki [95]. A special case of the MAP, the Markov-Modulated Poisson process, is considered by Heindl [57].

The Markov Arrival Process[85] is defined by an $m + 1$-state continuous time Markov chain, where state $m + 1$ is absorbing and the other states are transient. An arrival happens whenever the process moves to state $m + 1$, after which the process restarts from a randomly chosen state. Formally, let $\lambda_i$ be the rate at which the process leaves state $i$. When the process leaves state $i$, it moves to the absorbing state and restarts from state $j$ with probability $p_{ij}$. With probability $q_{ij}$, the process moves to state $j$, without passing through the absorbing state. Further details on this process and its properties are given in [85].

Another type of arrival process, the Transform Expand Sample process, is introduced by Livny et al.[81]. This process is considered by, among others, Altiok et al.[6] and Nielsen[94].

## 1.3 Contribution

As discussed in the previous section, most manufacturing models to date have assumed independence of all random variables in the system. In this thesis, we extend this literature by studying autocorrelation in machine times to failure(uptimes) in detail. Our work focuses on the practical aspects of detecting and modelling autocorrelated uptimes, as well as including them in simulations.

Our first contribution is a practical procedure to detect autocorrelation in uptimes. The procedure has very mild assumptions and compensates for the number of machines it is applied to. It also ensures that the probability of a Type I error, which is the incorrect rejection of a true null hypothesis(i.e., a “false alarm”), is kept low.

Second, we provide two ways to model autocorrelated uptimes. The first, an extension of the autoregressive and ARTA methods discussed previously, is to use ARMA models including GARCH terms. We also provide a method based on the Markov-Modulated Poisson Process, a special case of the Markov Arrival Process.

For both methods, we provide diagnostic plots and a quantitative way to select the most appropriate model for a given series of uptimes. This allows us to automatically select an appropriate model.

Finally, to enable Ford to use our methods in simulation, we provide a way to generate simulated uptimes from each of our models.

## 1.4 Structure of the thesis

In the next chapter, we consider the data Ford have available about one of their production lines and how breakdown and repair times for the machines may be calculated from this data. We also consider how to handle outliers in the data.

Thereafter, we consider how to detect whether the breakdown times for the machines are correlated with each other over time. We apply a suitable statistical test to each machine and combine the results appropriately to adjust for performing many tests.

In the two chapters after that, we apply two different approaches to modelling the breakdown times of a single machine. First, we consider using ARMA models from time series analysis. We also extend these models with GARCH terms, allowing the variance of the uptimes to change over time.

Second, we view the breakdown times of a machine as a series of arrivals, by considering each breakdown as an arrival. This leads us to apply the Markov-Modulated Poisson Process (MMPP) to the breakdown times.

For both ARMA models and the MMPP, we provide diagnostic plots that give a heuristic measure of the absolute quality of fit. We also compare the models objectively using the Akaike Information Criterion (AIC).

The thesis concludes with a brief summary of the conclusions and contributions of each chapter.

# Chapter 2

# Calculating up- and downtimes

## 2.1 Introduction

In this chapter, we present a procedure to calculate the up- and downtimes of Ford's machines based on the data collected by Ford. The data consists of a number of records, each of which indicates the start and end time of a machine failure, as well as the reason for the failure.

We begin the chapter by discussing the concepts of up- and downtimes, as well as the possible kinds of machine failure, in more detail. Then, we consider how Ford's data was collected and discuss a number of anomalies in the data.

We then consider how the data can be corrected to account for non-productive periods, during which the production line is not operational. This is particularly complex as the production line may occasionally be partially or fully operational during a nonproductive period.

Once we have obtained the up- and downtimes, we note that there are two important kinds of outlier in the data. First, some uptimes are very short, on the order of a few seconds. Secondly, some uptimes are abnormally long. We discuss how these outliers may be removed and consider the impact of doing this on further analysis.

The chapter concludes with a short summary of the data preparation procedure and its underlying assumptions.

## 2.2 Background

Before measuring the up times and down times of a machine, it is necessary to know the two major types of failures that have been considered in the literature. The first type of failure is operation dependent failures, and the second type of failure is time dependent failures.

Operation dependent failures are failures that are related to the processing of parts. Therefore, they can only occur when the machine is operational and not idle (neither starved nor blocked), and cannot happen when the station is forced down. According to Dallery and Gershwin[36], operation dependent failures are mainly due to mechanical causes, such as tool wear.

On the other hand, time dependent failures are not related to the processing of parts. They can occur at any time even when the station is forced down. Time dependent failures are mainly due to failures of electronic systems, and can occur after a certain amount of time due to some phenomenon other than wear [25].

Hanifin[55] observed 785 stoppages of one production line over a seven days period. Of the total time the line was down, 84% was due to operation dependent failures, and the rest 16% down time was caused by time dependent failures. It is also concluded by Buzacott and Hanifin[25] that in most production lines, most failures are operation dependent. As a result, most authors assume operation dependent failures [36].

Usually, the following information is collected in a data file: the machine that failed; the clock time at which the machine failed; the clock time at which the repair ended; the reason for the machine failure. As Buzacott and Hanifin[25] discussed, the assumption of operation dependent or time dependent failures of a station affects the way in which data on the operation of a transfer line is analysed.

The down time (or repair time) of a machine is the difference between the clock time at which the repair ended and the clock time at which the machine failed. The measurement of the down times is the same under the assumption of operation dependent failures and time dependent failures.

On the other hand, however, measuring the up times of a machine depends on whether the machine failures are assumed to be time dependent or operation dependent. For time dependent failures, an up time corresponds to the time between the end of the last repair and the instant of the next failure start. For operation dependent failures, an up time is the total working time between the end of the last repair and the instant of the next failure start. The working time corresponds to the time where the machine was processing parts, and does not include the time when the machine was forced down.

The engineers in Ford confirmed that the data set we are using captures only the failures on a machine, and a machine can only fail when it is operational and not idle. Therefore, we assume operation dependent failures when we analyse the data.

## 2.3 Data Collection at Ford

Ford's simulation team base their breakdown models on historical data. The data should record each failure and its corresponding reason for all the machines in the system. The recording of the start of a failure is important and it is necessary to know the relationship between a machine failure and a stoppage. Both of them can stop a machine from producing components. A machine failure or breakdown usually implies that the machine has stopped functioning the way in which it was intended or designed, whereas a machine can be stopped when it is being blocked, starved, or having a failure. According to Ford's engineers, a record in the data should be triggered by a failure, and not by a machine being blocked or starved.

The data is collected by manual observation and an automatic monitoring system [72]. Due to the cost and difficulty of manual data collection, electronic data collection is the main method used by Ford to collect machine breakdown data. The advantage of an automatic data collection process over manual data collection is that it records every failure. The disadvantages of the automatic data collection have been discussed in the work of Ladbrook[72], who is an engineer in Ford, and Lu[83], whose PhD thesis focused on modelling the down time of Ford's production line. The details of the disadvantages are discussed below:

- The monitoring system that records the failures may itself be unavailable:
  - The monitoring system at the plant may be shut down over the weekend. In addition to these planned outages, the system may also fail at any time. When the monitoring system is unavailable, no failures will be recorded, no matter how many machines fail. If a machine breaks down before the monitoring system goes down and is repaired while the monitoring system is down, this may cause the end of the repair process to go unrecorded.
- During a single machine breakdown, the system may record multiple breakdowns. This occurs for several reasons:
  - During a machine breakdown, the maintenance operator may perform "try outs" to test if the machine is repaired correctly. As the system cannot detect the "try outs" that happened within one failure, what was one breakdown could be recorded as two or more.
  - A similar situation to the one above is that if the machine is powered off during the repair process, the system may record two failures when there has only been one.
- The monitoring system does not account for non-productive periods:

- If a machine is broken down during the weekend, it is unknown whether the operators have chosen to work overtime to repair the machine. If the operators have chosen to work overtime, the completion of the repairs may have been recorded. However, the repair period may still be overstated, as the operators may not have worked all hours of the day. If the operators chose not to work overtime, the machine may be recorded as having been repaired early the following week. The repair time will therefore be overstated by including the weekend.
- There are shift breaks in each productive day, during which the operators are supposed to be non-productive, and not by the side of the line. However, since the automatic system cannot detect these shift breaks, a recorded repair duration may cover several non-productive time durations. This can make the repair time much longer than the real time an operator spent in fixing the machine.

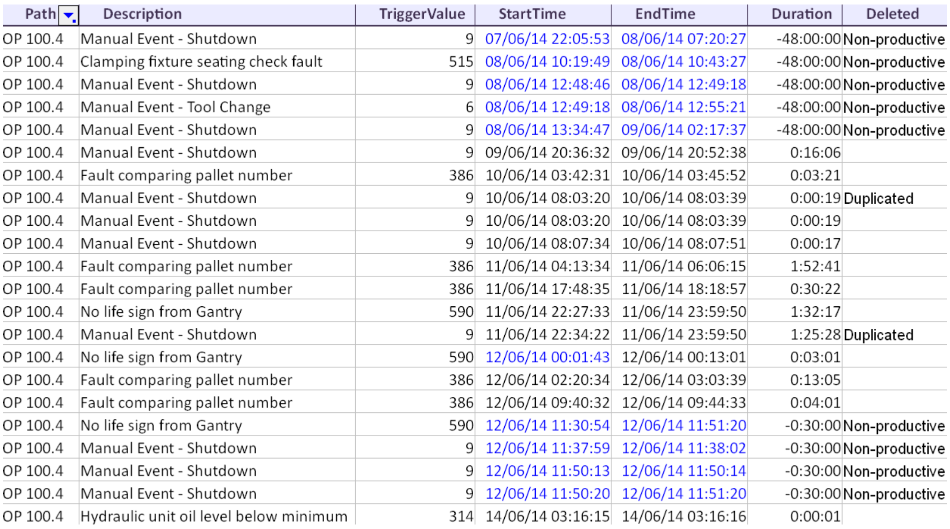
In addition to the disadvantages listed above, we have also found that the monitoring system may separately record further details of an individual failure. This leads to a series of records covering nested time periods.

This case study is based exclusively on the electronically gathered data, as it is the most extensive data set available at Ford. Due to the anomalies listed above, the data gathered by the monitoring system is not suitable for analysis without careful cleaning. The detailed data preparation process will be discussed in the remainder of this chapter.

## 2.4 Data Description

The data to be analyzed is from the line “Tiger Cylinder Block” in Dagenham engine plant of Ford. The range of the data is from March 2014 to March 2015. The original data collected by the automatic monitoring system contains a lot of information that is not relevant for the purpose of our study. Therefore, the engineers in Ford provided some semi-processed data files that contain the information crucial to our analysis. A screen shot of the semi-processed data is shown in Figure 2.1, in which the machine that failed, the failure description, the failure start and end times are recorded. Apart from that, the engineers in Ford added further information of the duration of a machine breakdown, and whether some data records should be deleted.

As seen from Figure 2.1, some data records have their “StartTime” and “EndTime” marked blue. The reason is that the event times of the machine failure or repair are within the non-productive periods of the plant. As an opposite to the fully operational times, the non-productive times are any time period during which the production line is



| Path | Description | TriggerValue | StartTime | EndTime | Duration | Deleted |
|---|---|---|---|---|---|---|
| OP 100.4 | Manual Event - Shutdown | 9 | 07/06/14 22:05:53 | 08/06/14 07:20:27 | -48:00:00 | Non-productive |
| OP 100.4 | Clamping fixture seating check fault | 515 | 08/06/14 10:19:49 | 08/06/14 10:43:27 | -48:00:00 | Non-productive |
| OP 100.4 | Manual Event - Shutdown | 9 | 08/06/14 12:48:46 | 08/06/14 12:49:18 | -48:00:00 | Non-productive |
| OP 100.4 | Manual Event - Tool Change | 6 | 08/06/14 12:49:18 | 08/06/14 12:55:21 | -48:00:00 | Non-productive |
| OP 100.4 | Manual Event - Shutdown | 9 | 08/06/14 13:34:47 | 09/06/14 02:17:37 | -48:00:00 | Non-productive |
| OP 100.4 | Manual Event - Shutdown | 9 | 09/06/14 20:36:32 | 09/06/14 20:52:38 | 0:16:06 | |
| OP 100.4 | Fault comparing pallet number | 386 | 10/06/14 03:42:31 | 10/06/14 03:45:52 | 0:03:21 | |
| OP 100.4 | Manual Event - Shutdown | 9 | 10/06/14 08:03:20 | 10/06/14 08:03:39 | 0:00:19 | Duplicated |
| OP 100.4 | Manual Event - Shutdown | 9 | 10/06/14 08:03:20 | 10/06/14 08:03:39 | 0:00:19 | |
| OP 100.4 | Manual Event - Shutdown | 9 | 10/06/14 08:07:34 | 10/06/14 08:07:51 | 0:00:17 | |
| OP 100.4 | Fault comparing pallet number | 386 | 11/06/14 04:13:34 | 11/06/14 06:06:15 | 1:52:41 | |
| OP 100.4 | Fault comparing pallet number | 386 | 11/06/14 17:48:35 | 11/06/14 18:18:57 | 0:30:22 | |
| OP 100.4 | No life sign from Gantry | 590 | 11/06/14 22:27:33 | 11/06/14 23:59:50 | 1:32:17 | |
| OP 100.4 | Manual Event - Shutdown | 9 | 11/06/14 22:34:22 | 11/06/14 23:59:50 | 1:25:28 | Duplicated |
| OP 100.4 | No life sign from Gantry | 590 | 12/06/14 00:01:43 | 12/06/14 00:13:01 | 0:03:01 | |
| OP 100.4 | Fault comparing pallet number | 386 | 12/06/14 02:20:34 | 12/06/14 03:03:39 | 0:13:05 | |
| OP 100.4 | Fault comparing pallet number | 386 | 12/06/14 09:40:32 | 12/06/14 09:44:33 | 0:04:01 | |
| OP 100.4 | No life sign from Gantry | 590 | 12/06/14 11:30:54 | 12/06/14 11:51:20 | -0:30:00 | Non-productive |
| OP 100.4 | Manual Event - Shutdown | 9 | 12/06/14 11:37:59 | 12/06/14 11:38:02 | -0:30:00 | Non-productive |
| OP 100.4 | Manual Event - Shutdown | 9 | 12/06/14 11:50:13 | 12/06/14 11:50:14 | -0:30:00 | Non-productive |
| OP 100.4 | Manual Event - Shutdown | 9 | 12/06/14 11:50:20 | 12/06/14 11:51:20 | -0:30:00 | Non-productive |
| OP 100.4 | Hydraulic unit oil level below minimum | 314 | 14/06/14 03:16:15 | 14/06/14 03:16:16 | 0:00:01 | |

FIGURE 2.1: A screenshot of the semi-analyzed data

shut down. Detailed discussions of the non-productive times of the production line are in section 2.6.

In the semi-processed data, if both of the failure start time and failure end time are within a non-productive period, the duration is marked with a negative number, and it is suggesed to delete the whole failure record from the data. This scenario is illustrated in Figure 2.2.

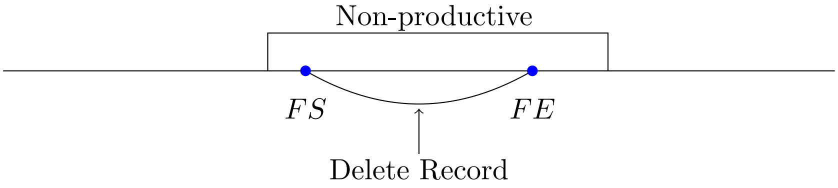


FIGURE 2.2: A failure starts and ends within a non-productive period

If the start time of a failure is within a non-productive period, and the failure ends during a fully operational time, then the duration of the failure does not include the time that is overlapped with the non-productive period. As shown in Figure 2.3, the repair duration is considered to start at the end of the non-productive time.

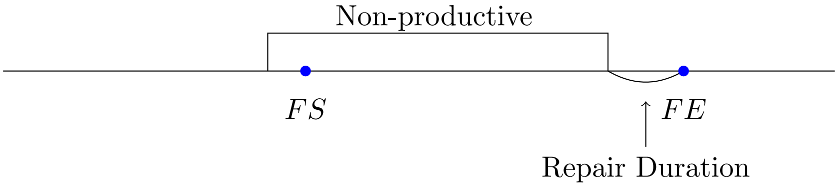


FIGURE 2.3: A failure starts within a non-productive period and ends outside a non-productive period

Similarly, if the end time of a failure is within a non-productive period, and the failure starts during a fully operational time, then the repair time is suggested to end at the time when the non-productive period starts. This is illustrated in Figure 2.4.

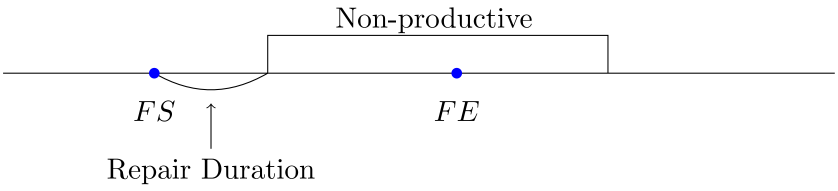


FIGURE 2.4: A failure starts during fully operational period and ends within a non-productive period

If both of the failure start and failure end are within fully operational times, but the repair duration covers a non-productive period, then that non-productive period will be subtracted from the repair duration. Figurer 2.5 illustrates this situation, in which the repair time of the failure is the sum of Repair Duration 1 and Repair Duration 2. This situation, however, has no special mark in the semi-processed data file. Therefore, it is not easy to distinguish this situation from most scenarios by eye, when a failure start and end during operational times, with no non-productive period in between.

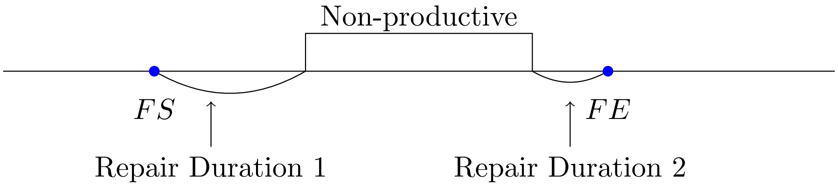


FIGURE 2.5: A failure duration covers a non-productive period

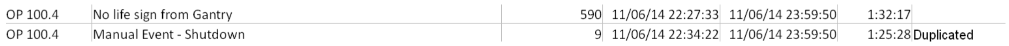
The purpose of our study is to model the times between breakdowns, i.e. the up times. The "Duration" column in the semi-processed data file indicates the repair times, i.e. the down times. In general, an up time corresponds to the time between the end of the last repair and the instant of the next failure start. As such, instead of the duration of down times, the exact times of when a machine failure starts and ends are more helpful for our study. However, like the down times, the non-productive periods need to be considered when calculating the up times. Therefore, the machine failure records that involve the non-productive periods should be adjusted carefully. This increases the complexity of handling the data and will be discussed further in Section 2.6.

## 2.5 Anomalies in the data

As discussed in section 2.3, there are a number of anomalies in the data that need to be carefully handled. In this section, we will discuss each type of anomaly in turn.

**Handling outages or nonproductive periods of the monitoring system** When the monitoring system is unavailable, failures may go unrecorded or the completion of repairs started previously may not be logged. In the former case, all record of the failure is lost, and we cannot recover the failure from the data. In the latter case, the data contains a record with a missing end time. As no further information was available to reconstruct the end time of these repairs, they were excluded from further analysis.

**Handling nested records** In some cases, the monitoring system logs failures in more detail than is required for this case study. This is illustrated in Figure 2.6, where the start time of the later repair, 22:34:22, is before the end time of the first repair, 23:59:50. For the purposes of this case study, all subsequent records providing further detail about the nature of the failure were discarded. In the semi-processed data file, the redundant machine failure records are marked as "Duplicated".

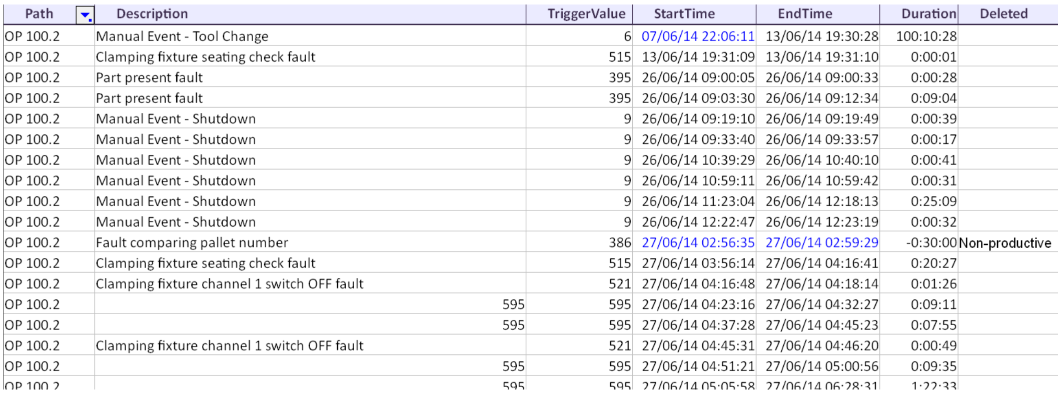


| | | | | | | | |
|---|---|---|---|---|---|---|---|
| OP 100.4 | No life sign from Gantry | | 590 | 11/06/14 22:27:33 | 11/06/14 23:59:50 | 1:32:17 | |
| OP 100.4 | Manual Event - Shutdown | | 9 | 11/06/14 22:34:22 | 11/06/14 23:59:50 | 1:25:28 | Duplicated |

Figure 2.6: A screen shot of an example of nested records

**Handling non-production machines** Most of the machines have names starting with OP in the data files. However, some machines have very different names, such as DAG0160, P0598L, and TBSYS80. After consulting with the engineers in Ford, these are part of the supporting system, and are not production machines. For instance, machine DAG0160 refers to equipment described as "Tiger machining chilled water". Therefore, as suggested by the engineers, the failure records of such machines should be discarded from the analysis.

**Handling failure descriptions** The data files contain a column called "Description", which is supposed to describe the reasons of each machine failure. This may be helpful for further analyzing and understanding the autocorrelated times to failure. However, some of the entries in the column are empty and a lot of them have the description of "Manual Event - Shutdown", as illustrated in Figure 2.7. According to the engineers in Ford, "Manual Event - Shutdown" basically means the machine is broken down, but they do not have the mechanical reason for why. The reason is that an operator has manually taken the machine offline, which could be for any number of reasons, but the engineers consider it is usually to repair it. As the exact reason of each manual shutdown is not clear, we cannot use the information of failure description for our analysis at the moment.



| Path | Description | | TriggerValue | StartTime | EndTime | Duration | Deleted |
|---|---|---|---|---|---|---|---|
| OP 100.2 | Manual Event - Tool Change | | 6 | 07/06/14 22:06:11 | 13/06/14 19:30:28 | 100:10:28 | |
| OP 100.2 | Clamping fixture seating check fault | | 515 | 13/06/14 19:31:09 | 13/06/14 19:31:10 | 0:00:01 | |
| OP 100.2 | Part present fault | | 395 | 26/06/14 09:00:05 | 26/06/14 09:00:33 | 0:00:28 | |
| OP 100.2 | Part present fault | | 395 | 26/06/14 09:03:30 | 26/06/14 09:12:34 | 0:09:04 | |
| OP 100.2 | Manual Event - Shutdown | | 9 | 26/06/14 09:19:10 | 26/06/14 09:19:49 | 0:00:39 | |
| OP 100.2 | Manual Event - Shutdown | | 9 | 26/06/14 09:33:40 | 26/06/14 09:33:57 | 0:00:17 | |
| OP 100.2 | Manual Event - Shutdown | | 9 | 26/06/14 10:39:29 | 26/06/14 10:40:10 | 0:00:41 | |
| OP 100.2 | Manual Event - Shutdown | | 9 | 26/06/14 10:59:11 | 26/06/14 10:59:42 | 0:00:31 | |
| OP 100.2 | Manual Event - Shutdown | | 9 | 26/06/14 11:23:04 | 26/06/14 12:18:13 | 0:25:09 | |
| OP 100.2 | Manual Event - Shutdown | | 9 | 26/06/14 12:22:47 | 26/06/14 12:23:19 | 0:00:32 | |
| OP 100.2 | Fault comparing pallet number | | 386 | 27/06/14 02:56:35 | 27/06/14 02:59:29 | -0:30:00 | Non-productive |
| OP 100.2 | Clamping fixture seating check fault | | 515 | 27/06/14 03:56:14 | 27/06/14 04:16:41 | 0:20:27 | |
| OP 100.2 | Clamping fixture channel 1 switch OFF fault | | 521 | 27/06/14 04:16:48 | 27/06/14 04:18:14 | 0:01:26 | |
| OP 100.2 | | 595 | 595 | 27/06/14 04:23:16 | 27/06/14 04:32:27 | 0:09:11 | |
| OP 100.2 | | 595 | 595 | 27/06/14 04:37:28 | 27/06/14 04:45:23 | 0:07:55 | |
| OP 100.2 | Clamping fixture channel 1 switch OFF fault | | 521 | 27/06/14 04:45:31 | 27/06/14 04:46:20 | 0:00:49 | |
| OP 100.2 | | 595 | 595 | 27/06/14 04:51:21 | 27/06/14 05:00:56 | 0:09:35 | |
| OP 100.2 | | 595 | 595 | 27/06/14 05:05:58 | 27/06/14 06:28:31 | 1:22:33 | |

Figure 2.7: A screen shot of an example of machine failure descriptions

## 2.6 Non-productive Time of the production line

Ford's engineers define the non-productive time as any time period during which the production line is shut down. During the non-productive times, the operators are not supposed to be by the side of the line. Therefore, no operation or repair can be done in these periods. As an opposite to the fully operational times, the non-productive times in Ford usually include weekends, shift breaks, bank holidays, summer breaks, and Christmas breaks.

Occasionally, the plant ran additional (overtime) shifts during non-productive times. This can be seen from the machine failure records captured by the electronic monitoring system, as some of the machines failed or were repaired during the non-productive times. The problem with this is that the operators do not necessarily publish when this occurred. More importantly, it is not known if they ran the whole line or just one section, or if they were fully manned.

As Ford aim to build a simulation model which only models the fully operational times, any breakdown record that involves the non-productive periods should be adjusted carefully. The detailed information of each type of non-productive time in Ford is listed below.

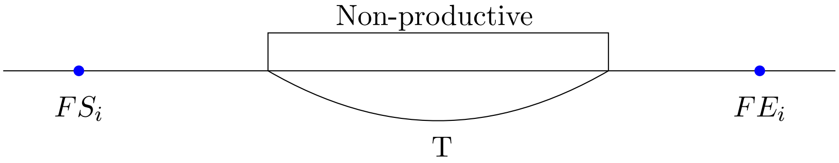
- Weekends: there are three shifts in a normal working day, which are from $6:00$ to $14:00$, from $14:00$ to $22:00$, and from $22:00$ to $6:00$ the next working day. The last production shift of the week ends at $6:00$ on Saturday. Therefore, the weekend time starts at $6:00$ on Saturday and ends at $6:00$ on Monday.
- Bank holidays: Similar to the weekends, the line is supposed to be shut down at $6:00$ on the start day of the bank holiday. The operators will resume their work at the first production shift after the bank holiday.

- Shift breaks: there are six shift breaks in 24 hours, which include the following periods: 02 : 30 to 03 : 00, 08 : 30 to 08 : 40, 11 : 30 to 12 : 00, 16 : 30 to 17 : 00, 18 : 30 to 18 : 40, 24 : 00 to 24 : 10. The total shift breaks duration per 24-hour is 2 hours.
- Summer breaks: According to the range of our data, the summer breaks started at 6 am on $28^{th}$ July, 2014 and ended at 6 am $15^{th}$ August, 2014.
- Christmas breaks: According to the range of our data, the Christmas breaks started at 6 am on $24^{th}$ December, 2014 and ended at 6 am $2^{nd}$ January, 2015.

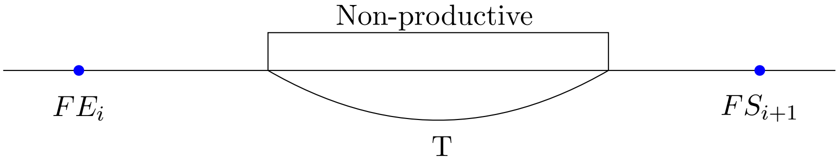
We will now consider how to calculate the up times and down times of Ford's machines, assuming operation-dependent failure. Let $FS_i$ and $FE_i$ be the start and end time of the $i^{th}$ machine failure. Therefore, $(FS_i, FE_i)$ corresponds to the $i^{th}$ down time duration and $(FE_i, FS_{i+1})$ corresponds to the $i^{th}$ up time duration. Since in the data file, a machine can fail and be repaired during a non-productive time period, let us define $T_j$ to be the $j^{th}$ period in a non-productive time that has overlapped with either a down time duration or an up time duration.

Let $N$ be the number of events happened during a non-productive time period. The events can be a failure start or a failure end. We will now consider how the up times and down times should be calculated in each case.

**Case 1. N = 0**

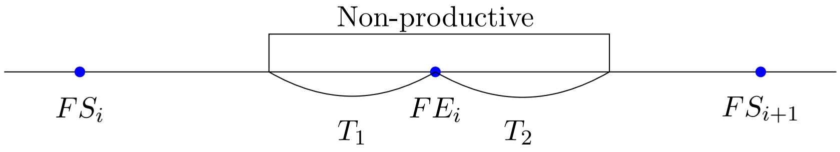


During the non-productive time, there is no operator repairing the machine. Therefore, the $i^{th}$ down time $D_i$ is calculated by $FE_i - FS_i - T$;

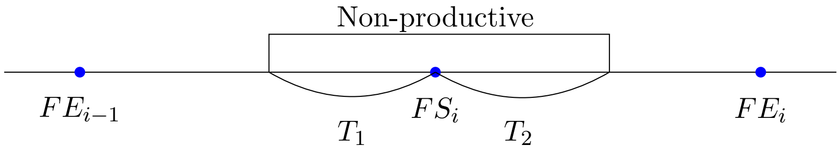


Similarly, during the non-productive time, the machine is not fully working. Therefore, the $i^{th}$ up time $U_i$ is calculated by $FS_{i+1} - FE_i - T$.

**Case 2. N = 1**

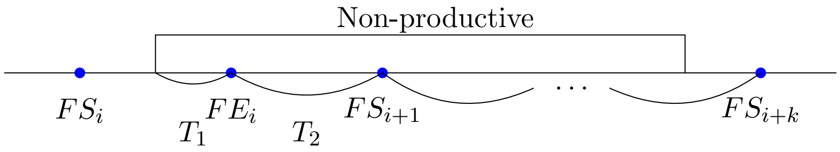


A failure was fixed during the non-productive time. This indicates that there are operators repairing the machine during $T_1$. Since we assume the operators were working full time during $T_1$, $D_i = FE_i - FS_i$. Further, we assume that after the machine was fixed at time $FE_i$, it was shut down immediately and switched on again after the non-productive time finished. Therefore, the $i^{th}$ up time $U_i$ is $FS_{i+1} - FE_i - T_2$.

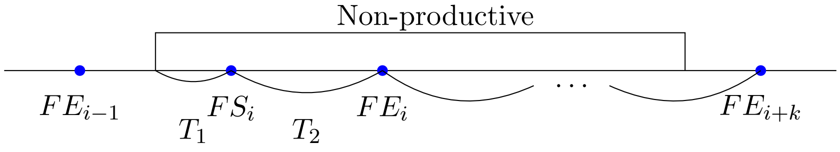


A failure started during the non-productive time. As we assume operation dependent failures, a machine can only fail when it is processing. Therefore, the machine should be operating during time period $T_1$. As such, the up time $U_{i-1}$ should be calculated by $FS_i - FE_{i-1}$. As we assume the operators chose to repair the machine after the non-productive time, so $D_i = FE_i - FS_i - T_2$.

**Case 3. N >= 2**



Similar to Case 2, $D_i = FE_i - FS_i$, $U_i = FS_{i+1} - FE_i$, $D_{i+1} = FE_{i+1} - FS_{i+1}$, $U_{i+1} = FS_{i+2} - FE_{i+1}$, etc. There is no subtraction of the non-productive times until the last event that happened during the non-productive time, which is $FE_{i+k-1}$. After $FE_{i+k-1}$, we assume the machine was shut down immediately and switched on after the non-productive time finished, so $U_{i+k-1} = FS_{i+k} - FE_{i+k-1} - T_{i+k+1}$.



Similarly, $U_{i-1} = FS_i - FE_{i-1}$, $D_i = FE_i - FS_i$, $U_{i+1} = FS_{i+2} - FE_{i+1}$, $D_{i+1} = FE_{i+1} - FS_{i+1}$, etc. There is no subtraction of the non-productive times until the last event that happened during the non-productive time, which is, $FS_{i+k}$. As we assume the operators started repairing the machine after the non-productive time, $D_{i+k} = FE_{i+k} - FS_{i+k} - T_{i+k+1}$.

## 2.7 Data Processing

In this section, we will discuss the details of how to process the data for further analysis. The processing was implemented using Python and will be illustrated using code snippets. The first step of processing the data is to parse the date and time given as DD/MM/YY HH:MM:SS to a time duration in seconds since 01/01/2014 00:00:00. This is to make the calculation of up and down times easier, and they will be measured in seconds. The Python code that illustrates how to do this is shown below.

```
def parse_datetime(strv, format = ’%d/%m/%y %H:%M:%S’):
    try:
        start = dt.datetime(2014, 1, 1)
        now = dt.datetime.strptime(strv, format)
        return (now - start).total_seconds()
    except BaseException:
        print ’Error on converting datetime:’, strv
        return -1
```

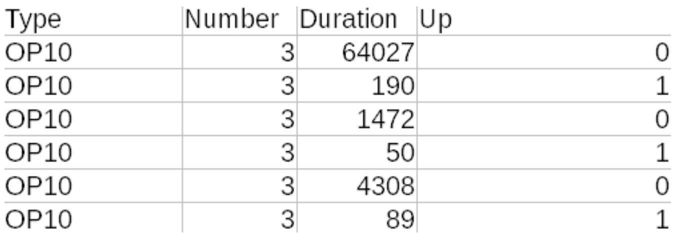
The next step is to calculate the up and down times from the machine failure records. As discussed in Section 2.6, a machine can fail and be repaired during a non-productive time period. Therefore, it is important to detect whether a failure start or failure end happens during a non-productive period. The Python code that illustrates this step can be seen below.

```
def loc_bisect(time, npros):
    i = bisect.bisect_right(npros, (time, 1e9))
    if i == 0:
        return (i-1, i)
    else:
        st, ed = npros[i-1]
        if ed < time:
            return (i-1, i)
        else:
            return (i-1, i-1)
```

After we know whether an event happens during a non-productive period or not, we can calculate the up and down time durations based on the logic discussed in Section 2.6. The code below illustrates how this can be achieved in Python.

```
def duration(start, end, npros):
    sloc = loc_bisect(start, npros)
    eloc = loc_bisect(end, npros)
    nps = []
    if sloc != eloc and sloc[0] == sloc[1]:
        start = npros[sloc[1]][1]
        nps = npros[sloc[1]+1:eloc[1]]
    elif sloc != eloc:
        nps = npros[sloc[1]:eloc[1]]
    subtraction = sum([e-s for (s,e) in nps])
    return end - start - subtraction
```

After calculating the up and down time durations, we output the processed information in a csv file with the format illustrated in Figure 2.8. The "Type" and "Number" columns represent the working station and machine number, respectively. For example, machine OP10.3 will be considered as "Type" OP10 and "Number" 3. The up and down time durations are measured with the time unit of seconds. In the "Up" column, values 0 and 1 are the indicator of down and up time durations, respectively. The down and up times are in order, with a down time followed by an up time.



| Type | Number | Duration | Up |
|---|---|---|---|
| OP10 | 3 | 64027 | 0 |
| OP10 | 3 | 190 | 1 |
| OP10 | 3 | 1472 | 0 |
| OP10 | 3 | 50 | 1 |
| OP10 | 3 | 4308 | 0 |
| OP10 | 3 | 89 | 1 |

FIGURE 2.8: Format of the processed data output in a csv file

## 2.8 Outliers in the uptimes

To conclude this chapter, we will consider the impact of the data anomalies and data preparation procedure on the quality of the calculated uptimes. In particular, we will be concerned with outliers in the uptimes.

First, consider the "try outs" mentioned in Section 2.3, where engineers start a machine to test whether it has been fixed. If the machine has not been fixed, a try out leads to a very short uptime, on the order of a few seconds. This suggests that only uptimes that are sufficiently long are valid. Based on their experience, Ford's engineers suggest removing uptimes shorter than 5 minutes.

Second, issues with the monitoring system may lead to overstating the length of an uptime. Consider a scenario illustrated in Figure 2.9. A machine starts running at time $T_1$, then breaks down at time $T_2$, is repaired at time $T_3$, and breaks down again at time $T_4$. If, due to a failure of the monitoring system, the end of the repair at time $T_3$ is not recorded, we have to remove this breakdown from the data. In this case, we record an uptime of $T_4 - T_1$, rather than the correct uptimes $T_2 - T_1$ and $T_4 - T_3$.

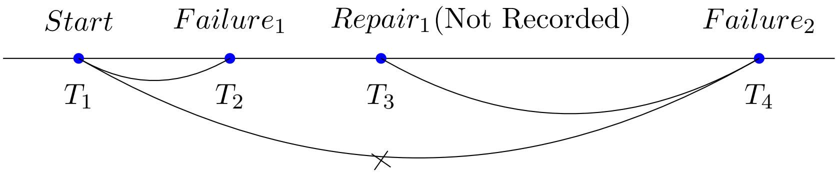


FIGURE 2.9: A scenario of miss recording of a repair time

Unfortunately, it is impossible to tell from the data what the correct uptimes should be. As such, we can only attempt to detect anomalously long uptimes and exclude them as outliers. In this thesis, we have chosen to consider every uptime longer than 1 week an outlier. The reason is that an up time longer than 1 week is exceedingly rare in our data. Moreover, any up time duration longer than 1 week is very far away from the other data points.

According to the data, there are 42 stations in the production line. Each station contains either one machine or parallel machines. For instance, machines OP70.1 and OP70.7 indicate they are from the same station OP70; while machine types OP30A and OP125 indicate that there is only one machine at the stations, respectively.

There are 84 individual machines in total, and the number of failure records in the data files is 293989. For each machine, the observations of machine failures in one year range from 991 to 17581. For most of the machines, the minimum uptimes are 1 or 2 seconds, while the maximum uptimes are quite different between machines. For instance, machine OP20.5 has the largest maximum uptime, which is about 355 hours. Despite from the same station, the maximum uptime for machine OP20.3 is about 51 hours. The smallest maximum uptime is obtained from machine OP130G, which is about 33 hours.

Table 2.1 and Table 2.2 illustrate the top 5 and lowest 5 percentage of data removed from the machines' failure records, respectively. As discussed before, any uptime that is less than 5 minutes or greater than 1 week will be deleted from the records. Of all the 84 machines in the production line, OP30A has the maximum percentage of deleted data, which is about 66.53%. Machine OP20.9 has the minimum percentage of deleted data among the machines, which is around 29.21%.

The majority of the data excluded are the uptimes that are less than 5 minutes. These small uptimes account for a large number of data records, and may have an impact on further data analysis. This will be discussed in detail later chapters. The uptimes that are larger than 1 week are relatively rare for each machine, and the impact of excluding these data may be small.

Table 2.1: Number of outliers of each type by machine, for top 5 deleted machines

| Machine | < 5 minutes | > 1 week | Total | Percentage of Removal |
|---|---|---|---|---|
| OP30A | 2826 | 1 | 4249 | 66.53% |
| OP10G | 3263 | 1 | 5255 | 63.30% |
| OP140 | 2547 | 1 | 40475 | 62.94% |
| OP170LEAK | 10997 | 1 | 17581 | 62.55% |
| OP160 | 5805 | 1 | 9309 | 62.36% |

Table 2.2: Number of outliers of each type by machine, for lowest 5 deleted machines

| Machine | < 5 minutes | > 1 week | Total | Percentage of Removal |
|---|---|---|---|---|
| OP80.3 | 1078 | 1 | 3303 | 32.64% |
| OP80.9 | 1071 | 1 | 3495 | 30.64% |
| OP20.11 | 1740 | 0 | 5725 | 30.39% |
| OP80.7 | 1128 | 1 | 3733 | 30.22% |
| OP20.9 | 636 | 1 | 2181 | 29.21% |

## 2.9 Conclusion

In this chapter, we have discussed the concepts of uptimes and downtimes, considered how Ford's data is collected and provided a way to calculate up- and downtimes from Ford's data. The method we have proposed handles various anomalies in the data and deals with nonproductive periods. However, due to limitations in the data collection process, the resulting uptimes may contain outliers. In the remainder of this thesis, we will consider how the uptimes may be modelled.

# Chapter 3

# Autocorrelation analysis

## 3.1 Introduction

In this chapter, we discuss how to detect whether the uptimes of Ford's machines are correlated with each other. We begin by reviewing how correlation within a time series is defined and how it may be estimated. We then use these concepts to review the Ljung-Box test, which allows us to test whether a given process is autocorrelated.

While we may apply the Ljung-Box test to determine whether the uptimes of a single machine are autocorrelated, repeating this procedure for all of Ford's machines results in us performing many hypothesis tests. This may lead us to make an unacceptable number of Type I errors, i.e. to detect autocorrelation in machines where there is none. This problem is known in the literature as *multiple hypothesis testing.* We review a number of approaches to this problem.

Finally, we apply the Ljung-Box test to Ford's data, using a Holm-Bonferroni correction to correct for multiple hypothesis testing. This procedure results in a list of machines that show autocorrelation.

## 3.2 Testing for autocorrelation

We will now consider how we may perform a statistical test to detect autocorrelation in the uptimes of a machine. First, we will review how the autocorrelation of a time series is defined and how it may be estimated. The discussion here is based on Box and Jenkins[22].

Let $X_1, X_2, \ldots$ be the random variables representing a given time series, and let $k$ be a positive integer. Then, the autocorrelation coefficient at lag $k$ is given by (3.1). The quantity $\rho_k$ represents the correlation within the time series at intervals of length $k$.

$$\rho_k = \frac{Cov(X_i, X_{i+k})}{Var(X_i)} \tag{3.1}$$

The right-hand side of (3.1) contains an index $i$. For a general time series, this quantity could depend on $i$, which implies that $\rho_k$ is not well-defined. To ensure that $\rho_k$ is well-defined, we assume the time series is *second-order stationary*. This means that the mean, variance and covariances of the time series do not depend on the index $i$, and so $\rho_k$ is well-defined.

Now suppose that we are given a finite sample $x_1, x_2, \ldots x_n$ from the time series. To estimate the autocorrelation, we can use the following estimator (3.2). Here, $a_i = x_i - \frac{1}{n}\sum_{j=1}^{n} x_j$.

$$\hat{\rho}_k = \frac{\sum_{l=k+1}^{n} a_l a_{l-k}}{\sum_{i=1}^{n} {a_i}^2} \tag{3.2}$$

Using the autocorrelation coefficient and its estimator defined above, the hypothesis “The uptimes of the machine are independent of each other” may be stated as “for all $k > 0$, $\rho_k = 0$”. Letting this hypothesis be our null hypothesis $H_0$, we can now consider the following test statistic [82]. Here, $\hat{\rho}_k$ is the estimator of $\rho_k$ given by (3.2).

$$Q = n(n+2)\sum_{k=1}^{m} \frac{\hat{\rho}_k^2}{n-k} \tag{3.3}$$

In (3.3), $n$ is length of the time series $x_1, \ldots, x_n$ and $m$ is a specified maximum lag to consider. The statistic $Q$ has an approximate $\chi^2_m$ distribution under $H_0$, provided that the variance of $X_i$ is finite.

## 3.3 Correcting for multiple hypothesis tests

To analyze the autocorrelation in the time-to-breakdown, we need to perform hypothesis tests for individual machines. The null hypothesis under the test is that there is no autocorrelation in the time-to-breakdown for each machine.

As we perform more than one statistical test, the chance that we reject the null hypothesis when it is true is larger. This is because when many hypotheses are tested, and each test has a specific Type I error (i.e., a “false alarm”) probability, the probability that at least some Type I errors are committed increases, often sharply, with the number of hypotheses. This may have serious consequences if the set of conclusions must be evaluated as a whole [121].

Abdi[1] gives an example of how to compute the probability of rejecting the null hypothesis at least once in a series of tests when the null hypothesis is true. Suppose for each test the probability of making a Type I error is equal to $\alpha = 0.05$. Given that "making a Type I error" and "not making a type I error" are complementary events, the probability of not making a Type I error on one test is equal to

$$1 - \alpha = 1 - 0.05 = 0.95$$

If the tests are independent, for a series of $C$ tests, the probability of not making a Type I error for the whole set of tests is

$$(1 - \alpha)^C$$

Therefore, the probability of making at least one Type I error on the series of tests is equal to

$$1 - (1 - \alpha)^C$$

For our example, with an $\alpha$ level of 0.05 and a set of 10 independent tests, the probability of incorrectly rejecting the null hypothesis is

$$1 - (1 - 0.05)^{10} = 0.401$$

Before we consider how to solve this problem, we will briefly give a more formal definition. We are given a number of hypothesis tests to perform, and obtain the $p$ value $p_i$ of test $i$, considered as a single test. We then need to decide which of the hypotheses to reject, given their $p$-values. The procedure we use to make this decision should guarantee that our probability of making at least one Type I error is at most $\alpha$.

It is important to note that we do not know which hypotheses are true. Indeed, if we did know this, there would be no reason for us to perform the tests. As such, our multiple testing procedure should work no matter which hypotheses are true. In Holm's [62] terminology, this is referred to as applying "for free combinations". Hochberg and Tamhane[61] call this "strong control" of the family-wise error rate. All the methods we will consider have this property, although alternatives exist that only apply when all null hypotheses are known to be true[61].

The most well known correction that provides strong control is the Bonferroni correction, which performs each test at a significance level of $\alpha/C$, where $\alpha$ is the desired overall alpha level and $C$ is the number of hypotheses. The Bonferroni method does not assume independence of the tests, as it follows from Boole's inequality, as shown below.

$$\begin{aligned}\alpha =&P(\text{Type I error occurs})\\ =&P\left(\bigcup_{i=1}^{C}\text{Type I error in Hypothesis i}\right)\\ \leq&\sum_{i=1}^{C}P(\text{Type I error in Hypothesis i})\\ =&C\alpha_i\end{aligned} \tag{3.4}$$

The Bonferroni method is a single-stage procedure to control the probability of at least one error in the family. It is an appealing method because it is applicable in essentially any multiple comparison situation. However, as the number of hypotheses increases, the average power for testing the individual hypotheses becomes smaller. Some modifications of the Bonferroni procedure have improved the situation, and they are discussed below.

A powerful sequentially rejective multiple test procedure has been proposed by Holm[62]. Holm's test is also based on the Boole's inequality, and it is actually a sequentially rejective Bonferroni test. After performing some statistical test, the $p$-value for each individual test $Y_i$, say $p_i$, is obtained. The classical Bonferroni test can be performed by comparing all the obtained $p_i's$ to $\frac{\alpha}{C}$, where $P(p_i \leq \alpha|H_0) = \alpha, \forall i$ and $C$ is the number of hypotheses. In Holm's procedure, the tests need to be ordered by their critical levels, denoting by $p^{(1)} \leq p^{(2)} \leq \ldots \leq p^{(C)}$. The test with the smallest $p$-value will be tested with a Bonferroni correction for all tests(say $C$ tests). If the test is not significant, then we do not reject any hypotheses. Otherwise, the test with the smallest $p$-value is rejected, and the test with the second smallest $p$-value is then corrected with a Bonferroni method for a family of $C - 1$ tests. The procedure stops when the first non-significant test is obtained or when all the tests have been performed. Figure 3.1 illustrates the procedure of the Holm's test[62].

The Holm's procedure increases the power of the statistical tests while keeping the family-wise Type I error under control. Based on the same test statistics, the probability of rejecting any set of false hypotheses using Holm's test is larger than or equal to the classical Bonferroni test. As Holm[62] discussed, the great advantage with this test(as well as the classical Bonferroni test) is its flexibility. There are no restrictions on the type of tests, the only requirement is that it should be possible to calculate the obtained level for each separate test.

There are other powerful and complex procedures to keep the family-wise Type I error under control. However, they are all based on the assumption that the individual tests are independent. Simes[123] proposed a global test for all hypothesis in a set of n hypothesis $\{H_1, H_2, \ldots, H_n\}$. The set of hypothesis are all true with probability $1 - \alpha$ if $p_i > \frac{i\alpha}{n}$ for $i = 1, \ldots, n$, where $p_i's$ are the ordered $p$-values. Simes[123] provided

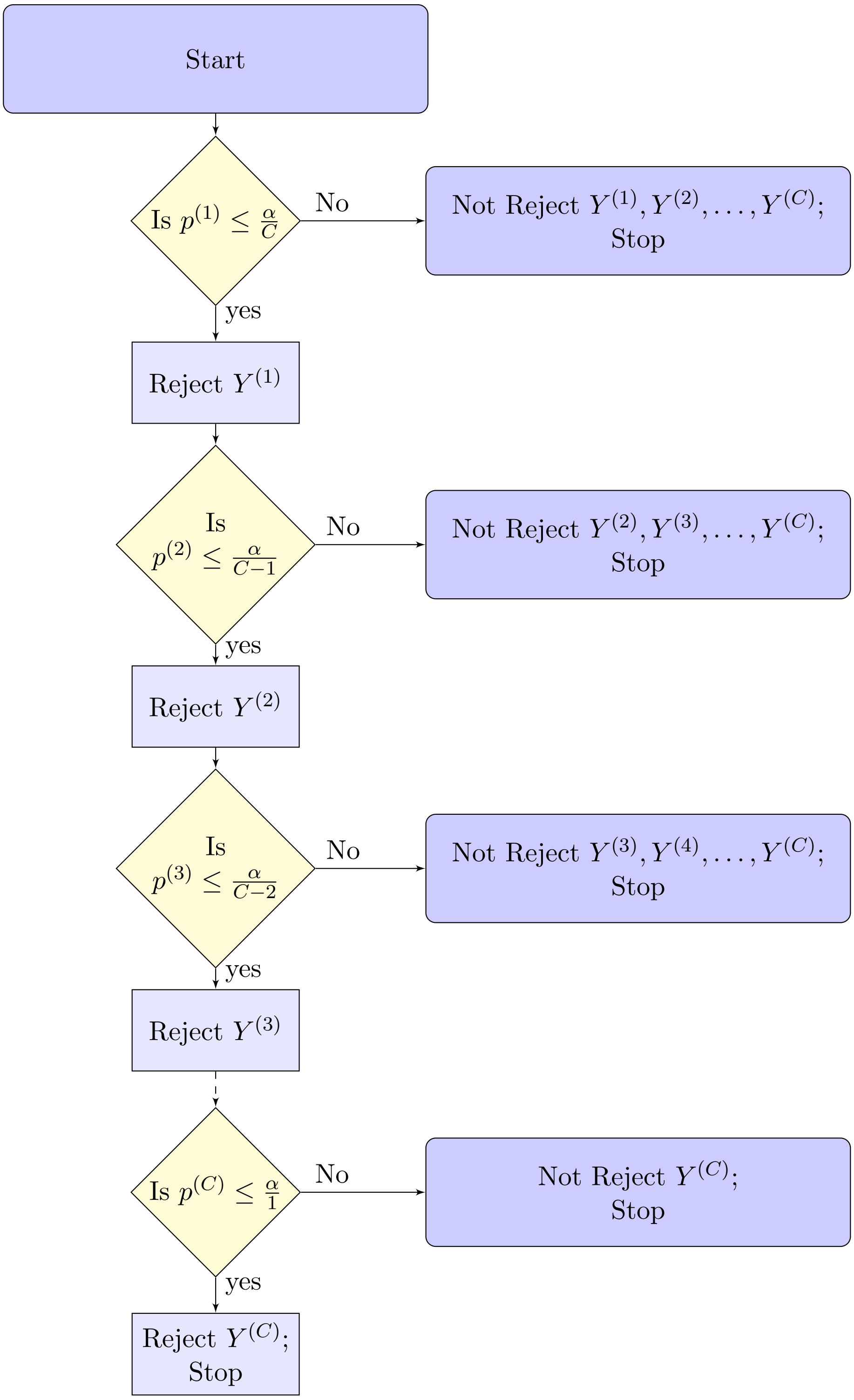

Start
Is $p^{(1)} \leq \frac{\alpha}{C}$
No
Not Reject $Y^{(1)}, Y^{(2)}, \ldots, Y^{(C)}$; Stop
yes
Reject $Y^{(1)}$
Is $p^{(2)} \leq \frac{\alpha}{C-1}$
No
Not Reject $Y^{(2)}, Y^{(3)}, \ldots, Y^{(C)}$; Stop
yes
Reject $Y^{(2)}$
Is $p^{(3)} \leq \frac{\alpha}{C-2}$
No
Not Reject $Y^{(3)}, Y^{(4)}, \ldots, Y^{(C)}$; Stop
yes
Reject $Y^{(3)}$
Is $p^{(C)} \leq \frac{\alpha}{1}$
No
Not Reject $Y^{(C)}$; Stop
yes
Reject $Y^{(C)}$; Stop

FIGURE 3.1: Flow Chart for Holm's Procedure

simulation results which indicate that the probability of a Type I error under the procedure does not exceed the significance level $\alpha$. However, Simes' test does not address the problem of how the individual hypothesis $H_i$, for $i = 1, \ldots, n$ are to be made for this procedure. Hochberg[60] and Hommel[63] provide more powerful methods to control the family-wise Type I error by combining Simes' results (provided that the Simes' tests have level $\alpha$). Rom[112] gives an improvement procedure for Hochberg's method with a slightly higher $p$-value. Therefore, it is more powerful than Hochberg's method.

In the discussion above, we have assumed that the procedure we apply provides us with a decision to reject or not reject each hypothesis. However, these methods did not give us more detailed information about the significance of individual hypotheses. In the case of a single hypothesis, a $p$-value provides us with this information. For a given multiple comparison procedure, there is an analogue of the $p$-value, known as the adjusted $p$-value [132].

To define the adjusted $p$-value, it is instructive to first recall the definition of a $p$-value. The $p$-value is the smallest significance level $\alpha$ for which we reject the null hypothesis. Similarly, the adjusted $p$-value of a hypothesis within a multiple hypothesis test is the smallest family-wise error rate $\alpha$ for which we reject the hypothesis, given the individual $p$-values of all hypotheses. These adjusted $p$-values may be compared to the desired family-wise error rate to decide whether to reject an individual hypothesis.

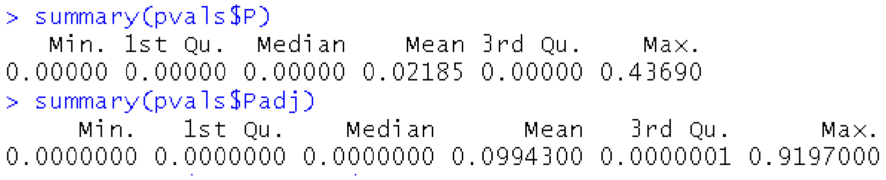
To illustrate the definition of an adjusted $p$-value, consider an experiment with 3 tests, with individual $p$-values 0.015, 0.037 and 0.023. If we are performing a Bonferroni correction with a family-wise error rate of $\alpha$, a hypothesis will be rejected if $p_i < \frac{\alpha}{3}$. Therefore, the adjusted $p$-value of hypothesis $i$ is given by $p_{i,adj} = 3\alpha$, resulting in adjusted $p$-values of 0.045, 0.111 and 0.069 in our example. We may then decide whether to reject hypotheses at family-wise error rate $\alpha = 0.05$ by comparing the adjusted $p$-values to $\alpha$. In the example, we reject only the first hypothesis, giving us the same result as if we had done a Bonferroni correction directly. Details of computing adjusted $p$-values for other corrections are given in [132].

## 3.4 Applying Ljung-Box test with Holm-Bonferroni correction to Ford's data

In this section, we discuss how to detect whether the times to breakdown of Ford's individual machines are correlated with each other. We also demonstrate the effects of correcting for multiple comparison tests using real data. In particular, the question of whether the test results of some hypothesis are different after the correction will be answered. Testing for autocorrelation in the data may have an effect on potential decision making, such as which model is to choose for each individual machine.

In the particular case of Ford's machines, it may be reasonable to assume independence of the hypothesis tests. However, there are several factors which may affect multiple machines, such as outages of the monitoring system. In addition to that, parallel machines on the same work station may also affect each other. Based on the experience of Ford's engineers, during seasons when the demand for engines is low, some of the machines may be switched off, leaving a few machines on the same work station working full time. Therefore, some machines may experience high frequency of breakdowns, while others have nearly zero failure during certain period of the year. However, the data of when and which machines are shut down due to the reason above is not available. As such, we will not assume independence and apply a Holm-Bonferroni correction. The correction will be applied using adjusted $p$-values, indicating how significant the autocorrelation is for each machine.

We apply the Ljung-Box test to Ford's data, using a Holm-Bonferroni correction to correct for multiple hypothesis testing. To illustrate the effects of correcting for multiple hypothesis test, we compare the $p$-value of individual hypothesis test with its adjusted $p$-value. Figure 3.2 shows the summary of the $p$- and adjusted $p$-values in R. The adjusted $p$-values have a larger mean than the $p$-values. In addition to that, most of the machines have very small $p$- and adjusted $p$-values, since the $3^{rd}$ quartile of both the $p$- and adjusted $p$-values are close to zero.



```
> summary(pvals$P)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
0.00000 0.00000 0.00000 0.02185 0.00000 0.43690 
> summary(pvals$Padj)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
0.0000000 0.0000000 0.0000000 0.0994300 0.0000001 0.9197000 
```

FIGURE 3.2: R results of the summary of $p$ and adjusted $p$-values

To have a clearer view of the difference between the $p$- and adjusted $p$-values, we plot the ratio of adjusted $p$-values against unadjusted $p$-values. This is illustrated in Figure 3.3, in which the horizontal line represents value 1, and therefore, all of the adjusted $p$-values are larger than the unadjusted $p$-values. The ratios vary, and some of the $p$-values can be over 30 times larger after being corrected for multiple hypothesis testing. However, since most of the unadjusted $p$-values are very close to zero, the adjusted $p$-values may still be small, and hence the null hypothesis may still be rejected after correcting for multiple hypothesis testing.

Table 3.1 shows the information of machines that no longer show evidence of having autocorrelated up times, after correcting the $p$-values. For instance, before correcting for multiple hypothesis tests, the $p$-value of the hypothesis test for Machine OP10.5 is 0.0047. Since it is below the threshold of 0.05, the null hypothesis that Machine OP10.5 has no autocorrelated up times is rejected. In other words, the up times of Machine

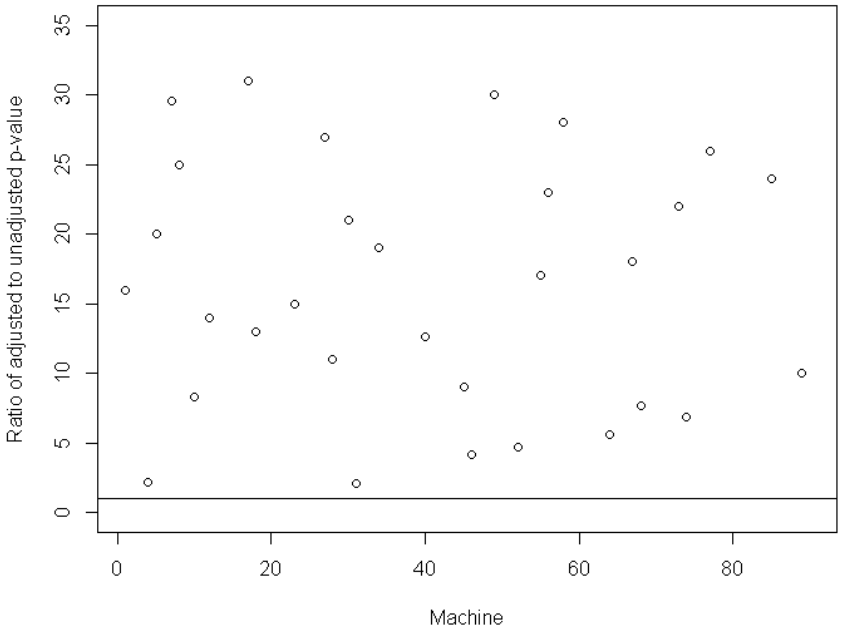


Figure 3.3: Ratio of adjusted $p$-values against unadjusted $p$-values

OP10.5 is considered to be statistically autocorrelated. However, after correcting for multiple hypothesis testing, the adjusted $p$-value for the hypothesis test is 0.0606, which is above the threshold for rejection. Therefore, there is no evidence to reject the null hypothesis, and we conclude that Machine OP10.5 does not have autocorrelated up times. Similar situations happen on Machine OP10.1 and Machine 100.6. As such, correcting for multiple hypothesis tests can have an effect on whether to reject the null hypothesis or not. In the particular case of Ford, before correcting for multiple hypothesis tests, the the number of machines that have autocorrelated up times is 78. After the correction, the number of machines drops to 75.

Table 3.1: Information of machines that no longer show evidence of having autocorrelated uptimes after correcting the $p$-values

| Machine | < 5 minutes | > 1 week | Total | $p$-value | adjusted $p$-value |
|---|---|---|---|---|---|
| OP10.5 | 1198 | 1 | 3039 | 0.0047 | 0.0606 |
| OP10.1 | 1383 | 1 | 3609 | 0.0063 | 0.0698 |
| OP100.6 | 310 | 2 | 1001 | 0.0048 | 0.0606 |

As discussed in Section 2.8, any up time that is shorter than 5 minutes be removed from the data. It is worth checking if excluding these data records will affect the autocorrelation analysis. Table 3.2 illustrates machines that have autocorrelated up times in the original data, but show no evidence of having autocorrelated up times after excluding some of the data records. In Table 3.2 , "$p_{rawadj}$" represents the adjusted $p$-values of the null hypothesis based on the original data, whereas "$p_{adj}$" is the adjusted $p$-values of the null hypothesis based on the truncated data. For instance, Machine OP100.4 has a $p_{rawadj}$ that is less than 0.05, and therefore it is considered to have autocorrelated up times based on the original data. After removing some of the data records, the $p_{adj}$ is greater than 0.05. Hence, no evidence suggests Machine OP100.4 has up times that are autocorrelated. Similar situations happen on Machine OP10.5, OP10.1, OP30, OP100.6, OP60, and OP120.1, in which the original data shows evidence of autocorrelated up times, but the truncated data does not.

TABLE 3.2: Information of machines that no longer show evidence of having autocorrelated uptimes after excluding some data records

| Machine | $<$ 5 minutes | $>$ 1 week | Total | $p_{adj}$ | $p_{rawadj}$ |
|---|---|---|---|---|---|
| OP100.4 | 303 | 1 | 991 | 0.9197 | 8.238859e-06 |
| OP10.5 | 1198 | 1 | 3039 | 0.0606 | 0.000000e+00 |
| OP10.1 | 1383 | 1 | 3609 | 0.0698 | 1.347589e-12 |
| OP30 | 902 | 1 | 1965 | 0.9197 | 4.308998e-12 |
| OP100.6 | 310 | 2 | 1001 | 0.06060 | 8.103295e-06 |
| OP60 | 754 | 1 | 1497 | 0.9197 | 0.000000e+00 |
| OP120.1 | 7341 | 1 | 11439 | 0.9197 | 5.931093e-04 |

Table 3.3 illustrates machines that do not have autocorrelated up times in the original data, but show evidence of having autocorrelated up times after excluding some of the data records. For instance, Machine OP10A has a $p_{rawadj}$ that is 1, and therefore, it is considered to not have autocorrelated up times based on the original data. After removing 5897 data records, which account for over 50% of the total records for that machine, the $p_{adj}$ of the null hypothesis is below the threshold of 0.05. Therefore, Machine OP10A is considered to have autocorrelated up times. A similar situation happens on Machine OP10A-30G, in which there is no evidence of autocorrelated up times in the original data, but there is after truncating some of the data records.

TABLE 3.3: Information of machines that show evidence of having autocorrelated uptimes only after excluding some data records

| Machine | $<$ 5 minutes | $>$ 1 week | Total | $p_{adj}$ | $p_{rawadj}$ |
|---|---|---|---|---|---|
| OP10A | 5897 | 1 | 11291 | 3.785972e-12 | 1 |
| OP10A-30AG | 4363 | 1 | 7947 | 3.340906e-06 | 1 |

## 3.5 Conclusion

In this chapter, we discussed how autocorrelation in time series may be defined and how it may be detected using a Ljung-Box test. We then discussed the problem of multiple hypothesis testing and various solutions to it, and illustrated the effects of correcting for multiple hypothesis testing using real data.

Applying the Ljung-Box test, corrected for multiple comparisons, to Ford's data showed that most machines have statistically significant autocorrelation. In some cases, the $p$-value of the Ljung-Box test seems to be severely affected by the presence of significant numbers of outliers in the uptimes. As such, the thresholds for excluding data from the original data records need to be carefully chosen, as they may affect future decision making on which model to choose for each individual machine.

# Chapter 4

# Time series analysis of the up times

## 4.1 Introduction

In the previous chapter, we analysed the times between breakdowns of Ford's machines and found that, for most machines, they were positively correlated. The implications of this result are discussed in this chapter. In particular, as the times between breakdowns(up times) are autocorrelated, they cannot be treated as an independent and identically distributed sample from a fixed density. This implies that they cannot be directly modelled with parametric or nonparametric density estimation approaches and other techniques are needed.

In this chapter, we consider treating the times between breakdowns as a time series and apply Autoregressive Moving Average (ARMA) models to predict the next uptime from some of the past uptimes and previous error terms. We begin the chapter by reviewing ARMA models, including their definition and some fitting methods.

We then consider how ARMA models may be applied to the machine uptimes. We present details of the methodology applied and some diagnostics of the models chosen to assess their fit.

The chapter concludes with a brief summary of ARMA models and their applicability to modelling the uptimes of Ford's machines.

## 4.2 Gaussian Autoregressive Moving Average (ARMA) models

In this section, we briefly review Gaussian Autoregressive Moving Average models. As these models consist of both Autoregressive and Moving Average components, we begin by reviewing the simpler Autoregressive and Moving Average models. Thereafter, we combine them into ARMA models. The discussion here is based on the textbook by Box and Jenkins[21].

### 4.2.1 White noise

The simplest process we will review is Gaussian white noise. In a white noise process, every observation is independent of the previous one, so the process has no "memory". Furthermore, each observation has the same distribution. In the special case of Gaussian white noise, each observation is normally distributed with zero mean and a fixed variance.

Figure 4.1 shows an example white noise series. As can be seen in the figure, the series shows no obvious pattern. This is confirmed by its autocorrelation plot, shown in Figure 4.2.

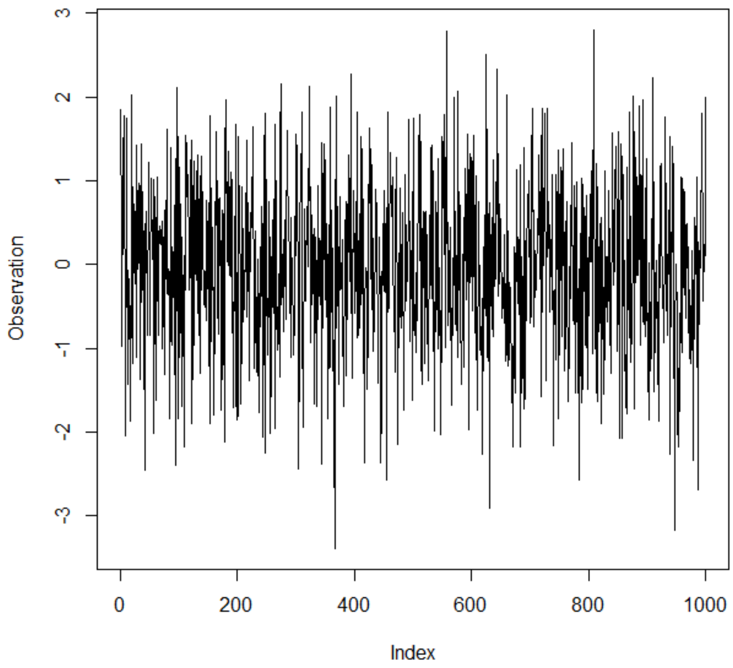


FIGURE 4.1: An example time series generated from a Gaussian white noise process with zero mean and unit variance

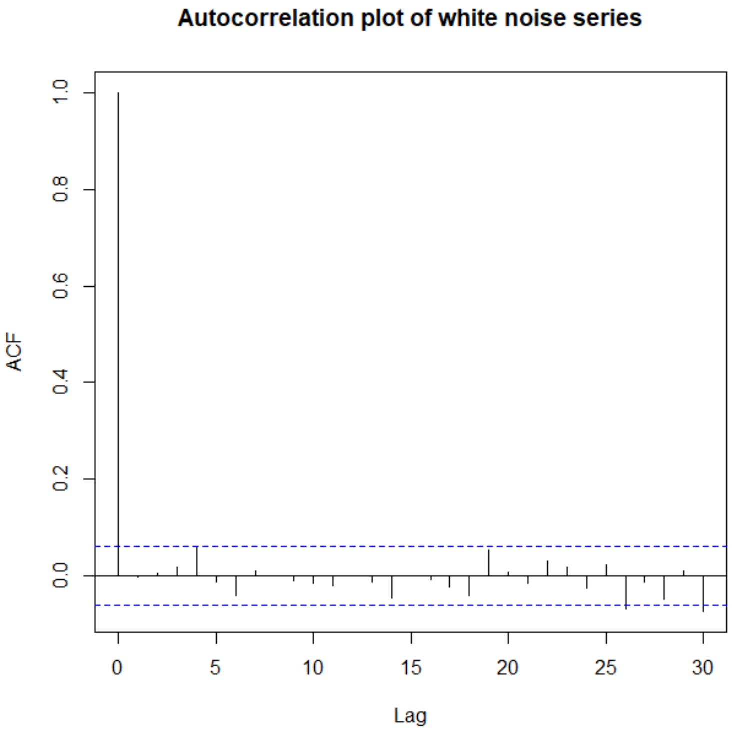


FIGURE 4.2: Autocorrelation plot generated from the example white noise sequence

### 4.2.2 Moving Average(MA) models

Moving average models use the past forecast errors in a regression-like model. Therefore, the observed series is a moving average of an underlying series of independent and identically distributed (iid) normal variables. Let $\epsilon_i$ be a Gaussian white noise process, with zero mean and variance $\sigma^2$. Then, the series $Y_1, Y_2, \ldots$ is a moving average (MA) process if for some $k$, $Y_i$ is given by (4.1).

$$Y_i = \sum_{l=1}^{k} \beta_l \epsilon_{i-k} + \epsilon_i \tag{4.1}$$

The $\beta_l$ are constants that control how strongly the process is affected by prior data. if all the $\beta_l$ are zero, the process is a white noise process. The larger the $\beta_l$, the more impact prior values have on the process.

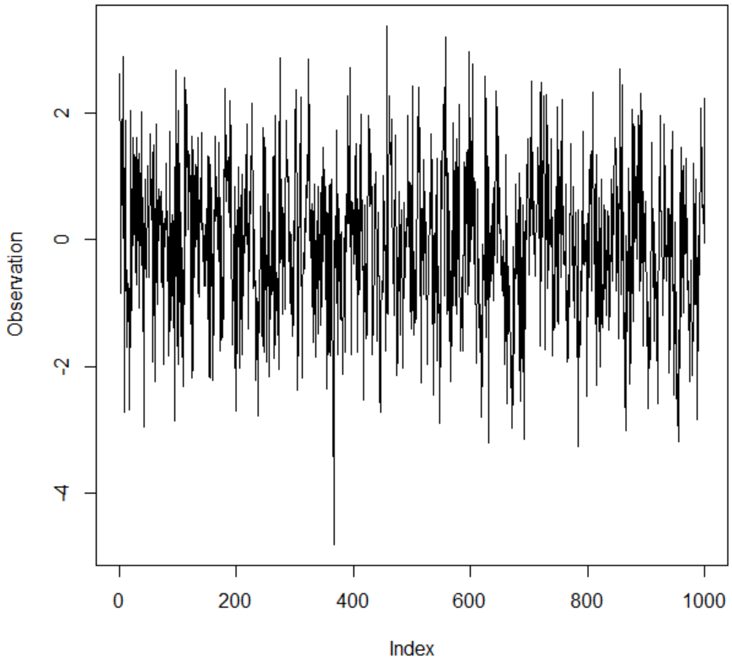


FIGURE 4.3: An example time series generated from an MA(3) process

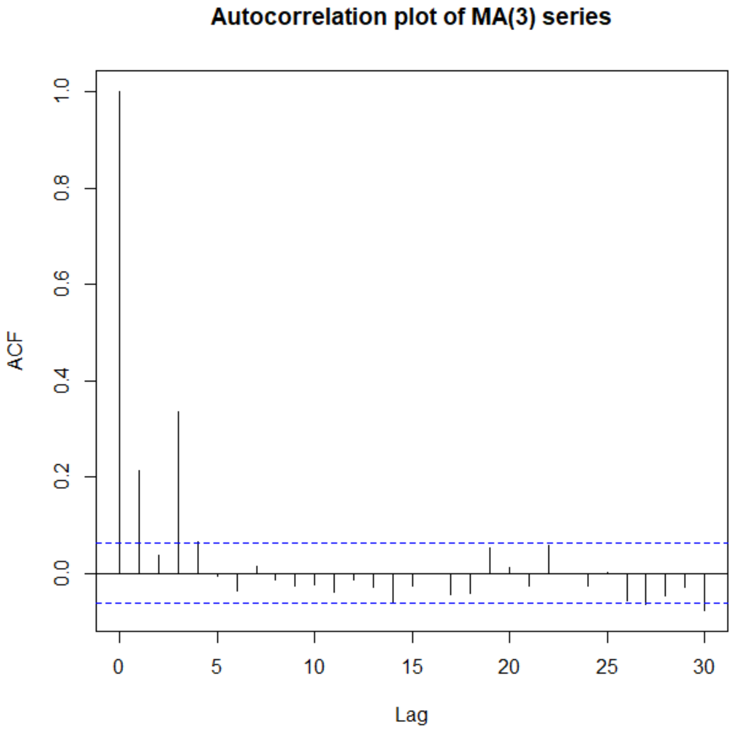


FIGURE 4.4: Autocorrelation plot generated from the example MA(3) sequence

Figure 4.3 shows an example $MA(3)$ process, where $\beta_1 = 0.5$, $\beta_2 = -0.2$ and $\beta_3 = 0.5$. The autocorrelation plot of this series is shown in Figure 4.4. As can be seen in the figure, the autocorrelation is negligible outside of the first three lags. This is the characteristic property of an $MA(q)$ process: Only the first $q$ autocorrelations are nonzero.

### 4.2.3 Autoregressive(AR) models

We will now briefly review Gaussian Autoregressive (AR) models. In these models, the value of a time series is allowed to depend on its past values. As the name indicates, Gaussian AR models use a normally-distributed error term. The term autoregressive indicates that it is a regression of the variable against itself.

Let $Y_1, Y_2, \ldots$ be a stationary time series, with a possibly non-zero mean. Then, the $AR(k)$ model for the sequence of $Y_i$ is as given by (4.2). Here, the $\mu$ is constant; the $\beta_l$ control the strength of “memory”and changing them results in different time series patterns; the $\epsilon_i$ are independently and identically distributed normal variables $N(0, \sigma^2)$, which make up the variability that is part of the system when it moves from one period to the next. The variance of the error term will only change the scale of the series, but not the patterns; The integer $k$ represents the order of the AR model.

$$Y_i = \mu + \sum_{l=1}^{k} \beta_l Y_{i-l} + \epsilon_i \tag{4.2}$$

Figure 4.5 shows an example for $k = 1$, with $\beta_1 = 0.5$. The corresponding autocorrelation plot is shown in Figure 4.6. The autocorrelation plot shows a gradually decaying pattern. Any $AR(p)$ process will have nonzero autocorrelation for all lags, decaying with time.

### 4.2.4 Autoregressive Moving Average (ARMA) models

We will now combine autoregressive and moving average models into Autoregressive Moving Average (ARMA) models. As before, let $Y_i$ be a stationary time series, and let $p$ and $q$ be positive integers. Then, $Y_i$ is an $ARMA(p, q)$ process if, for $\epsilon_i$ iid normal with zero mean and constant variance, (4.3).

$$Y_i = \mu + \sum_{l=1}^{p} \beta_l Y_{i-l} + \sum_{k=1}^{q} \gamma_k \epsilon_{i-k} + \epsilon_i \tag{4.3}$$

Figure 4.7 shows an example time series, generated by combining the example MA(3) process and AR(1) processes above. The corresponding autocorrelation plot is in Figure 4.8. Depending on the values of the coefficients, an ARMA(p,q) correlation plot may

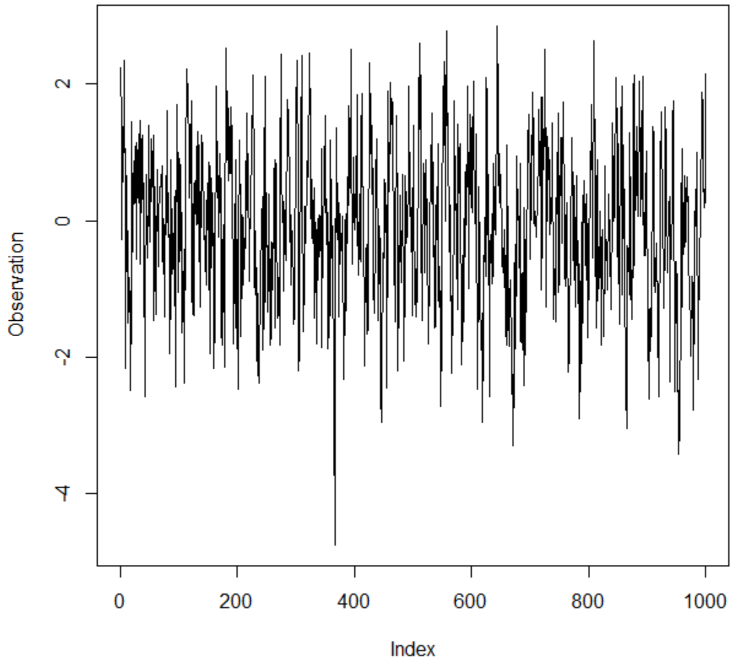


FIGURE 4.5: An example time series generated from an AR(1) process

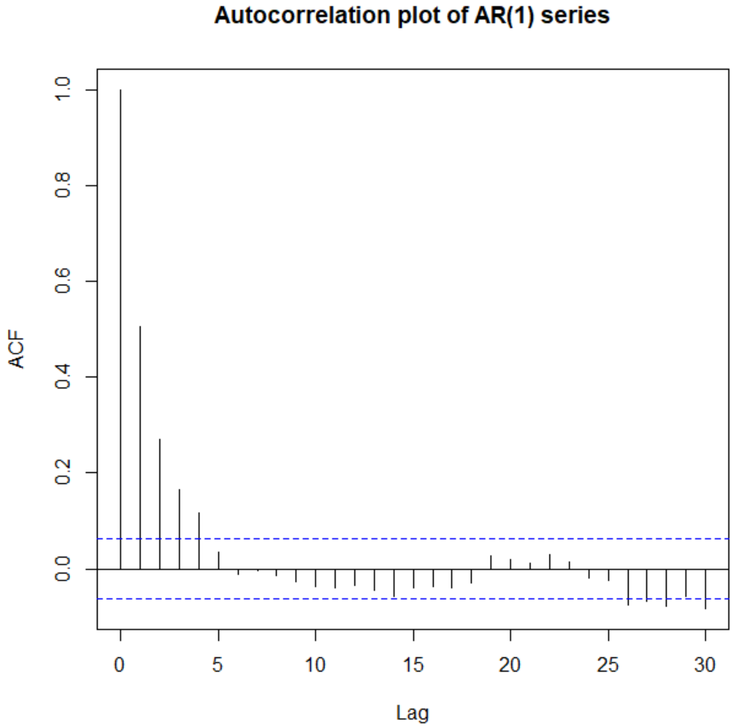


FIGURE 4.6: Autocorrelation plot generated from the example AR(1) sequence

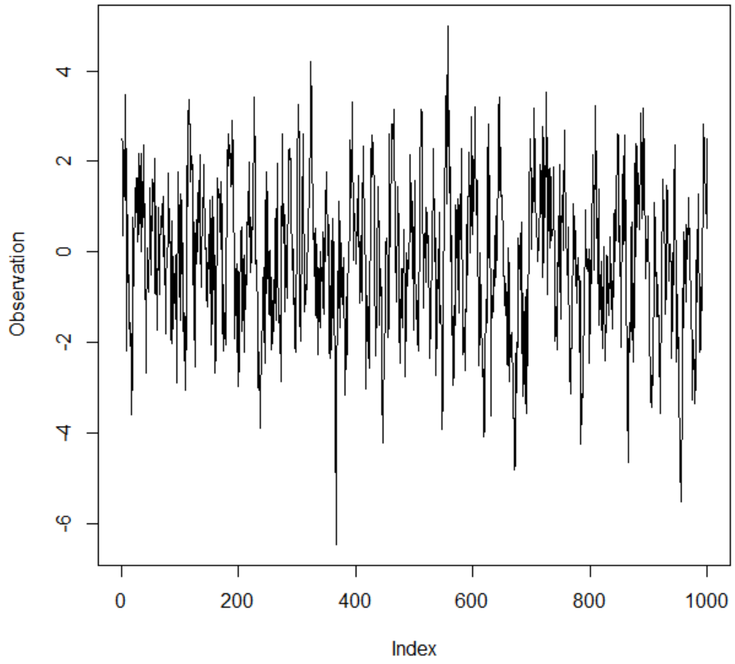


Figure 4.7: An example time series generated from an ARMA(1,3) process

look similar to either an AR(p) or MA(q) correlation plot. In this particular case, the correlation plot closely resembles that obtained earlier for AR(1).

### 4.2.5 Fitting ARMA models

We will now discuss how an ARMA model may be fitted to an observed time series $y_1, y_2, \dots y_n$. To do so, there are two problems that need to be solved. First, the parameters of an $ARMA(p, q)$ model for given $p$ and $q$ need to be estimated. Second, an appropriate $p$ and $q$ need to be chosen. The approaches to these problems discussed here are those of the R forecast package[64].

We will consider the second problem, of choosing $p$ and $q$, first, as its solution has implications for the problem of estimating the parameters. The problem of choosing a suitable $p$ and $q$ may be defined as choosing the "best" $ARMA(p, q)$ model out of a set of possible such models. In the literature, a number of methods to solve this *model selection* problem have been proposed. For a review of model selection, see Zucchini[137].

The particular approach to model selection used by the forecast package is based on Akaike's Information Criterion(AIC). This criterion, for any statistical model with $k$ parameters, is given by (4.4). Here, $L$ is the maximum of the likelihood of the model. For an $ARMA(p, q)$ model with a constant term, this equation reduces to (4.5).

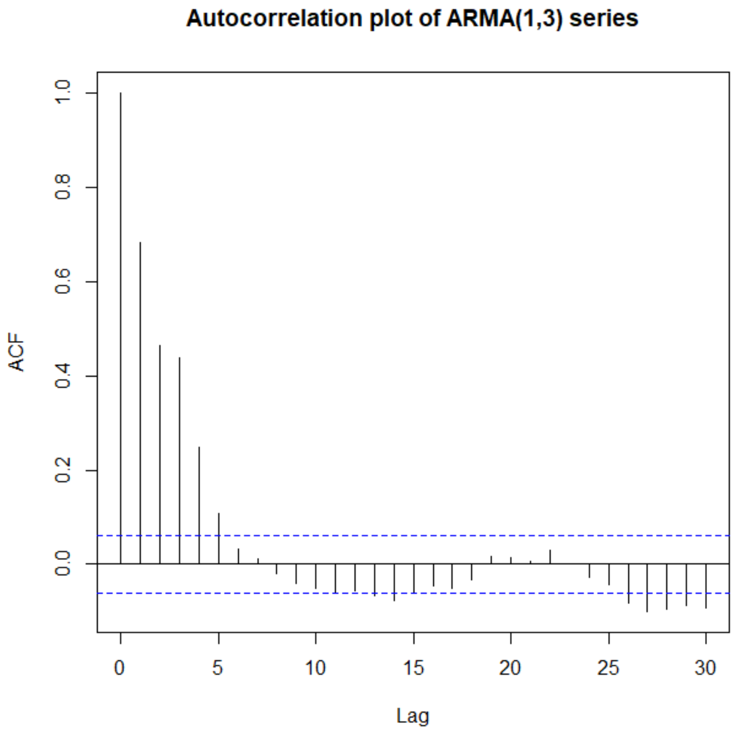


Figure 4.8: Autocorrelation plot generated from the example ARMA(1,3) sequence

$$AIC = -2logL + 2k \tag{4.4}$$

$$AIC = -2logL + 2(p + q + 1) \tag{4.5}$$

Given a set of models and their maximum-likelihood estimates, model selection based on AIC selects the model with the smallest AIC. For a large set of models, such as the set of all $ARMA(p, q)$ models with $p \leq 20$ and $q \leq 20$, this may not be computationally feasible.

As an alternative to minimising AIC globally over the model set, the forecast package implements a stepwise model selection algorithm. This algorithm starts with the best model out of a small set of candidates and then iterates, attempting to make small changes to the values of $p$ and $q$ at each step until no further change improves the model. Pseudocode for the algorithm is given in Algorithm 1.

```
Set the current model to the best model chosen from $ARMA(2,2)$, $ARMA(0,0)$,
  $ARMA(1,0)$, $ARMA(0,1)$;
repeat
    Set (p,q) to the order of the current model;
    Select the best candidate model from $ARMA(p-1,q)$, $ARMA(p+1,q)$,
      $ARMA(p,q-1)$, $ARMA(p,q+1)$, $ARMA(p-1,q-1)$, $ARMA(p+1,q+1)$;
until The best candidate model is no better than the current model;
```

**Algorithm 1:** The stepwise model selection process used in the forecast package[64] for ARMA models

In Algorithm 1, a model may occasionally be difficult to fit due to numerical difficulties or due to the parameters chosen. In these cases, the forecast package excludes the candidate model. Details of this can be found in [64].

Since the forecast package uses AIC to select $p$ and $q$, it must use maximum-likelihood estimation to estimate the parameters of each model. However, since in $ARMA(p,q)$ models each observation depends on the $p$ previous observations and $q$ previous errors, the likelihood depends on the unknown values of the series before the first observation. A solution, implemented in R, is to assume an uninformative prior on these unknown values and apply Kalman filters to compute the likelihood. The details of this process may be found in the textbook by Durbin and Koopman[42]. The piece of R code that demonstrates how to fit ARMA models for the uptimes of Ford's machines is seen below. In particular, lines 8 and 9 are the implementation of Algorithm 1.

```r
1   diagplots <- function(f, r, idxs, fname){
2     par(mfrow=c(2,2))
3     #Plot 1: designed to assess whether residuals are constant-variance over time.
4     plot(idxs,r,xlab='Index',ylab='Residual')
5     #Plot 2: Standard residuals vs fitted plot to assess uniform variance.
6     plot(f,r,xlab='Fitted value',ylab='Residual')
7     #Plot 3: QQ-plot of residuals, to assess normality
8     qqnorm(r)
9     qqline(r)
10    #Plot 4: Acf plot of residuals, to assess independence
11    acf(r,main='Correllogram of residuals')
12    savePlot(fname,'png')
13    par(mfrow=c(1,1))
14  }
15
16  diagar <- function(m,x,fname){
17    order <- m$arma[1]+m$arma[2]
18    idxs <- (order+1):length(x)
19    f <- fitted(m)[idxs]
20    r <- resid(m)[idxs]
21    diagplots(f, r, idxs,fname)
22  }
23
24  plotassessfit <- function(act, sim, fname.acf, fname.qq, lag.max=35){
25          par(mfrow=c(1,1))
26          obs.acf <- acf(act, lag.max = lag.max, plot=FALSE)
```

```
        sim.acf <- acf(sim, lag.max = lag.max, plot=FALSE)
        plot(0:35, obs.acf$acf, 'l', xlab='Lag', ylab='Autocorrelation')
        lines(0:35, sim.acf$acf, col='blue')
        savePlot(fname.acf, 'png')
        qqplot(act, sim,xlab='Data', ylab='Simulated')
        abline(0,1)
        savePlot(fname.qq, 'png')
}

for(l in levels(rejected$Type)){
    max.n <- max(rejected$Number[rejected$Type==l])
    if(max.n >= 0){
        for( n in 0:(max(rejected$Number[rejected$Type==l]))){
            upt <- toreal(get.machine(durs, l, n))
            orig <- get.machine(durs, l, n)
            if(length(upt) > 1){
                model <- auto.arima(upt, d=0, D=0, max.p = max.p, max.q = max.q, max.P = 0, max.Q = 0, approximation=FALSE, stepwise=TRUE, max.order=max.p+max.q)
                rejected$arma.p[rejected$Type == l & rejected$Number == n] = model$arma[1]
                rejected$arma.q[rejected$Type == l & rejected$Number == n] = model$arma[2]
                aic <- transform_aic(AIC(model), orig)
                rejected$arma.aic[rejected$Type == l & rejected$Number == n] = aic
                diagar(model, upt, paste('arma-diag-', l, '-', n, '.png',sep=''))
                simmed <- simulate.Arima(model, length(upt)*10, future=FALSE)
                plotassessfit(orig, fromreal(simmed), paste('arma-acf-', l, '-', n, '.png',sep=''),
                              paste('arma-qq-', l, '-', n, '.png', sep=''))
                plotassessfit(upt, simmed, paste('arma-trans-acf-', l, '-', n, '.png',sep=''),
                              paste('arma-trans-qq-', l, '-', n, '.png', sep=''))
            }
        }
    }
}
```

## 4.3 Applying ARMA models to the uptimes of Ford's machines

In this section, we consider how the ARMA models reviewed previously may be applied to the uptimes of Ford's machines. We begin by discussing the fitting methodology and then assess the ARMA fits obtained.

ARMA models, as reviewed previously, assume that each observation has a normal distribution. This implies that the observation may be on the entire real line. Normally, a machine uptime is strictly positive, and a logarithmic transform would suffice to transform the uptime to an observation on the entire real line. However, since we have removed both small and large values, the uptimes in our data are between 5 minutes

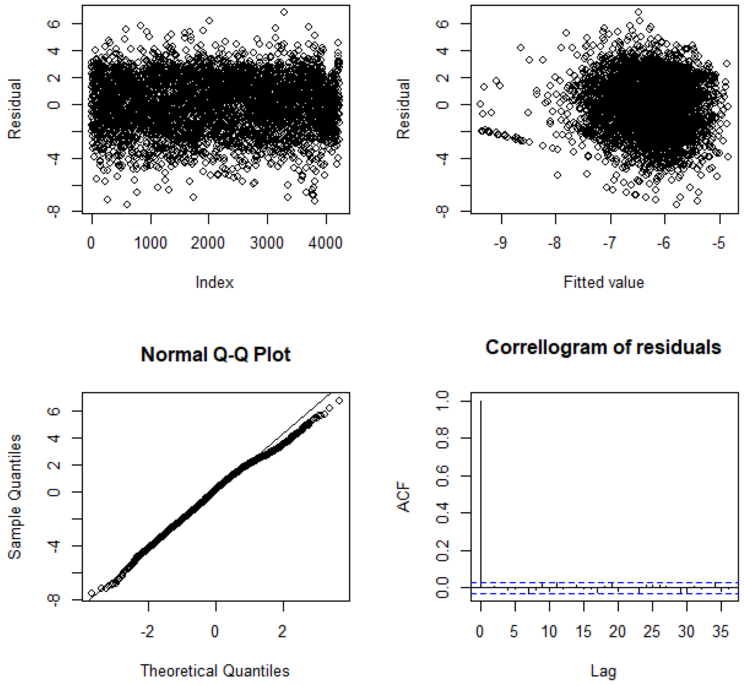


FIGURE 4.9: An example diagnostic plot for an ARMA fit

and 1 week. Letting $a$ be 5 minutes and $b$ one week, an observation on $(a, b)$ may be transformed to the real line using (4.6). Here, $u$ is the uptime to be transformed and $y$ is the transformed real-valued observation.

$$y = \log\left(\frac{u-a}{b-u}\right) \tag{4.6}$$

After transforming the individual uptimes, we fit ARMA models using the procedure described in Subsection 4.2.5. To assess whether these models fit the data, we have used a set of diagnostic plots. Before we discuss the quality of the fits we obtained, we will first discuss these plots. An example of the diagnostic plots is given in Figure 4.9.

In Figure 4.9, the top-left plot shows the residuals versus the index of the residual. If the model fits adequately, this should show a band of points distributed randomly around zero. Since the residuals should have constant variance, the width of this band should not vary. In the particular example shown here, this appears to be the case.

The top-right corner of Figure 4.9 shows the residuals versus the estimated transformed uptime. If the model's fit is adequate, the fitted value and residual should be unrelated to each other and this plot should show a random distribution of points. In this particular example, there is no obvious pattern in the distribution of the points, although a small cluster of points in the bottom-left appears to lie on a line.

The QQ-plot in the bottom-left corner of Figure 4.9 allows us to assess whether the residuals are normally distributed. If the model's fit is adequate, this should be the case and the QQ-plot will approximate a straight line. In this particular example, the plot is satisfactory.

Finally, the bottom-right corner of Figure 4.9 shows the estimated autocorrelation function of the residuals. If there is any unmodelled correlation in the uptimes, this plot should show significant peaks. In this example, none of the peaks are large enough to cause us any concern.

We will now briefly discuss the results of fitting ARMA models to the uptimes of Ford's machines, highlighting particular machines or patterns of interest.

The ARMA model fits well for most of the machines, and the diagnostic plots of the fitting results have the similar patten as in Figure 4.9. There are some machines that show different patterns.

As a first example, consider the diagnostic plots shown in Figure 4.10. For this machine, the QQ plot, residual plot and residual versus fitted value plot all appear to indicate that the fit is relatively poor. In particular, the residual versus fitted value plot shows two distinct clusters of data points.

To investigate this further, we have plotted the uptimes themselves in 4.11. As can be seen in the figure, there's a long run of short uptimes around index 1200. This is most likely due to a run of "try outs" that were unusually long. To determine whether the run of try outs is the cause of the lack of fit seen in Figure 4.10, we have fitted an ARMA model to the series up to index 1150. The diagnostic plots for this fit are shown in Figure 4.12. This figure strongly resembles Figure 4.10 and appears to indicate that the residuals of this model are not symmetric. In all, a different model will be needed to fit this series well. Similar behaviour can be seen in one other machine, OP55.

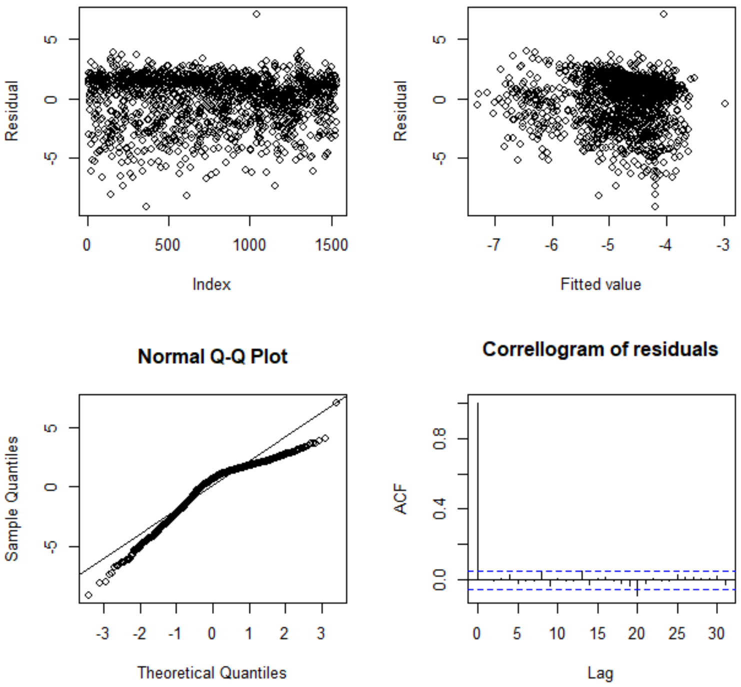


FIGURE 4.10: Diagnostic plots for machine OP130-1

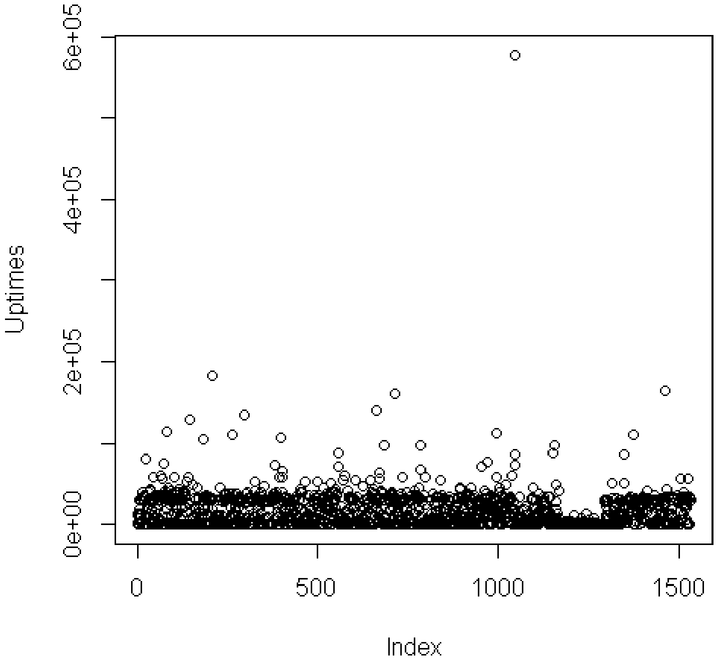


FIGURE 4.11: Uptimes for machine OP130-1

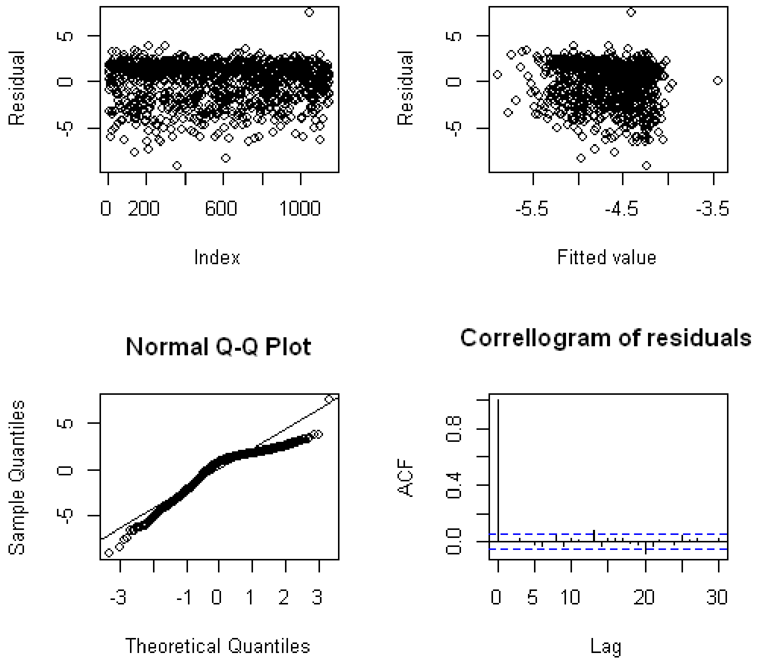


FIGURE 4.12: Diagnostic plots for machine OP130-1, using a prefix of the full series

A second pattern in the poorer ARMA fits is illustrated in Figure 4.13. Most of the plots in this figure look quite reasonable. The exception is the residual versus index plot, which seems to indicate smoothly-changing variance. This suggests that an ARMA model is inappropriate for this machine as the variance of the error terms needs to change over time. An ARMA-GARCH model might be an appropriate choice for machines of this type. The diagnostic plots for the ARMA fits for all machines are given in Appendices A.

## 4.4 Simulating uptimes from ARMA models

In this section, we discuss how uptimes may be generated from a fitted $ARMA(p, q)$ model. The discussion here is based on the implementation of the arima.sim function[108] in R. We will break the process of simulating from an $ARMA(p, q)$ model into several stages:

1. Generating a white noise process

2. Generating an $MA(q)$ series from a white noise process

3. Generating an $ARMA(p, q)$ series from an $MA(q)$ series

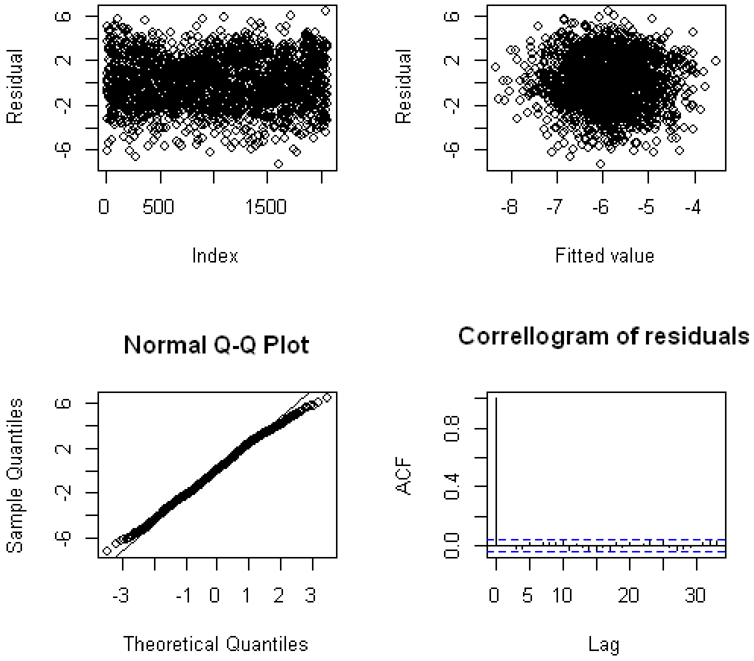


FIGURE 4.13: Diagnostic plots for machine OP110-0

4. Transforming the generated times

**Generating a white noise process** To generate a white noise process with zero mean and variance $\sigma^2$, we generate a series of iid $N(0, \sigma^2)$ random variables. This may be done using any common method of generating Gaussian random variables, for instance a Box-Muller transform. For the details of Box-Muller transform, see Box and Muller[23].

**Generating an $MA(q)$ series from white noise** To generate an $MA(q)$ series from white noise, we need to compute linear combinations of the white noise values. To be more precise, if $\epsilon_i$ is a sequence of white noise variables, the $MA(q)$ process $m_i$ is given by $m_i = \epsilon_i + \sum_{j=1}^{q} \beta_j \epsilon_{i-j}$, where $\beta_j$ are the MA coefficients.

**Generating an $ARMA(p, q)$ series from an $MA(q)$ series** To generate an $ARMA(p, q)$ series from an $MA(q)$ one, we need to perform autoregressive filtering of order $p$ on the $MA(q)$ series. To do so, we compute the output $x_i$ as $x_i = \mu + m_i + \sum_{j=1}^{p} \beta_j x_{i-j}$, where the $\beta_j$ are the coefficients of the AR portion of the model.

The series $x_i$ depends on its own past values, and so depends on initialisation values. For the purposes of simulating a stationary ARMA model, these values are essentially arbitrary and may be taken as zero. However, the initial part of the series should be discarded as it may depend on the choice of initialisation values.

**Transforming the generated times** The times generated above can be any real number. In Ford's application, an uptime can only be between 5 minutes and 1 week. Accordingly, we apply the inverse of the data transformation we have used before. This generates uptimes $u_i = \frac{e^{y_i}b+a}{e^{y_i}+1}$.

## 4.5 Allowing the variance to vary

### 4.5.1 Introduction

In the ARMA models we have discussed so far in this chapter, we assumed that the error process $\epsilon_t$ was white noise, that is, the $\epsilon_t$ are independent and identically distributed random variables. When we applied these models to Ford's data, we noticed that the residuals did not necessarily have constant variance. This indicates that the underlying error process may not have a constant variance. In this section, we consider how the ARMA model may be extended to allow GARCH errors, which enable the variance of the observations to change over time.

### 4.5.2 Defining GARCH models

First, we will briefly review how GARCH models may be defined. The discussion here is based on [13]. To define a GARCH model, we first consider a zero-mean process $y_t$, defined by (4.7). Here, $\epsilon_t$ is given by (4.8).

$$y_t = \sqrt{h_t}\epsilon_t \tag{4.7}$$

$$\epsilon_t \sim N(0,1) \tag{4.8}$$

The key to defining GARCH models is the process $h_t$ in (4.7). Unlike in Gaussian white noise, where $h_t$ is constant, in a $GARCH(p,q)$ model, $h_t$ is given by (4.9).

$$h_t = \alpha_0 + \alpha_1\epsilon_{t-1}^2 + \cdots + \alpha_q\epsilon_{t-q}^2 + \beta_1 h_{t-1} + \cdots + \beta_p h_{t-p} \tag{4.9}$$

Equation (4.9) is similar to the definition of an $ARMA(p,q)$ process, except that the errors have been replaced by their squared values. As such, in a GARCH process, the sign of the error does not matter, only its size. Furthermore, using the squared values ensures that $h_t$ does not become negative, so that $y_t$ is well-defined.

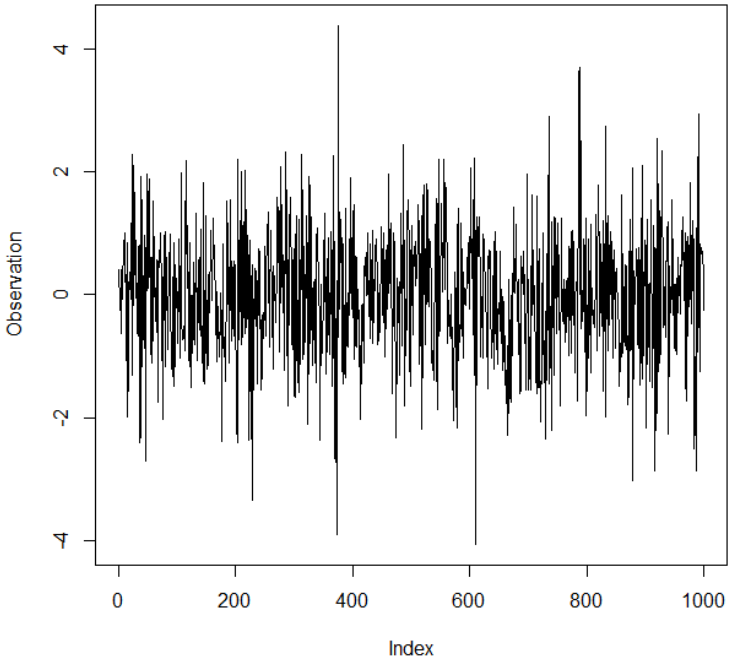


FIGURE 4.14: An example GARCH(1,1) time series

An example GARCH(1,1) time series is shown in Figure 4.14. For this example, $\alpha_0 = 0.1$, $\alpha_1 = 0.8$ and $\beta_1 = 0.1$. The series varies around zero. Although the series' occasionally has large values, it's not entirely clear from this figure that the variance is changing over time.

Figure 4.15 shows the standard deviation of each observation in the GARCH series, conditional on the past values. This figure shows that the standard deviation varies and occasionally becomes much larger for a brief period.

As the variance of each observation in a GARCH series depends on the previous variances and previous squared errors, the squares of the series are correlated. This is shown in the autocorrelation plot in Figure 4.16, which indicates that the squares of the series are significantly autocorrelated. However, as shown in Figure 4.17, the GARCH series itself does not have significant autocorrelation.

To incorporate a time-varying variance in an $ARMA(p, q)$ process, we take (4.3) defining the $ARMA(p, q)$ process and replace the errors $\epsilon_t$ by a GARCH process. This gives (4.10) defining an $ARMA - GARCH(p, q, r, s)$ process.

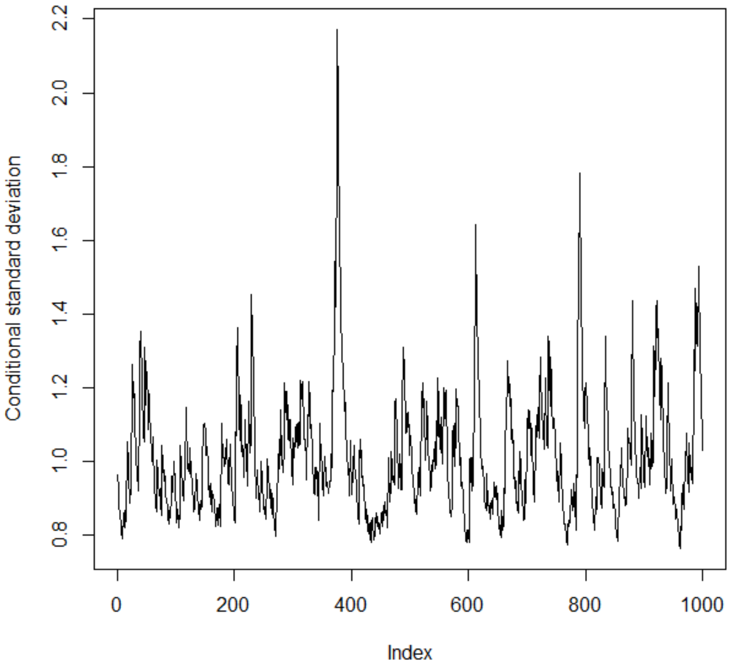


FIGURE 4.15: Conditional standard deviation for the example GARCH(1,1) series

$$x_t = \sum_{k=1}^{q} \alpha_k \epsilon_{t-k} + \sum_{k=1}^{p} \beta_k x_{t-k} + \mu + \epsilon_t \tag{4.10}$$

$$\epsilon_t \sim N(0, h_t) \tag{4.11}$$

$$h_t = \gamma_0 + \gamma_1 \epsilon_{t-1}^2 + \cdots + \gamma_s \epsilon_{t-s}^2 + \delta_1 h_{t-1} + \cdots + \delta_r h_{t-r} \tag{4.12}$$

### 4.5.3 Fitting ARMA-GARCH models

We will now briefly discuss how a GARCH model may be fitted to an observed series. The discussion here is based on [13] and the approach taken by the fGarch R package [133].

To fit an $ARMA - GARCH(p, q, r, s)$ model, we will use maximum likelihood. For the time being, we will assume that the model order, that is, the parameters $p$, $q$, $r$ and $s$, is known. Conditional on the past values of the series, the log-likelihood for a single observation $x_t$ is given by (4.13). Here, $\theta$ denotes the vector of all parameters of the $ARMA - GARCH$ model. The variable $\mu_t$ is the conditional mean of $x_t$, $\sigma_t$ the conditional standard deviation. The term $C$ is a constant independent of $x_t$ and $\Theta$.

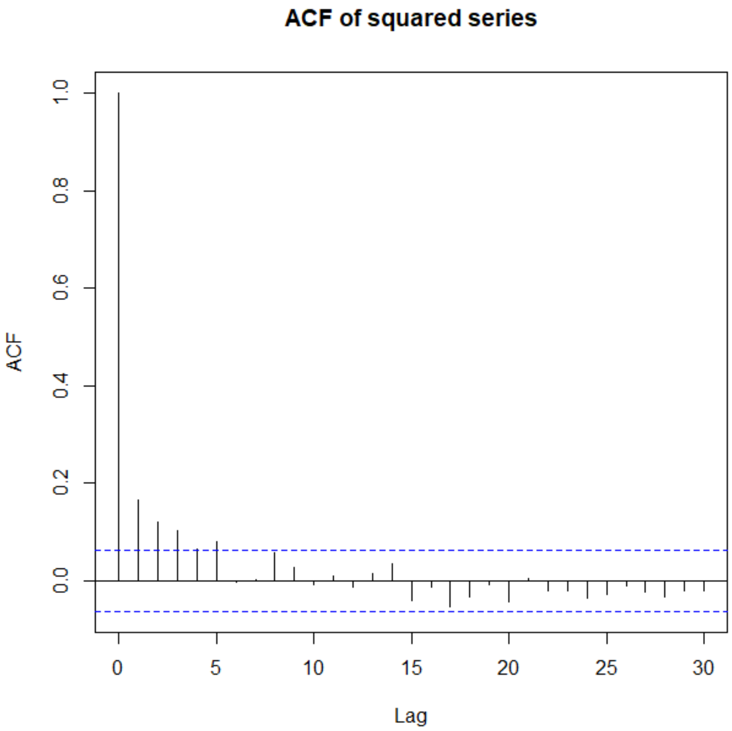


FIGURE 4.16: Autocorrelation of squared GARCH(1,1) series

$$l_t(x_t;\theta) = -\log \sigma_t - \frac{(x_t - \mu_t)^2}{2\sigma_t^2} + C \tag{4.13}$$

Given suitable starting values, the likelihood of the time series $x$ may be obtained using (4.14).

$$l(x;\theta) = \sum_{t=1}^{n} l_t(x_t;\theta) \tag{4.14}$$

To maximise (4.14), any nonlinear optimisation method may be used. The remaining problem is how we may find suitable values of $p$, $q$, $r$ and $s$. In this work, we have done so by first picking the parameters $p$ and $q$, using the ARMA fitting approach of Algorithm 1. To select $r$ and $s$, we then select from the following set of possible choices:

1. $p = 1$, $q = 0$

2. $p = q = 1$

3. $p = 2$, $q = 1$

4. $p = 1$, $q = 2$

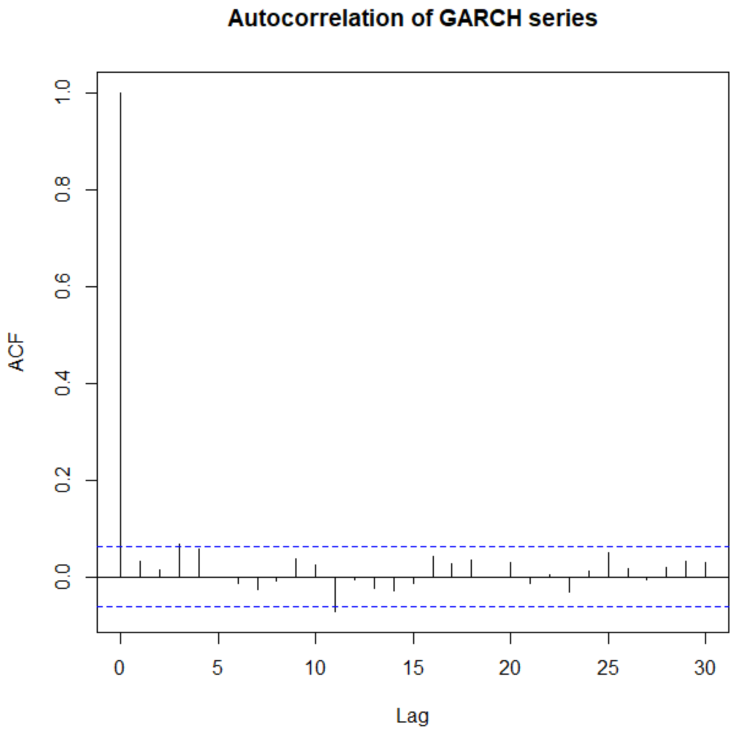


FIGURE 4.17: Autocorrelation of GARCH(1,1) series

As described in [13], the second choice above, $GARCH(1,1)$ captures most time series in applied work, and the final two choices are very rarely needed. As such, we expect that the restriction to these particular parameter values is unlikely to reduce the quality of fit significantly.

### 4.5.4 Applying ARMA-GARCH models to machine uptimes

We will now discuss the results of applying ARMA-GARCH models to the uptimes of Ford's machines. As before, we will mainly use diagnostic plots. In addition to these plots, we will also use the Akaike Information Criterion (AIC), and compare its value to that of the previous ARMA fits for the same machines. The analysis was performed in R, using the code included below.

```
find.garch.fit <- function(arma, upt){
    possible.garch <- list(c(1,0), c(1,1), c(2,1), c(1,2))
    best.AIC <- Inf
    best.model <- NULL
    for(pq in possible.garch){
        f <- as.formula(paste("~arma(",arma$arma[1], ",", arma$arma[2],")+garch("
    ,pq[1],",", pq[2], ")", sep="" ))
        this.garch <- garchFit(f, data = upt, trace=FALSE, cond.dist="norm")
        if(this.garch@fit$ics[[’AIC’]] < best.AIC){
            best.AIC <- this.garch@fit$ics[[’AIC’]]
            best.model <- this.garch
        }
    }
    best.model
}

diaggarch <- function(m, fname){
  r <- m@residuals/m@sigma.t #Standardise residuals for garch.
  f <- m@fitted
  diagplots(f, r, 1:length(f), fname)
}

garchSpecFromGarch <- function(garch){
    rawparms <- coef(garch)
    order <- garch@fit$series$order
    if(order[['u']] > 0){
        ar <- sapply(1:order[['u']], function (i) {rawparms[[paste("ar", i, sep="
    ")]]})
    } else {
        ar <- c()
    }
    if(order[['v']] > 0){
        ma <- sapply(1:order[['v']], function (i) {rawparms[[paste("ma", i, sep="
    ")]]})
    } else {
        ma <- c()
    }
    if(order[['p']] > 0){
        alpha <- sapply(1:order[['p']], function (i) {rawparms[[paste("alpha", i,
     sep="")]]})
    } else {
        alpha <- c()
    }
    if(order[['q']] > 0){
        beta <- sapply(1:order[['q']], function (i) {rawparms[[paste("beta", i,
    sep="")]]})
    } else {
        beta <- c()
    }
    garchSpec(model = list(omega = rawparms[['omega']],
                          ar = ar,
                          ma = ma,
                          mu = rawparms[['mu']],
                          alpha = alpha,
                          beta = beta,
                          delta = 2
                          ), cond.dist = "norm")
}

simgarch <- function(garch, n){
    spec <- garchSpecFromGarch(garch)
    as.numeric(garchSim(spec, n))
}

for(l in levels(rejected$Type)){
    max.n <- max(rejected$Number[rejected$Type==l])
    if(max.n >= 0){
        for( n in 0:(max(rejected$Number[rejected$Type==l]))){
                model.garch <- find.garch.fit(model, upt)
                diaggarch(model.garch, paste('garch-diag-', l, '-', n, '.png',
    sep=''))
                plot(model.garch@sigma.t, type='l', ylab='Conditional standard
    deviation')
                savePlot(paste('garch-sigma-', l, '-',n, '.png'), 'png')
                rejected$garch.p[rejected$Type == l & rejected$Number == n] =
    model.garch@fit$series$order[['p']]
                rejected$garch.q[rejected$Type == l & rejected$Number == n] =
    model.garch@fit$series$order[['q']]
                rejected$garch.aic[rejected$Type == l & rejected$Number == n] =
    transform_aic(model.garch@fit$ics[['AIC']]*length(upt), orig)
                simmed <- simgarch(model.garch, length(upt)*10)
                tryCatch(plotassessfit(orig, fromreal(simmed), paste('garch-acf-'
    , l, '-', n, '.png',sep=''), paste('garch-qq-', l, '-', n, '.png', sep='')),
    error =function(e) print(e))
        }
    }
}
```

An example of the diagnostic plots for an $ARMA - GARCH$ model is shown in Figure 4.18. This figure is based on the same machine as Figure 4.9. The main difference between these plots is that Figure 4.18 is based on the standardised residuals. These are given by $\frac{r_t}{\hat{\sigma}_t}$, i.e. they are the ordinary residuals divided by their estimated standard deviation. If the fit is acceptable, the diagnostic plots should behave the same as those for an ARMA model[136]. In this particular case, since the $ARMA$ fit was acceptable, the $ARMA - GARCH$ fit is not substantially better.

In our previous discussion of ARMA fits, we discussed how the variance appeared to change over time for machine OP-110-0, as well as some machines that showed a similar pattern. In theory, ARMA-GARCH models should be able to provide more satisfactory fits for these machines.

Figure 4.19 contains the diagnostic plots for the ARMA-GARCH fit of OP110-0. As before, most of the plots are satisfactory. The plot of the residuals versus time is substantially better than for an ARMA fit, although it still appears to show some residual change in variance over time. This suggests that perhaps including higher-order GARCH terms may be valuable for this series. For a number of other machines, the picture is the same as for OP110-0, i.e. the ARMA-GARCH model provides some improvement but does not necessarily generate a perfect fit.

In other cases, the ARMA-GARCH model allows the model to compensate somewhat for data anomalies. An example of this is machine OP80-4, whose diagnostic plot is shown in Figure 4.20. The corresponding plot for an ARMA model is shown in Figure 4.21 Comparing the two figures, the anomalous residuals near the end of the series are less apparent in the GARCH version and the residual versus fitted value plot looks somewhat better. However, the GARCH terms were unable to fully capture the anomaly in the data.

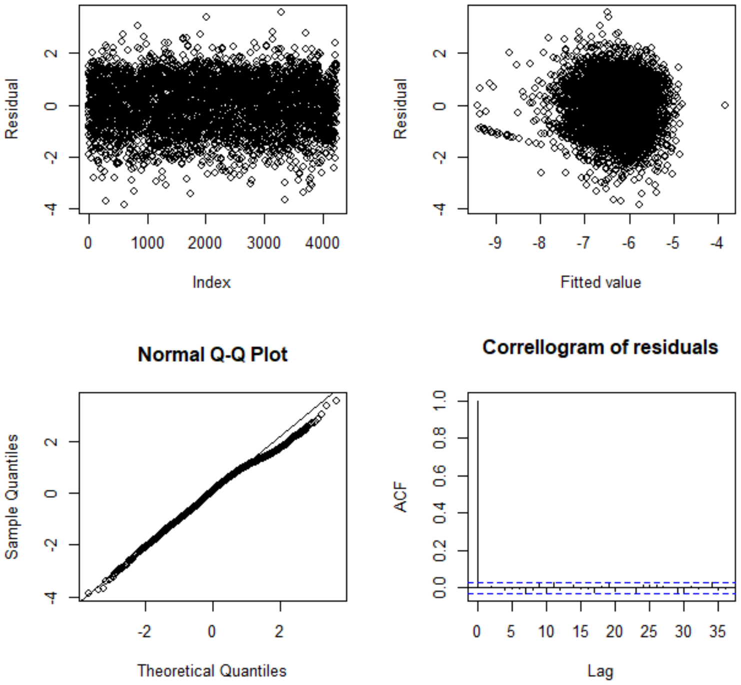


FIGURE 4.18: An example diagnostic plot for an ARMA-GARCH fit

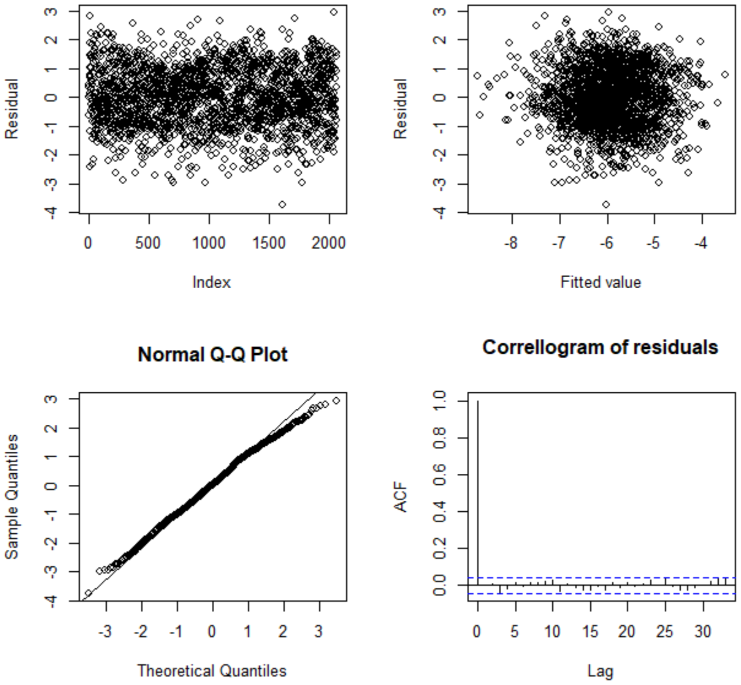


FIGURE 4.19: ARMA-GARCH diagnostic plots for machine OP110-0

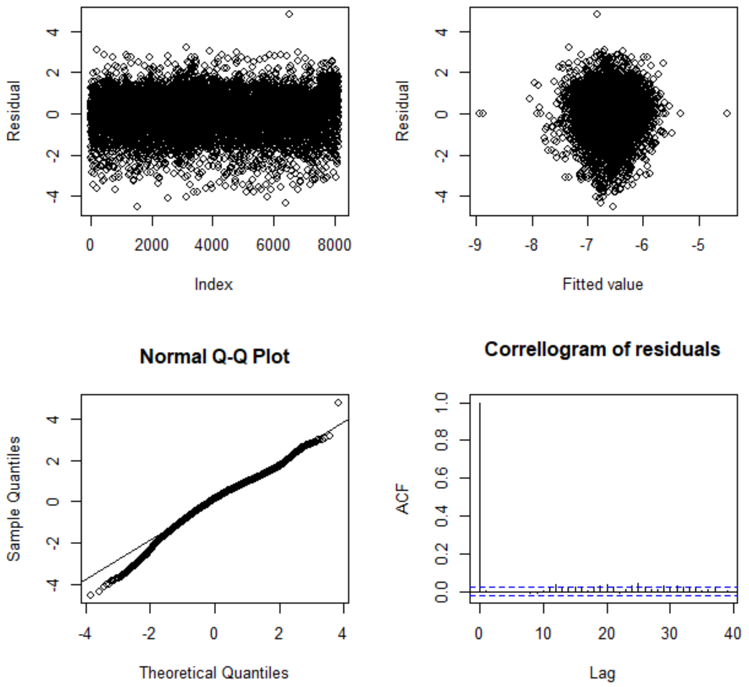


FIGURE 4.20: ARMA-GARCH diagnostic plots for machine OP80-4

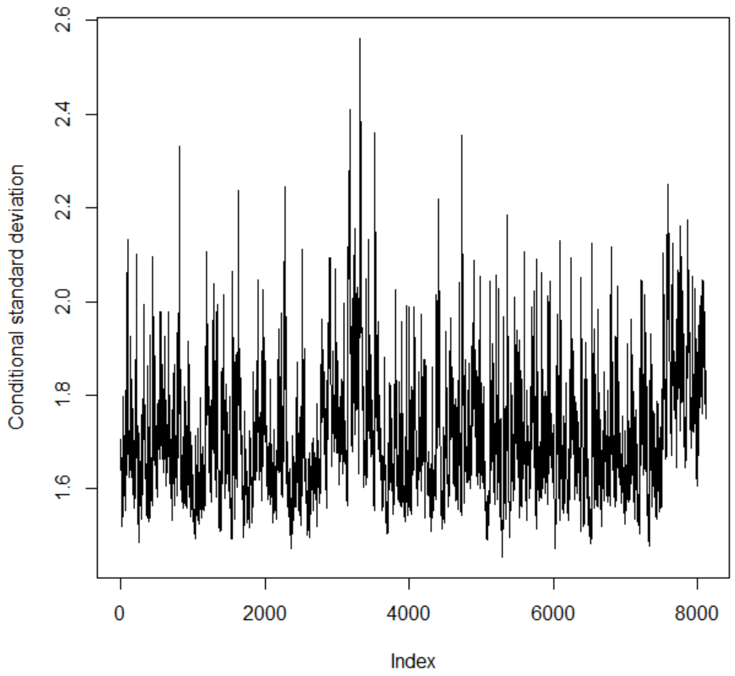


FIGURE 4.22: Conditional standard deviation for machine OP80-4

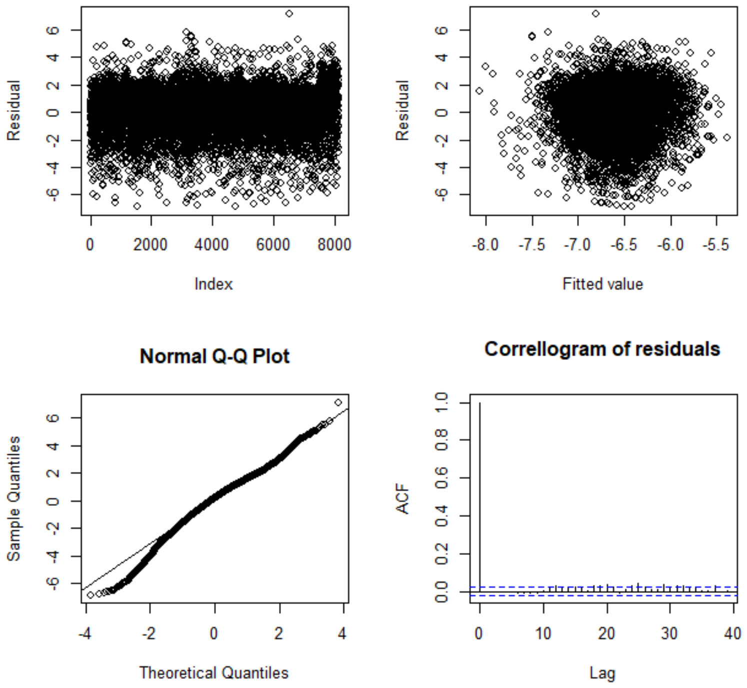


Figure 4.21: ARMA diagnostic plots for machine OP80-4

Another way to assess whether GARCH terms are having a significant impact on the model is to plot the conditional standard deviation of each observation. In an ARMA model without GARCH terms, this would be a constant. Significant changes in this value therefore indicate that the GARCH terms are substantially affecting the model.

Figures 4.22 and 4.23 show the conditional standard deviation for machines OP80-4 and OP110-0, respectively. The plot for OP80-4 shows that the standard deviation varies over the entire series, not only in the anomaly near the end. This suggests that the GARCH model is having an impact beyond just compensating for the anomalous data at the end of the series. For OP110-0, the standard deviation varies more smoothly, with some periods having substantially lower standard deviation.

To provide a more objective comparison between ARMA and ARMA-GARCH fits for the uptime data, we can consider the AIC value of each model. For 57 of the 77 machines we fitted the models to, the AIC of the ARMA-GARCH model was lower, indicating the GARCH terms improved the fit. In many cases, the improvement in AIC was small, but some machines showed more significant improvements. Diagnostic plots for all machines are provided for reference in Appendix B.

In all, both the AIC comparison and the diagnostic plots show that adding GARCH terms to the ARMA models can improve the fit to Ford's uptime data. Though the

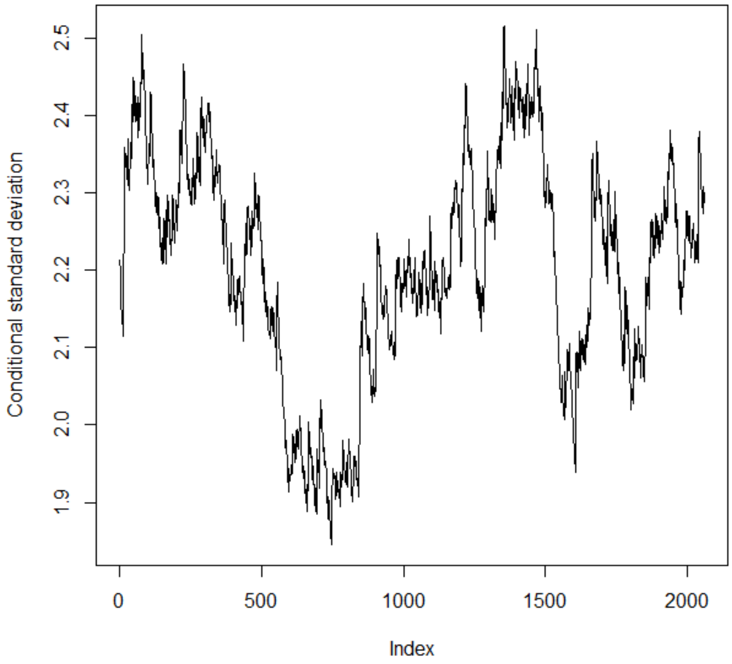


FIGURE 4.23: Conditional standard deviation for machine OP110-0

improvement in fit is small, some of the diagnostic plots we obtained suggest that higher-order GARCH terms might improve this situation.

The fitting results for ARMA-GARCH model and the AIC for both ARMA and ARMA-Garch models are illustrated in Table 4.1.

Table 4.1: Selected model order (p, q, r and s) and AIC for ARMA and ARMA-GARCH fits

| Type | Machine | P | Q | R | S | ARMA AIC | GARCH AIC |
|---|---|---|---|---|---|---|---|
| OP10 | 3 | 1 | 1 | 1 | 0 | 38204 | 38208 |
| OP 170_SCR | 0 | 3 | 2 | 1 | 1 | 111887 | 111879 |
| OP20 | 4 | 2 | 2 | 1 | 1 | 44903 | 44895 |
| OP 70 | 6 | 2 | 2 | 1 | 1 | 67278 | 67277 |
| OP20 | 10 | 3 | 1 | 1 | 2 | 58001 | 57992 |
| OP10 | 6 | 4 | 1 | 1 | 1 | 47863 | 47856 |
| DAG0160 | 0 | 1 | 1 | 1 | 2 | 7858 | 7856 |
| OP 40 | 0 | 1 | 1 | 1 | 0 | 22225 | 22230 |
| OP120 | 2 | 2 | 2 | 1 | 1 | 95997 | 95988 |
| OP10 | 2 | 1 | 2 | 1 | 1 | 26967 | 26957 |
| OP20 | 1 | 6 | 4 | 1 | 2 | 36846 | 36846 |

*Continued on next page*

Table 4.1 – *Continued from previous page*

| Type | Machine | P | Q | R | S | ARMA AIC | GARCH AIC |
|---|---|---|---|---|---|---|---|
| OP 140 | 0 | 2 | 4 | 1 | 1 | 31583 | 31568 |
| OP170_LEAK | 0 | 2 | 2 | 1 | 0 | 118374 | 118373 |
| OP20 | 5 | 3 | 1 | 1 | 1 | 48802 | 48765 |
| OP 125 | 0 | 1 | 2 | 1 | 0 | 91994 | 91975 |
| OP20 | 8 | 3 | 2 | 1 | 1 | 49855 | 49843 |
| OP10G | 0 | 1 | 1 | 2 | 1 | 38220 | 38205 |
| OP20 | 6 | 3 | 3 | 1 | 1 | 63717 | 63654 |
| OP130 | 2 | 1 | 2 | 1 | 2 | 35789 | 35757 |
| OP 150 | 0 | 3 | 3 | 1 | 1 | 33447 | 33445 |
| OP20 | 9 | 3 | 1 | 1 | 0 | 32042 | 32042 |
| OP 160 | 0 | 1 | 3 | 1 | 1 | 65133 | 65073 |
| OP 45A | 0 | 1 | 1 | 1 | 1 | 26753 | 26749 |
| OP 10A | 0 | 1 | 1 | 1 | 1 | 93780 | 93761 |
| OP20 | 11 | 2 | 2 | 1 | 1 | 74371 | 74343 |
| OP10A-30AG | 0 | 1 | 1 | 1 | 0 | 64588 | 64590 |
| OP130G | 0 | 1 | 1 | 1 | 0 | 63336 | 63335 |
| OP130 | 1 | 3 | 1 | 1 | 1 | 32649 | 32634 |
| OP10 | 4 | 3 | 2 | 1 | 0 | 32315 | 32320 |
| OP20 | 2 | 2 | 2 | 1 | 1 | 54014 | 53991 |
| OP 100 | 2 | 3 | 1 | 1 | 0 | 18750 | 18738 |
| OP20G | 0 | 2 | 1 | 1 | 1 | 65548 | 65550 |
| OP20 | 12 | 3 | 2 | 1 | 1 | 40432 | 40408 |
| OP20 | 7 | 4 | 4 | 2 | 1 | 45235 | 45239 |
| OP100G | 0 | 3 | 1 | 2 | 1 | 71513 | 71497 |
| OP20 | 3 | 3 | 3 | 1 | 0 | 41068 | 41069 |
| OP 110 | 0 | 3 | 1 | 1 | 1 | 39559 | 39549 |
| OP 135 | 0 | 6 | 0 | 1 | 1 | 86981 | 86970 |
| OP 30A | 0 | 1 | 1 | 1 | 1 | 27315 | 27310 |
| OP70 | 8 | 1 | 3 | 1 | 1 | 63160 | 63155 |
| OP80 | 9 | 3 | 3 | 1 | 0 | 47981 | 47984 |
| OP70 | 5 | 2 | 2 | 1 | 0 | 59973 | 59973 |
| OP70 | 10 | 3 | 2 | 1 | 2 | 68788 | 68779 |
| OP90 | 2 | 3 | 2 | 1 | 1 | 71787 | 71786 |
| OP80 | 5 | 3 | 1 | 1 | 0 | 53409 | 53416 |
| OP80G | 0 | 1 | 1 | 1 | 0 | 49306 | 49306 |
| OP30-40G | 0 | 1 | 4 | 1 | 0 | 32408 | 32414 |
| OP45-50C | 0 | 3 | 1 | 2 | 1 | 72207 | 72202 |
| OP90 | 9 | 1 | 2 | 1 | 2 | 91766 | 91761 |
| OP80 | 1 | 2 | 2 | 1 | 0 | 46251 | 46249 |

*Continued on next page*

Table 4.1 – *Continued from previous page*

| Type | Machine | P | Q | R | S | ARMA AIC | GARCH AIC |
|---|---|---|---|---|---|---|---|
| OP90 | 4 | 2 | 2 | 1 | 2 | 92467 | 92448 |
| OP55 | 0 | 4 | 0 | 1 | 1 | 29828 | 29820 |
| OP70 | 4 | 3 | 1 | 1 | 2 | 88706 | 88699 |
| OP90 | 5 | 2 | 2 | 1 | 1 | 75329 | 75323 |
| OP90 | 1 | 2 | 2 | 1 | 1 | 75707 | 75699 |
| OP70 | 3 | 3 | 3 | 1 | 1 | 55014 | 55010 |
| OP90 | 8 | 2 | 2 | 1 | 2 | 86907 | 86884 |
| OP80 | 6 | 2 | 2 | 1 | 0 | 47405 | 47407 |
| OP70 | 7 | 2 | 2 | 1 | 1 | 104101 | 104071 |
| OP70 | 12 | 3 | 1 | 1 | 2 | 91447 | 91433 |
| OP90 | 6 | 3 | 1 | 1 | 1 | 98229 | 98199 |
| OP80 | 2 | 3 | 3 | 2 | 1 | 42948 | 42952 |
| OP90G | 0 | 2 | 2 | 1 | 2 | 97238 | 97194 |
| OP70 | 2 | 2 | 2 | 1 | 1 | 84757 | 84728 |
| OP80 | 7 | 2 | 2 | 1 | 0 | 50823 | 50825 |
| OP90 | 10 | 3 | 1 | 1 | 1 | 82300 | 82300 |
| OP70 | 9 | 2 | 2 | 1 | 1 | 84415 | 84368 |
| OP80 | 3 | 3 | 1 | 2 | 1 | 43519 | 43511 |
| OP80 | 8 | 3 | 1 | 2 | 1 | 47125 | 47125 |
| OP70G | 0 | 1 | 3 | 1 | 1 | 71183 | 71185 |
| OP90 | 7 | 3 | 1 | 1 | 1 | 83292 | 83287 |
| OP90 | 12 | 3 | 1 | 1 | 1 | 82112 | 82109 |
| OP80 | 10 | 3 | 2 | 2 | 1 | 44993 | 45002 |
| OP80 | 4 | 9 | 0 | 1 | 2 | 140610 | 140517 |
| OP90 | 3 | 3 | 2 | 1 | 1 | 80999 | 81005 |
| OP50_BLTRD | 0 | 3 | 1 | 1 | 1 | 85899 | 85879 |
| OP70 | 1 | 2 | 2 | 1 | 1 | 77430 | 77416 |

### 4.5.5 Generating uptimes from an ARMA-GARCH model

As with ARMA models previously, we will now consider how we may generate uptimes from an ARMA-GARCH model. This allows Ford to use these models in simulation. The method presented here is that used in R's fGarch package[133].

Before we can start simulating from the model, we need to compute suitable initialisation values. The fGarch package uses the following approach to generate initial innovations $\epsilon$, standard deviations $h$ and observations $x$, using the notation from (4.10).

1. Generate $z$ according to a standard normal distribution.

2. Set $h = \frac{\gamma_0}{1-\sum_{i=1}^{s}\gamma_i-\sum_{i=1}^{r}\delta_i}$

3. Set $\epsilon = \sqrt{h}z$

4. Set $y = \frac{\mu}{1-\sum_{i=1}^{p}\beta_i}$

Given the initialisation values above, the conditional standard deviation at time $t$, $h_t$, the innovation $\epsilon_t$ and the observation $y_t$ are generated as follows:

$$h_t = \gamma_0 + \sum_{i=1}^{s}\gamma_i\epsilon_{t-i}^2 + \sum_{i=1}^{r}\delta_i h_{t-i} \tag{4.15}$$

$$\epsilon_t = \sqrt{h_t}z_t \tag{4.16}$$

$$y_t = \mu + \sum_{i=1}^{p}\beta_i y_{t-i} + \sum_{i=1}^{q}\alpha_i\epsilon_{t-i} \tag{4.17}$$

The equations above may be iterated as often as desired to generate the simulated observations of the ARMA-GARCH process. As with an ARMA process, the initial part of the sequence should be discarded. To obtain uptimes from the ARMA-GARCH observations, we again invert the data transformation.

## 4.6 Conclusion

In this chapter, we have discussed how ARMA models may be fitted to the uptimes of Ford's machines. For the vast majority of the machines, the ARMA models produce quite reasonable fits.

In one exceptional case, the residuals appear to be non-symmetrically distributed. This behaviour rules out ARMA models and any other models with symmetric error distributions.

For other machines, such as OP110-0, the residuals do appear to be symmetric, but their variance may change over time. In an attempt to improve these fits, we discussed fitting ARMA-GARCH models to Ford's data. In general, this led to a small improvement in the quality of fit. However, some of the diagnostic plots we obtained suggest that higher-order GARCH terms might lead to even better fits.

# Chapter 5

# Modelling uptimes using a Markov-Modulated Poisson Process

## 5.1 Introduction

In this chapter, we consider a different view of machine uptimes based on a Markov-Modulated Poisson Process (MMPP). In an MMPP, machine breakdowns occur with a finite number of different rates. This allows us to directly model machines that have high or low breakdown rates that vary over time. The change in breakdown rates over time also leads to positive autocorrelation between uptimes, as we have observed in Ford's data.

In the next section, we begin by reviewing the definition of the MMPP and some of its uses in the literature. We also consider how an MMPP may be fitted to an observed series of uptimes.

Thereafter, we apply the MMPP to Ford's data and discuss the results.

The chapter concludes with a brief summary of the MMPP, fitting MMPPs and our results in applying MMPPs to Ford's data.

## 5.2 The Markov-Modulated Poisson Process

### 5.2.1 Definition of the Markov-Modulated Poisson Process

The Markov-modulated Poisson process(MMPP) is an extension of the Poisson process, where the arrival rate varies between a finite number of possible values. As the MMPP is related to several other types of arrival processes and Markov Models, we will begin by briefly recalling these. Thereafter, we will define the MMPP itself and discuss its applications.

**Counting Process** A stochastic process $\{N(t), t \leq 0\}$ is a counting process if $N(t)$ represents the total number of events that have occurred up to time $t$. A counting process $N(t)$ must satisfy the following conditions[12]:

- $N(t) \geq 0$;
- $N(t)$ is integer valued;
- If $s < t$, then $N(s) \leq N(t)$;
- For $s < t$, $N(t) - N(s)$ equals the number of events that have occurred in the interval $(s, t]$.

Counting processes have been used to model arrivals, which are the occurrences of some types of events. For example, $N(t)$ is the number of customers who arrive at a shop by time $t$. Figure 5.1 shows a possible sample path of a counting process, where $T_1, T_2, T_3$ that represent the arrival times are the only sources of randomness.

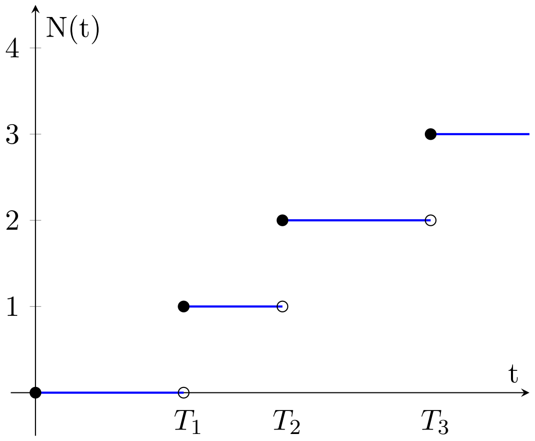


FIGURE 5.1: A sample path of a counting process

One of the most widely-used counting process is the Poisson process, which is usually used in cases where the occurrences of events appear to happen at a certain rate, but still at random. For instance, we know that the number of car accidents that happen in an area has a rate of 3 per month. Other than this information, the exact time of an accident is completely unknown. A Poisson process is defined as follows.

**Poisson Process** The counting process $\{N(t), t \geq 0\}$ is a Poisson process with rate $\lambda$, $\lambda > 0$ provided that the following axioms hold [12][30]:

- $N(0) = 0$
- for any $t, s \geq 0$, $N_{t+s} - N_t$ is independent of $\{N_u; u \leq t\}$;
- The number of events in any interval of length $t$ is Poison distributed with mean $\lambda t$. That is, for any $t, s \geq 0$,

$$P\{N(t+s) - N(s) = n\} = e^{-\lambda t}\frac{(\lambda t)^n}{n!}, n = 0, 1, \ldots \tag{5.1}$$

The Poisson processes are suitable to model many counting phenomena, but they are insufficient in some cases because the assumption of a deterministic arrival rate is unrealistic[19]. For instance, in financial data, people tend to observe bursts of trading activities followed by periods of quite activities[37]. Therefore, there is a non-constant behavior for the number of activities, and the Poisson model may not be suitable to model such a phenomena.

Cox[32] introduced the idea of doubly stochastic Poisson process, and it is studied in detail by Bartlett [10]. A doubly stochastic Poisson process is a generalization of a Poisson process where the arrival rate itself is a stochastic process. The aim of such a generalization is to allow an external process, which is called information process to influence the intensity of the occurrences of arrivals. A doubly stochastic process is also known as a Cox process, and is defined as follows.

**Doubly Stochastic Poisson Process** $\{N(t), t \geq 0\}$ is a doubly stochastic Poisson process with intensity process $\{\lambda(t, x(t)),\ t \geq 0\}$ if for almost every given path of the process $\{x(t), t \geq 0\}$ , $N\{.\}$ is a Poisson proses with intensity function $\{\lambda(t, x(t)),\ t \geq 0\}$. In other words, $\{N(t), t \geq 0\}$ is conditionally a Poisson process with intensity function $\{\lambda(t, x(t)),\ t \geq 0\}$ given $\{x(t), t \geq 0\}$[52].

As in Grandell[52], the process $\{x(t), t \geq 0\}$ in many applications conveys useful information. Therefore, $\{x(t), t \geq 0\}$ is called the information process. In simple cases, the information process is a finite collection of random variables independent of time. The applications of the doubly stochastic Poisson process are diverse when the parameters of interests are modeled as random variables instead of unknown but not random.

The Markov-Modulated Poisson Process (MMPP) is a special case of the doubly-stochastic Poisson process. In an MMPP, the information process $x(t)$ is a Markov process. Before we define the MMPP itself, we will briefly review Markov processes. The definition of a Markov process is taken from [30].

**Markov Processes** Let $S$ be a finite set of possible states. A Markov Process is a stochastic process $X_t$, such that, for all $t, s \geq 0$ and $j \in S$:

$$P(X_{t+s} = j | X_u; u \leq t) = P(X_{t+s} = j | X_t) \tag{5.2}$$

Intuitively, the value $X_t$ captures all the information about the process up to and including time $t$. In general, the right-hand side of (5.2) may depend on $t$. We will assume that this is not the case, and (5.2) may be rewritten as (5.3). Markov processes with this property are said to be *time-homogeneous*.

$$P(X_{t+s} = j | X_t = i) = P_s(i, j) \tag{5.3}$$

The function $P_s(i, j)$ defines a specific Markov process. A Markov Process may also be defined using a matrix of transition rates, called the *generator* of the process. The generator $Q$ has the following properties:

1. $Q_{ii} = -\lambda_i$, where $\lambda_i$ is the rate at which the process leaves state $i$
2. $Q_{ij} = \lambda_i P(i, j)$, where $P(i, j)$ is the probability that the process jumps in one step from state $i$ to state $j$

**Markov-Modulated Poisson Process** A Markov-Modulated Poisson Process(MMPP) is a doubly-stochastic Poisson process, whose information process is an $m$-state irreducible Markov process $X(t)$ [16]. Since $X(t)$ can take a finite number of values, the MMPP can be viewed as switching between a finite number of Poisson processes, based on the Markov process [43]. Whenever $X(t) = i$, that is, the Markov process is in state $i$, arrivals occur with the associated rate $\lambda_i$. In this chapter, we will assume that the MMPP is defined by the generator matrix $Q$ of its Markov process and the arrival rates $\lambda$.

The MMPP can also be seen as a Hidden Markov Model[51] where the the state is not directly visible, but the output, which are the event times that depend on the states, are visible. Expressing an MMPP as a Hidden Markov Model allows methods for the Hidden Markov Models to be used for calculating likelihood and estimating model parameters[16].

The application of the MMPP is mainly in queuing theory. For instance, Du[41] models the arrival process of a single-server queue as a three-level MMPP. Other authors, such as Latouche and Ramaswami[73], Lucantoni et al.[85], and Lucantoni[84] provide algorithms to solve the MMPP/G/1 queue to derive the queue statistics of interest. There are other interesting applications of the MMPP. Scott[119] uses a two-state MMPP to model fraud committed by a criminal on the telephone network. Davison and Ramesh[38] apply an MMPP to model the times of exposures to air pollution at a number of receptors in Western Europe.

### 5.2.2 Fitting an MMPP

We will now consider how an MMPP may be fitted to observed data. The method discussed here is based on the maximum-likelihood estimator (MLE). As shown by Rydén [113], the MLE is a consistent estimator for the MMPP.

The main difficulty in implementing the MLE for an MMPP is that the likelihood involves the state of the hidden markov chain, which must be integrated out. This leads to a difficult to maximise likelihood function. To avoid this problem, Rydén[113][114] proposed using EM algorithms to obtain the MLE.

Before we discuss the details of Rydén's EM algorithm from [114], we will very briefly review the general scheme of an EM algorithm. Consider a statistical model parameterised by the parameters $\phi$ and the likelihood function $L(\phi; y)$. Suppose further that there exist hidden variables $X$, whose values are $x$, that we cannot observe. Let $L^c(\phi; y, x)$ be the *complete likelihood*, that is, the likelihood function including the hidden values $x$.

To find the maximum of $L$, an EM algorithm proceeds as follows. First, we pick starting values $\phi_0$. Then, we iterate, replacing $\phi_0$ by $\hat{\phi}$, given by (5.4). As mentioned by Rydén [114], $L(\hat{\phi}; y) \geq L(\phi_0; y)$.

$$\hat{\phi} = argmax_{\phi} Q(\phi; \phi_0) \tag{5.4}$$

$$Q(\phi; \phi_0) = E_{\phi_0}\left(\log L^c(\phi; y, X) | y\right) \tag{5.5}$$

In the literature, evaluating (5.5) is known as the *E-step* of the algorithm and maximising (5.4) is known as the *M-step*. According to Rydén[114], the *M-step* often has a closed-form solution, while maximising the likelihood $L$ directly is difficult.

To state Rydén's EM algorithm for an MMPP, we require some more notation. First, let $Q$ be the generator of the MMPP's hidden chain, and assume the MMPP has $r$ states. Furthermore, let $\lambda_i$ be the arrival rate in state $i$, and let $\Lambda = diag\{\lambda_1, \lambda_2, \ldots, \lambda_r\}$. The parameter space for the MMPP is given by $\Phi = \{(Q, \Lambda) : Q \text{ is irreducible and } \lambda_i > 0 \text{ for at least one } i\}$.

Next, let $f$ be the transition density matrix of the MMPP, viewed as a Markov-Renewal process, given by (5.6). Then, the hidden Markov chain has transition matrix $P$, given by (5.7). We denote the probability density $\pi$ satisfying (5.8), by $\pi(Q, \Lambda)$. Furthermore, let $\bar{F}(t)$ be an $r \times r$ matrix, whose $ij^{th}$ element is the probability that an MMPP starting in state $i$ is in state $j$ at time $t$ and no arrivals occur in $[0, t]$. This matrix is given by (5.9).

$$f(y) = \exp\{(Q - \Lambda)y\}\Lambda \tag{5.6}$$

$$P = (\Lambda - Q)^{-1}\Lambda \tag{5.7}$$

$$\pi P = \pi \tag{5.8}$$

$$\bar{F}(t) = \exp\{(Q - \Lambda)t\} \tag{5.9}$$

Finally, for a particular sequence of uptimes $y_1, y_2, \ldots y_n$, let $t_0 = 0$ and $t_k = y_1 + \ldots + y_k$. With the notation discussed above, one step of the EM algorithm is performed using the pseudocode in Algorithm 2. In this algorithm, $f$ and $\bar{F}$ must be evaluated using the parameters $(Q_0, \Lambda_0)$. Further details of implementing this algorithm may be found in [114].

```
Set $L(0) = \pi^0$;
for $k \leftarrow 1$ to $n$ do
    Set $L(k) = L(k-1)f(y_k)$;
end
Set $R(n+1) = \mathbf{1}$;
for $k \leftarrow n$ to 1 do
    Set $R(k) = f(y_k)R(k+1)$;
end
for $i \leftarrow 1$ to $r$ do
    Set $B_i = 0$;
    for $j \leftarrow 1$ to $r$ do
        Set $A_{ij} = 0$;
    end
end
for $k \leftarrow 1$ to $n$ do
    Set $A_{ij} \leftarrow A_{ij} + L(k-1) \int_{t_{k-1}}^{t_k} \bar{F}(t - t_{k-1})\mathbf{1}_i\mathbf{1}_j^T f(t_k - t)dtR(k+1)$;
    Set $B_i \leftarrow B_i + L(k)R(k+1)$;
end
for $i \leftarrow 1$ to $r$ do
    The new estimate $\hat{\lambda}_i = \frac{B_i}{A_i i}$ for $j \leftarrow 1$ to $r$, such that $i \neq j$ do
        Update the estimates $\hat{q_{ij}} = q_{ij}^0 \frac{A_{ij}}{A_i i}$
    end
end
```

**Algorithm 2:** Rydén's[114] EM algorithm for the MLE of an MMPP

## 5.3 Fitting MMPPs to Ford’s data

We will now consider how an MMPP may be fitted to the uptimes of Ford’s machines. For each machine, we obtain the sequence of its uptimes and view a machine breakdown as an arrival. In this way, a machine may be viewed as an arrival process.

To fit an MMPP to the interarrival sequence, we have used the R package HiddenMarkov, which implements Rydén’s[114] EM algorithm. The EM algorithm was initialised using the method of Deng and Mark[39]. To select the order of the MMPP, model selection as performed using AIC, choosing an order between 2 and 10. The R code implementing these methods is seen below.

```
library(’HiddenMarkov’)

window.smooth <- function(x, r){
        res <- c()
        for(i in 1:length(x)){
                if(i < r+1){
                        nxt <- sum(x[1:i])/i
                } else if(i < length(x) - r){
                        nxt <- sum(x[(i-r):(i+r)])/(2*r+1)
                } else {
                        nxt <- sum(x[(i-r):(length(x))])/(length(x) - i + r + 1)
                }
                res <- c(res, nxt)
        }
        res
}

box.range <- function(x, N){
        m <- min(x)
        M <- max(x)
        m + (0:N)*(M - m)/N
}

cluster.range.iter <- function(y, b){
        totals <- vector(mode="numeric", length(b)-1)
        counts <- vector(mode="numeric", length(b)-1)
        for(k in 2:(length(b))){
                in_bounds <- y >= b[k-1] & y < b[k]
                counts[k-1] <- sum(in_bounds)
                totals[k-1] <- sum(y[in_bounds])
        }
        next_b <- b
        numer <- totals[1:(length(totals)-1)] + totals[2:length(totals)]
        denom <- counts[1:(length(counts)-1)] + counts[2:length(counts)]
        next_b[2:(length(b)-1)] <- numer/denom
        next_b
}

cluster.range <- function(y, N, tol=1e-3) {
  b <- box.range(y, N)
  while(T){
    next.b <- cluster.range.iter(y,b)
        if(sum(abs(next.b - b) < tol))
                break;
  }
  next.b
}

stationary <- function(Q, lambda){
  Lambda <- diag(lambda)
  P <- solve(Lambda - Q, Lambda)
  A <- rbind(diag(length(lambda)) - t(P), 1)
  x <- c(vector(mode="numeric", length(lambda)), 1)
  qr.solve(A, x)
}

params.from.range <- function(y, b){
  N <- length(b) - 1
  assignments <- vector(mode="numeric", length(y))
  lambda <- vector(mode="numeric", N)
  Q <- matrix(0, N,N)
  for(i in 1:N){
        in.i <- y >= b[i] & y < b[i+1]
        assignments[in.i] <- i
        lambda[i] <- sum(in.i) / sum(y[in.i])
  }
  truncy <- y[1:(length(y)-1)]
  truncated <- assignments[1:(length(y)-1)]
  shifted <- assignments[2:length(y)]
  for(i in 1:N){
        for(j in 1:N){
                if(i != j){
                        in.scope <- truncated == i & shifted == j
                        Q[i,j] <- sum(in.scope)/sum(truncy[in.scope])
                }
        }
        Q[i,i] <- -sum(Q[i,])
  }
  delta <- tryCatch(stationary(Q, lambda), error = function (e){rep.int(1.0/
    length(lambda),length(lambda))})
  list(lambda = lambda, Q = Q, delta = delta)
}

mmppfit <- function(upt, k){
        supt <- window.smooth(upt, 2)
        b <- cluster.range(supt, k)
        parms <- params.from.range(supt, b)
        m0 <- mmpp(cumsum(upt), parms$Q, parms$delta, parms$lambda, F)
        bwc <- bwcontrol(prt=F)
        BaumWelch(m0, bwc)
}
for(l in levels(rejected$Type)){
    max.n <- max(rejected$Number[rejected$Type==l])
    if(max.n >= 0){
        for( n in 0:(max(rejected$Number[rejected$Type==l]))){
            upt <- get.machine(durs, l, n)
            if(length(upt) > 1){
                min.AIC <- Inf
                best.model <- 0
                for( order in 2:max.mmpp){
                    model <- tryCatch(mmppfit(upt, order), error= function(e)
    list(LL=-Inf))
                    aic <- 2*order^2 - 2*model$LL
                    if(aic < min.AIC){
                        min.AIC <- aic
                        rejected$MMPPorder[rejected$Type == l & rejected$Number
    == n] <- order
                        rejected$MMPPaic[rejected$Type == l & rejected$Number ==
    n] <- aic
                        best.model <- model
                    }
                }
                best.model$delta <- best.model$delta / sum(best.model$delta)
                simmed <- simulate(best.model, nsim=10*length(upt))
                sim.upt <- simmed$tau[2:length(simmed$tau)] - simmed$tau[1:(
    length(simmed$tau)-1)]
                plotassessfit(upt, sim.upt, paste('mmpp-acf-', l, '-', n, '.png',
     sep=''),
                              paste('mmpp-qq-', l, '-', n, '.png', sep=''))
                realified <- cliptoreal(sim.upt)
                plotassessfit(toreal(upt),
                              realified,
                              paste('mmpp-acf-trans-', l, '-', n, '.png', sebcop=
    ''),
                              paste('mmpp-qq-trans-', l, '-', n, '.png', sep=''))
                }
        }
    }
}
```

Listing 5.1: Fitting MMPPs in R

To illustrate the EM algorithm's behaviour, we will now briefly look at its convergence in one particular case, machine OP130.1. In this example, we will use an MMPP with 2 states. Figure 5.2 shows the log-likelihood of the MMPP versus the number of iterations performed. Figure 5.3 shows the arrival rates of the Poisson process in the two states. Both figures indicate that the method makes rapid progress initially, but takes many iterations for the convergence criterion to be met. This suggests that computational effort may be saved by relaxing the convergence criterion, as many of the iterations performed have little effect.

For Ford's machines, the procedure discussed above chose MMPP orders of 2 or 3 only, never using a larger MMPP. This could indicate that the larger MMPPs do not provide a significant enough improvement in the fit to justify their extra parameters. However, it is equally possible that the EM algorithm failed to find the maximum of the likelihood for larger models.

To assess the quality of the MMPP's fit to the uptime data, we will consider a series of diagnostic plots. Unlike with ARMA models, there is no suitable notion of a "residual" for an MMPP. As such, the diagnostic plots available are somewhat limited. The plots we will use are based on simulating a long sequence of uptimes from the fitted MMPP. For all the plots we will consider, we've taken a "long" sequence to mean ten times as long as the observed uptime series.

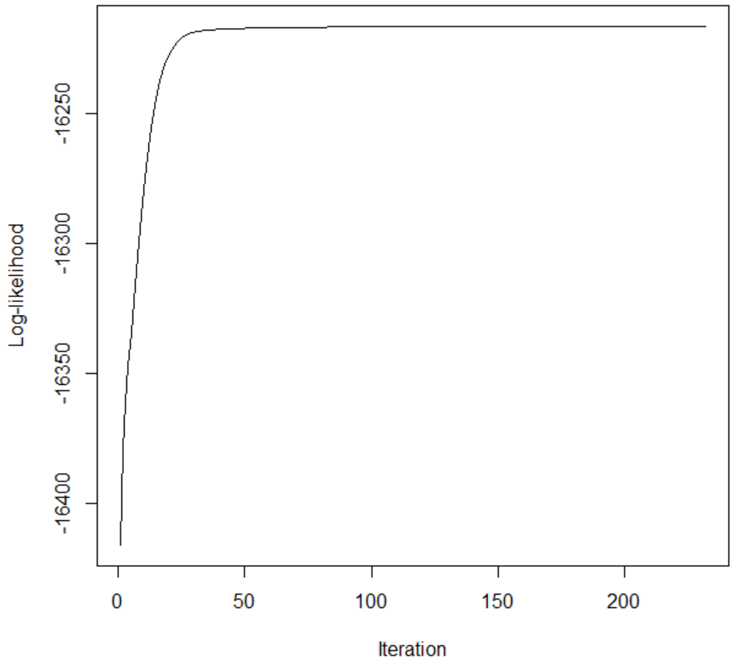


FIGURE 5.2: Log-likelihood of the MMPP versus the number of EM iterations performed, for machine OP130-1, using a 2-state MMPP

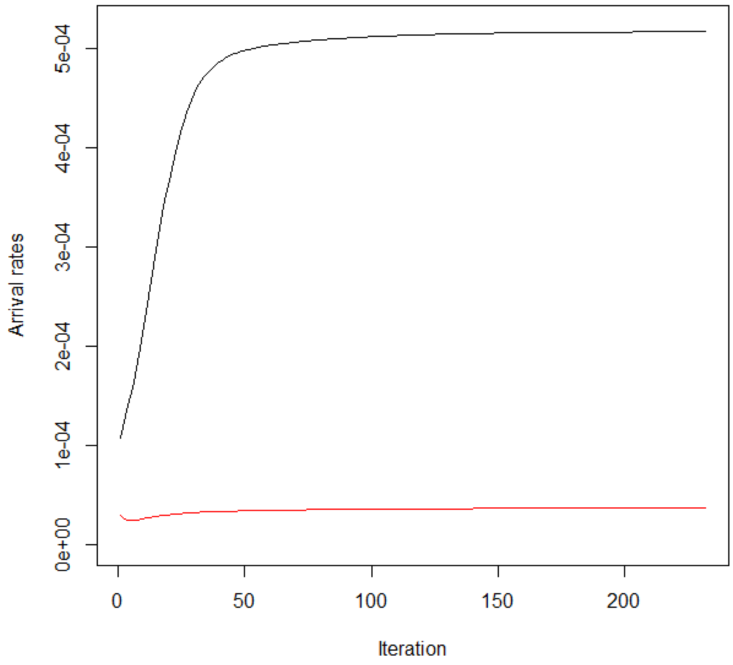


FIGURE 5.3: Estimated arrival rates of the MMPP versus the number of EM iterations performed, for machine OP130.1, using a 2-state MMPP

The first plot we use is an autocorrelation plot, as shown in Figure 5.4. In this figure, the black line is the sample autocorrelation of the original sequence. The blue line is estimated from the simulated uptimes. If the MMPP fits the data well, we'd expect the blue line to be similar to the black line, but somewhat smoother. In this particular case, the fit looks quite reasonable.

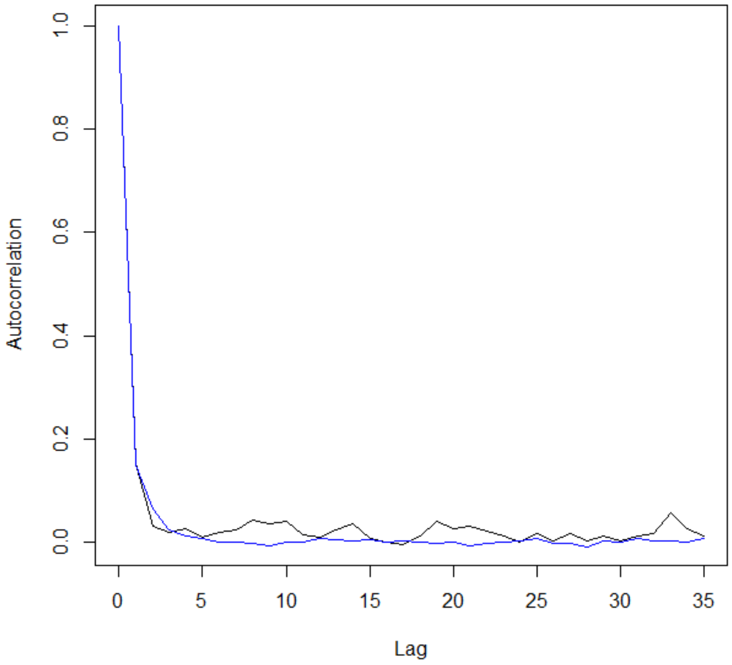


FIGURE 5.4: The autocorrelation plot for the simulated (blue) and observed (black) uptimes of machine OP70-4

The second plot we use is a QQ-plot of the uptimes from the original data against the simulated uptimes. This plot allows us to check whether the model reproduces a similar distribution of uptimes. An example of this plot is shown in Figure 5.5. In this figure, we'd expect to see the dots, representing the data, to match the black line if the fit is perfect. In this particular case, the data matches the line reasonably well.

We will now consider a number of machines and assess the fit of the MMPP for them, using the diagnostic plots explained above. In the analysis, we will consider only machines where the ARMA fits were unsatisfactory.

The first machine we consider is OP90G-0. The diagnostic plots in Figures 5.6 and 5.7 look satisfactory. That said, there are some outliers in the QQ plot that suggests the model may not be able to capture the extremes of the data's tail behaviour.

Next, we consider machine OP80G-0, whose diagnostic plots are given in Figures 5.8 and 5.9. The ACF plot looks reasonable enough, but the QQ plot shows significant deviation

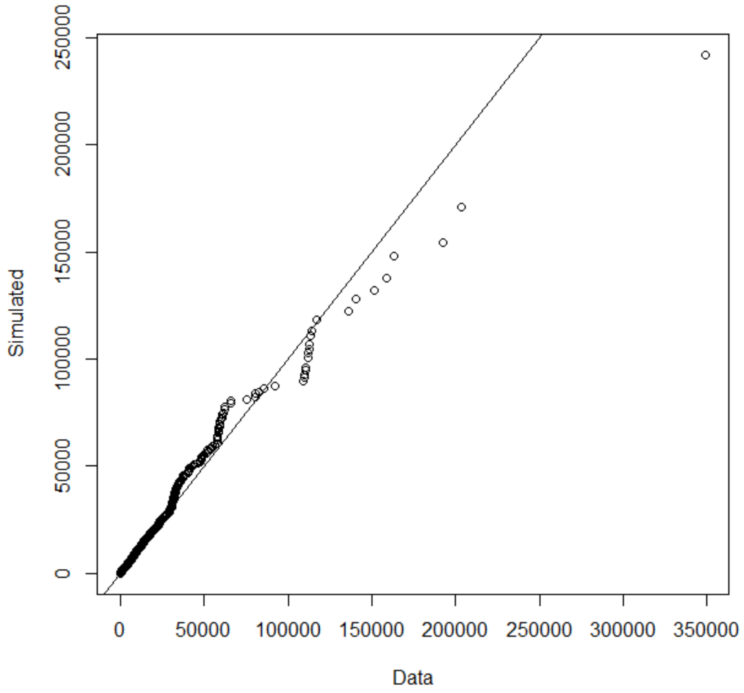


Figure 5.5: QQ-plot of simulated versus observed uptimes for machine OP70-4

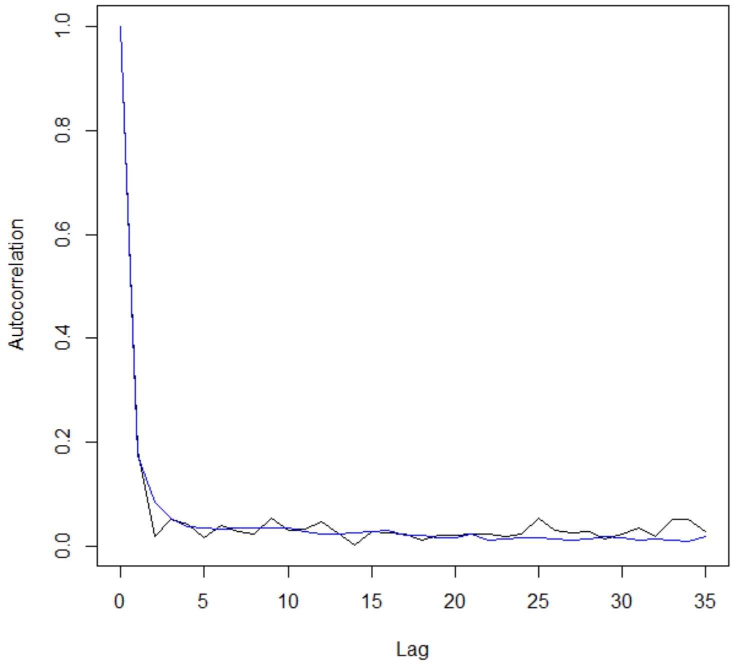


Figure 5.6: The autocorrelation plot for the simulated (blue) and observed (black) uptimes of machine OP90G-0

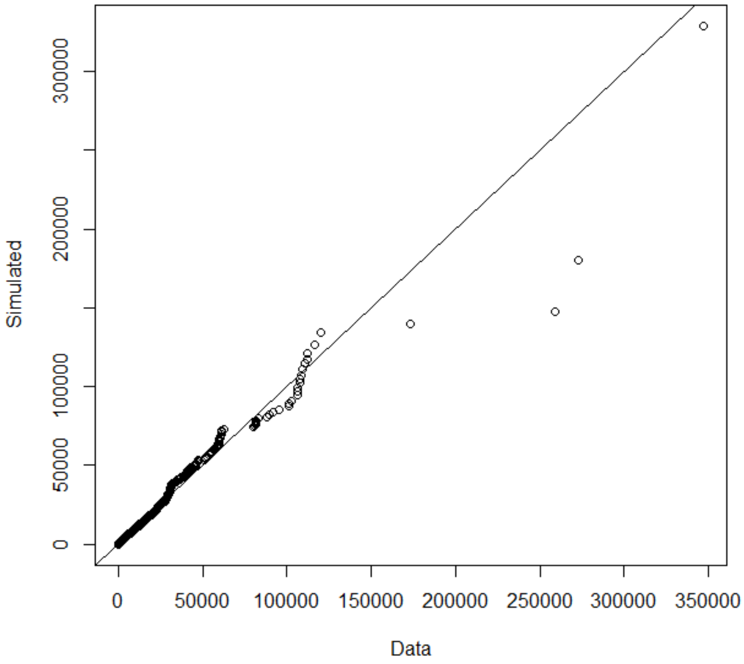


FIGURE 5.7: QQ-plot of simulated versus observed uptimes for machine OP90G-0

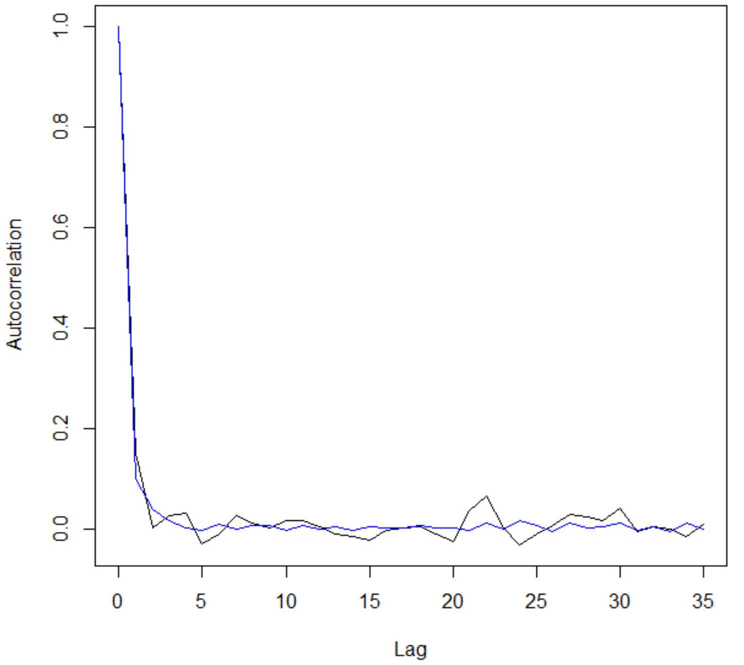


FIGURE 5.8: The autocorrelation plot for the simulated (blue) and observed (black) uptimes of machine OP80G-0

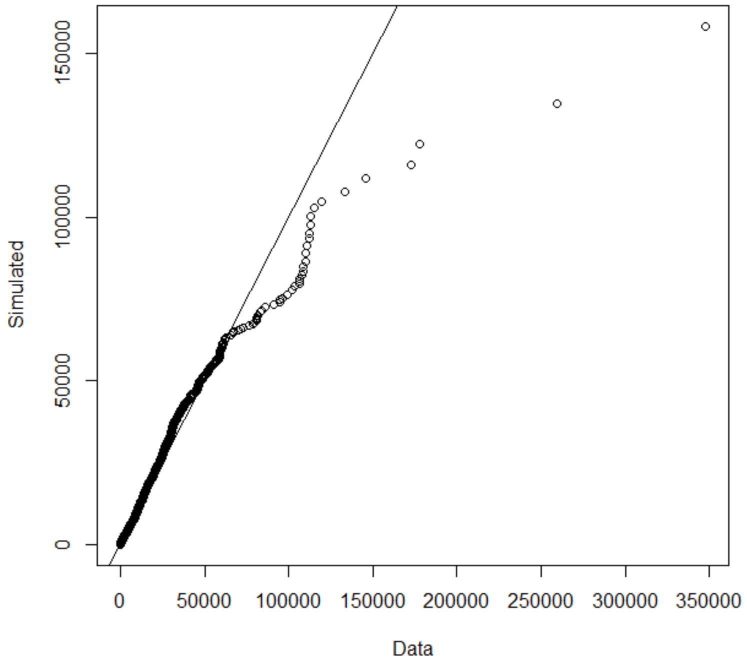


Figure 5.9: QQ-plot of simulated versus observed uptimes for machine OP80G-0, based on the MMPP

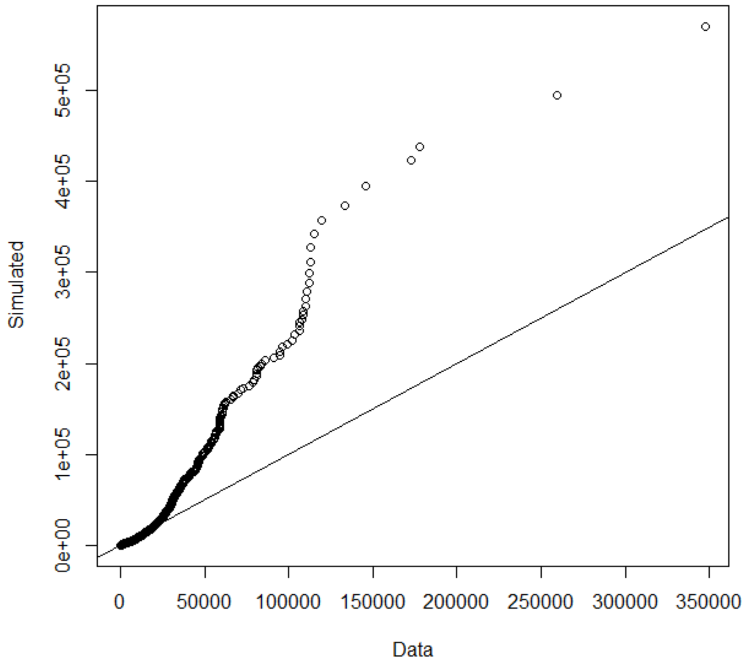


Figure 5.10: QQ-plot of simulated versus observed uptimes for machine OP80G-0, based on an ARMA-GARCH model

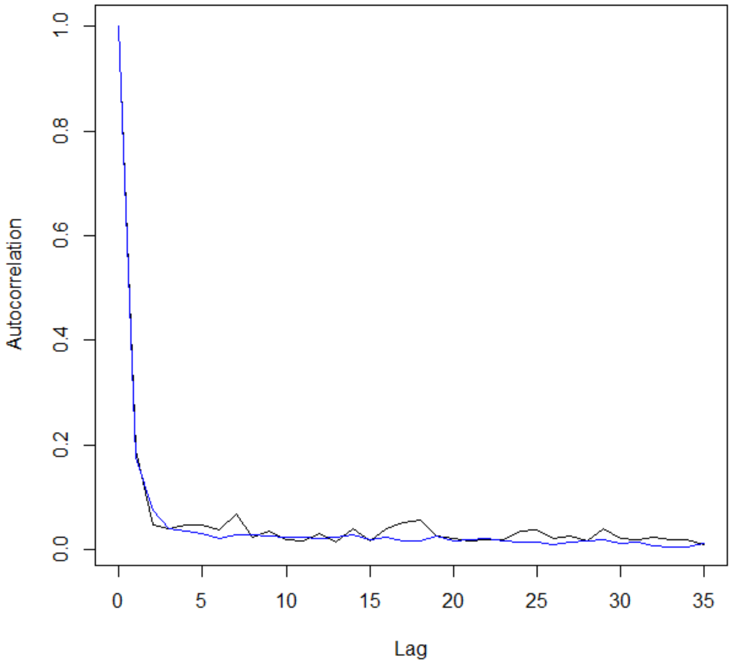


Figure 5.11: The autocorrelation plot for the simulated (blue) and observed (black) uptimes of machine OP80-4

in the tail. Comparing this figure to the same plot for a series simulated for an ARMA model, Figure 5.10, shows that the MMPP captures the observed distribution far better than the ARMA model.

For the next machine, OP80-4, Figures 5.11 and 5.12 show that again, the MMPP captures the ACF reasonably well, but the distribution's tail behaviour is not captured as well. As can be seen in Figures 5.13 and 5.14, the same is true for machine OP55-0.

Next, we consider machine OP20-5. For this machine, the ACF plot, Figure 5.15 shows the MMPP understates the autocorrelation. However, this does not appear to affect the fit to the distribution, as shown in Figure 5.15. The corresponding plots for an ARMA model, Figures 5.17 and 5.18, show that the ARMA model also understates the autocorrelation. Moreover, the ARMA model's fit to the distribution is substantially worse than the MMPP's.

Finally, we consider machine OP130-1, whose diagnostic plots are shown in Figures 5.19 and 5.20. These plots look rather similar to those for OP20-5 and show similar behaviour for the MMPP.

The diagnostic plots shown above indicate that the MMPP generally captures the autocorrelation of the uptimes reasonably well and tends to capture the distribution of uptimes better than an ARMA model. However, the comparison so far has been rather

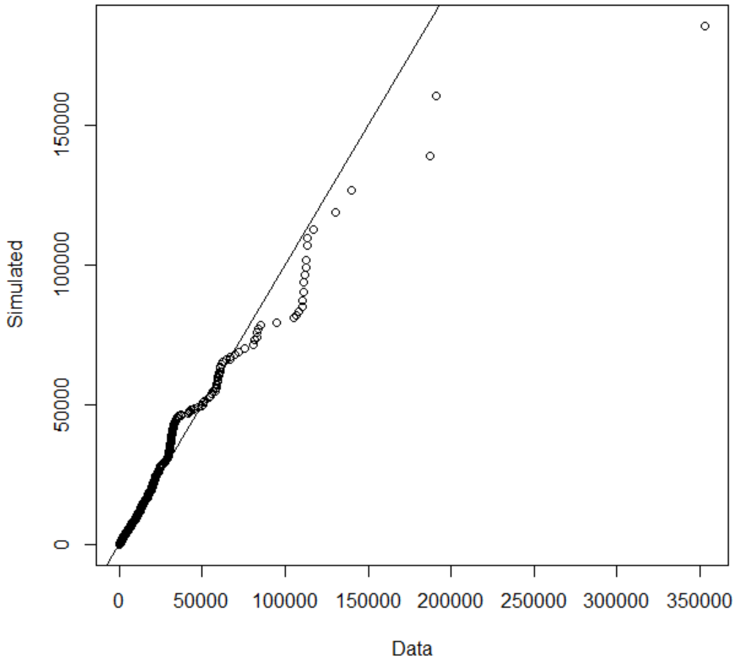


Figure 5.12: QQ-plot of simulated versus observed uptimes for machine OP80-4

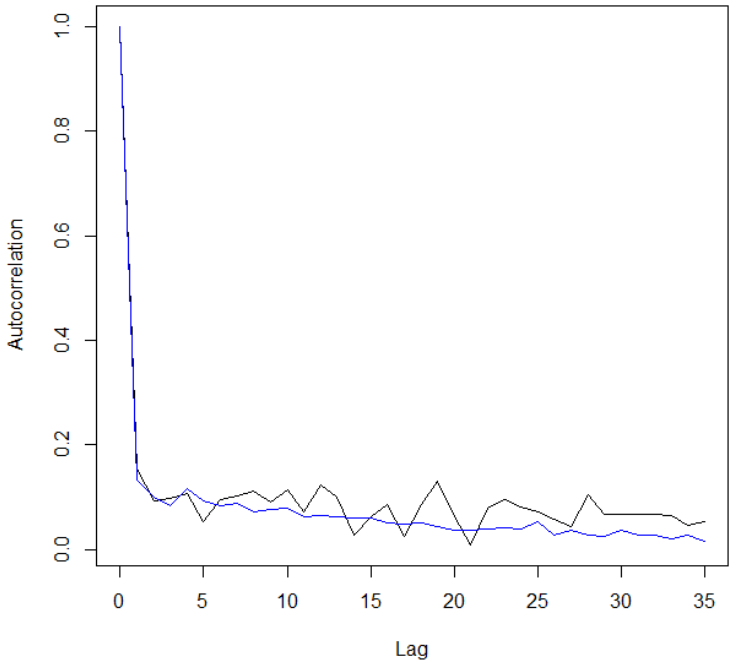


Figure 5.13: The autocorrelation plot for the simulated (blue) and observed (black) uptimes of machine OP55-0

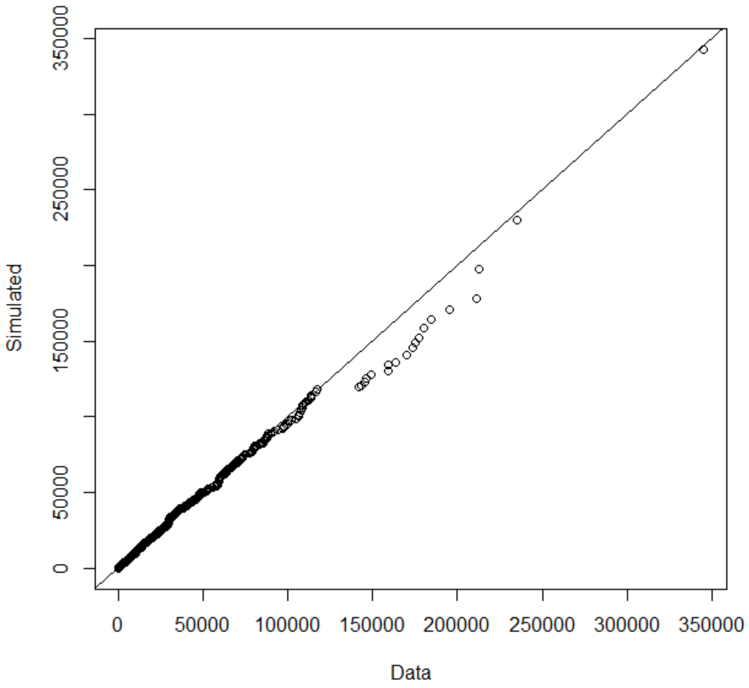


Figure 5.14: QQ-plot of simulated versus observed uptimes for machine OP55-0

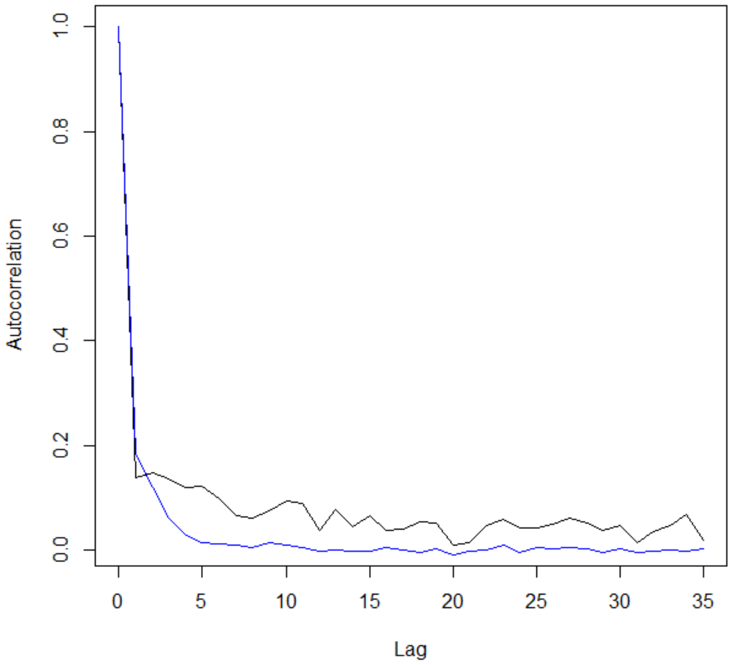


Figure 5.15: The autocorrelation plot for the simulated (blue) and observed (black) uptimes of machine OP20-5, for an MMPP

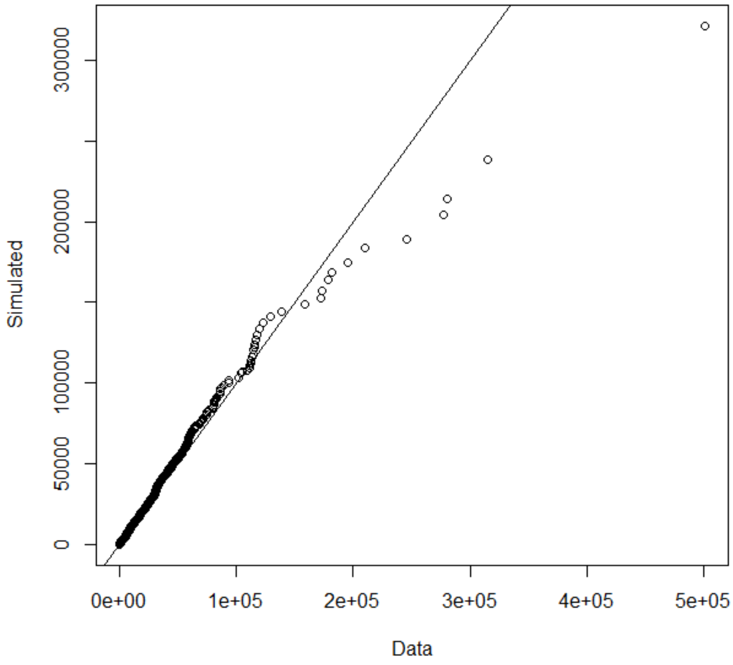


FIGURE 5.16: QQ-plot of simulated versus observed uptimes for machine OP20-5, for an MMPP

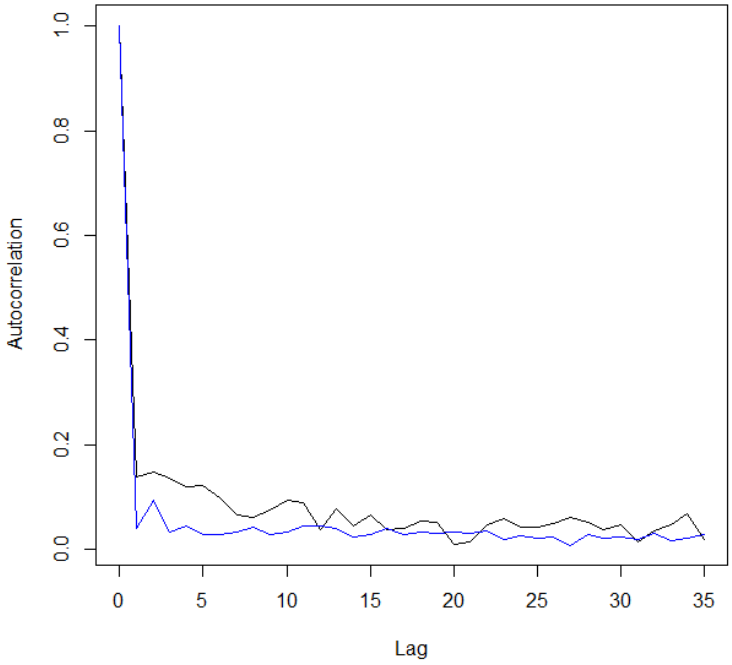


FIGURE 5.17: The autocorrelation plot for the simulated (blue) and observed (black) uptimes of machine OP20-5, for an ARMA-GARCH model

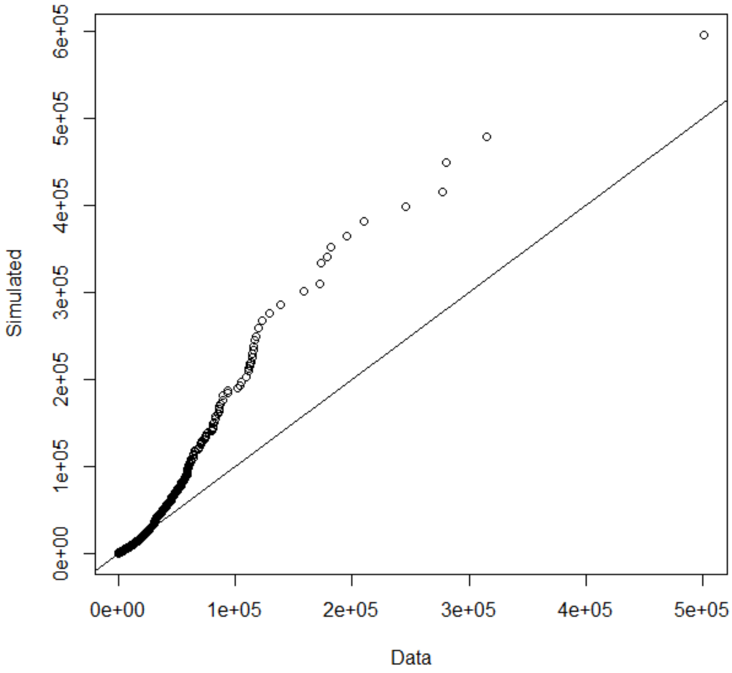


FIGURE 5.18: QQ-plot of simulated versus observed uptimes for machine OP20-5, for an ARMA-GARCH model

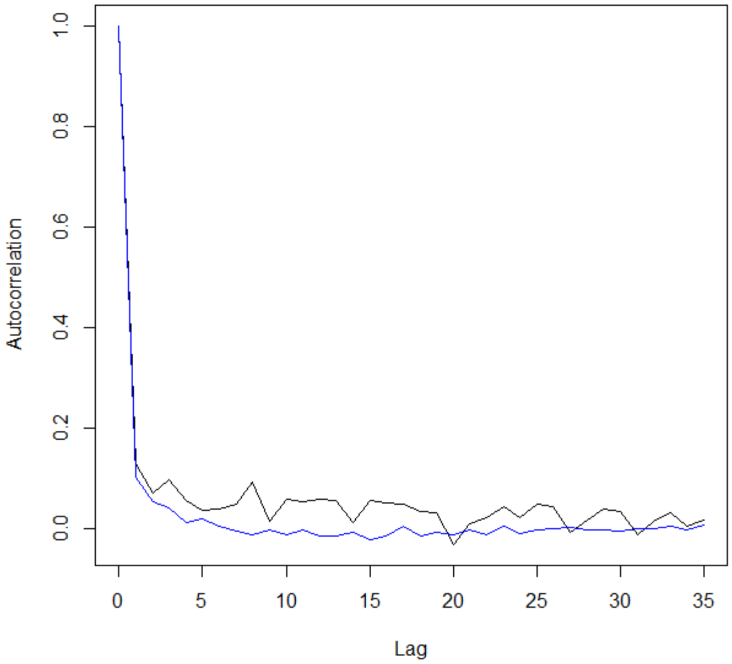


FIGURE 5.19: The autocorrelation plot for the simulated (blue) and observed (black) uptimes of machine OP130.1

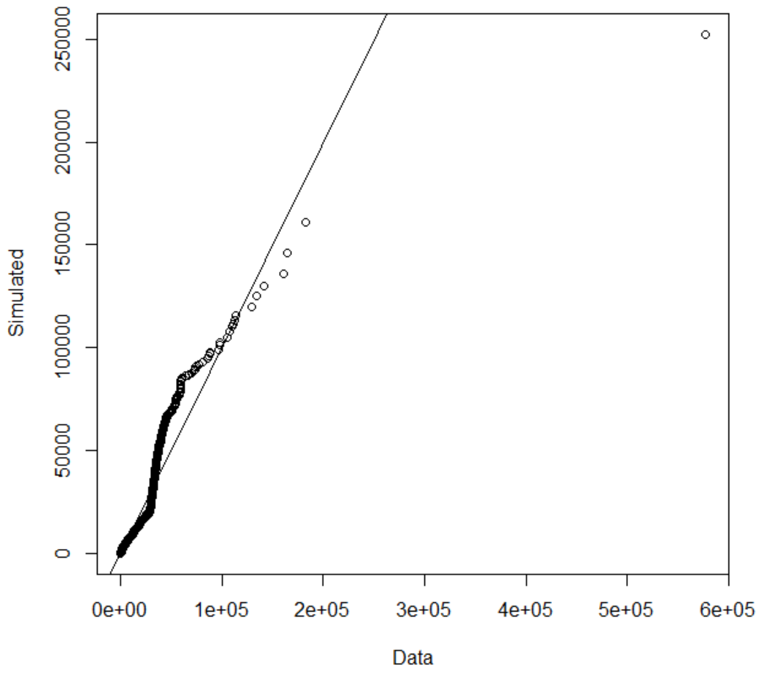


FIGURE 5.20: QQ-plot of simulated versus observed uptimes for machine OP130.1

subjective. To provide a more objective comparison, we need to compare the AIC values of the MMPP to the ARMA and ARMA-GARCH models obtained in the previous chapter. However, as the ARMA models were based on a data transformation, the AIC values need to be adjusted. For this particular data transformation, a change of variables gives the adjustment (5.10), where $AIC_{ARMA}$ is the AIC of the ARMA model on transformed data. Here, $y_i$ is the $i$th uptime and $a$ and $b$ are the lower and upper thresholds used in removing outliers, respectively.

$$AIC = AIC_{ARMA} - 2\sum_{i=1}^{n} \log\left(\frac{b-a}{(y_i - a)(b - y_i)}\right) \tag{5.10}$$

For the uptimes of Ford's machines, (5.10) is almost always smaller than the AIC of the MMPP. The exceptions are machines OP130-1 and OP55. This indicates that while the diagnostic plots indicate the MMPP may capture some of the uptimes' behaviour better than an ARMA or ARMA-GARCH model, this does not necessarily result in a lower AIC value. For reference, fitting results are in Table 5.1, and diagnostic plots for all machines are provided in Appendices C.

Table 5.1: Selected model order and AIC for MMPP fits

| Type | Machine | Order | AIC |
|---|---|---|---|
| OP10 | 3 | 3 | 38275 |
| OP 170_SCR | 0 | 3 | 113679 |
| OP20 | 4 | 2 | 45937 |
| OP 70 | 6 | 3 | 68355 |
| OP20 | 10 | 2 | 59154 |
| OP10 | 6 | 2 | 48421 |
| DAG0160 | 0 | 2 | 8024 |
| OP 40 | 0 | 3 | 22301 |
| OP120 | 2 | 2 | 98644 |
| OP10 | 2 | 2 | 27430 |
| OP20 | 1 | 2 | 37281 |
| OP 140 | 0 | 2 | 31702 |
| OP170_LEAK | 0 | 2 | 120616 |
| OP20 | 5 | 3 | 50059 |
| OP 125 | 0 | 2 | 93025 |
| OP20 | 8 | 2 | 50783 |
| OP10G | 0 | 2 | 38689 |
| OP20 | 6 | 3 | 65002 |
| OP130 | 2 | 2 | 35979 |
| OP 150 | 0 | 3 | 33797 |
| OP20 | 9 | 2 | 32349 |
| OP 160 | 0 | 3 | 66065 |
| OP 45A | 0 | 2 | 27278 |
| OP 10A | 0 | 2 | 94745 |
| OP20 | 11 | 3 | 75918 |
| OP10A-30AG | 0 | 3 | 65678 |
| OP130G | 0 | 3 | 63593 |
| OP130 | 1 | 2 | 32442 |
| OP10 | 4 | 2 | 32485 |
| OP20 | 2 | 2 | 55219 |
| OP 100 | 2 | 2 | 18919 |
| OP20G | 0 | 3 | 66501 |
| OP20 | 12 | 2 | 41295 |
| OP20 | 7 | 2 | 45909 |
| OP100G | 0 | 3 | 72695 |
| OP20 | 3 | 2 | 41701 |
| OP 110 | 0 | 3 | 40551 |
| OP 135 | 0 | 2 | 88465 |

*Continued on next page*

Table 5.1 – *Continued from previous page*

| Type | Machine | Order | AIC |
|---|---|---|---|
| OP 30A | 0 | 2 | 27818 |
| OP70 | 8 | 2 | 64643 |
| OP80 | 9 | 2 | 48683 |
| OP70 | 5 | 2 | 61201 |
| OP70 | 10 | 2 | 70302 |
| OP90 | 2 | 3 | 73022 |
| OP80 | 5 | 2 | 54569 |
| OP80G | 0 | 2 | 49548 |
| OP30-40G | 0 | 3 | 32630 |
| OP45-50C | 0 | 3 | 73223 |
| OP90 | 9 | 3 | 93575 |
| OP80 | 1 | 2 | 46979 |
| OP90 | 4 | 2 | 94975 |
| OP55 | 0 | 3 | 29801 |
| OP70 | 4 | 3 | 90772 |
| OP90 | 5 | 2 | 77245 |
| OP90 | 1 | 3 | 77002 |
| OP70 | 3 | 2 | 56018 |
| OP90 | 8 | 2 | 88534 |
| OP80 | 6 | 3 | 48028 |
| OP70 | 7 | 2 | 106973 |
| OP70 | 12 | 3 | 93498 |
| OP90 | 6 | 3 | 100939 |
| OP80 | 2 | 3 | 43418 |
| OP90G | 0 | 3 | 97637 |
| OP70 | 2 | 3 | 86662 |
| OP80 | 7 | 3 | 51647 |
| OP90 | 10 | 2 | 84259 |
| OP70 | 9 | 2 | 86724 |
| OP80 | 3 | 3 | 44410 |
| OP80 | 8 | 2 | 47955 |
| OP70G | 0 | 2 | 72267 |
| OP90 | 7 | 2 | 85535 |
| OP90 | 12 | 2 | 83975 |
| OP80 | 10 | 3 | 45546 |
| OP80 | 4 | 3 | 143337 |
| OP90 | 3 | 2 | 82937 |
| OP50_BLTRD | 0 | 2 | 87217 |

*Continued on next page*

Table 5.1 – *Continued from previous page*

| Type | Machine | Order | AIC |
|---|---|---|---|
| OP70 | 1 | 3 | 79517 |

## 5.4 Simulating MMPPs

In the previous section, we used simulated data from MMPPs to validate their fit to Ford's uptime data. Furthermore, Ford need to be able to generate simulated uptimes to include MMPPs in their simulated production lines. We will now discuss how these uptimes may be generated. The discussion here is based on the approach used by R's HiddenMarkov package. We split the generation process into the following parts:

1. Generating an initial state
2. Generating sojourn times in every state
3. Generating arrivals during the sojourn time

**Initialising the simulation** To initialise the simulation, we randomly generate a starting state according to the stationary distribution of the MMPP's hidden chain. This gives us an initial state $Y_0 = k$ at time zero.

**Generating sojourn times** Whenever the MMPP moves to a particular state, we need to generate the duration the MMPP will spend in this state. To do so, we use the fact that $-Q_{ii}$ is the total rate at which the hidden chain leaves state $i$. Therefore, if the MMPP is in state $Y_i$ at time $X_i$, the time spent in state $Y_i$ is distributed $Exp(-Q_{Y_iY_i})$. Drawing from this distribution gives us the time the process will leave state $Y_i$, as $X_{i+1} = X_i + Exp(-Q_{Y_iY_i})$. The state $Y_{i+1}$ may be drawn from the set of possible states, with $P(Y_{i+1} = j) = \frac{Q_{Y_ij}}{-Q_{Y_iY_i}}$.

**Generating arrivals during the sojourn time** Having simulated the time the MMPP will spend in state $Y_i$, we need to generate the arrivals that will occur during this time. To do so, we initialise $T_{i0} = X_i$, and draw inter-arrival times $A_{ij}$ from $Exp(\lambda_i)$. Recursively, we set $T_{ij} = T_{i(j-1)} + A_{ij}$ and draw inter-arrival times until $T_{ij} > X_{i+1}$. As long as $T_{ij} \leq X_{i+1}$, an arrival is generated at time $T_{ij}$.

## 5.5 Conclusion

In this chapter, we have discussed the application of Markov-Modulated Poisson Processes (MMPPs) to the uptimes of Ford's machines. We have also presented some model diagnostics, which indicated that the MMPP generally captured the uptimes' autocorrelation well and was substantially better than an ARMA model at capturing the uptimes' distribution. However, comparing AIC values to those obtained for ARMA-GARCH models in the previous chapter showed that, despite these advantages of the MMPP, ARMA-GARCH models are still preferred in most cases. The exceptions, machines OP130.1 and OP55, were particularly hard to fit with ARMA-GARCH models.

In all, our results indicate that the MMPP may be a good alternative to ARMA-GARCH models if the distribution of the uptimes needs to be captured exactly. In addition, an MMPP may provide a better fit, as measured by AIC, when an ARMA-GARCH model fits poorly.

# Chapter 6

# Application to Simulation at Ford

## 6.1 Introduction

Previously, we discussed how the uptimes may be predicted for a single machine, using either an ARMA-GARCH or an MMPP model. In this chapter, we discuss how this technique may be used in the context of a large-scale simulation model of a real-world production line, as used at Ford.

A production line at Ford consists of multiple workstations. Each workstation has either a single machine or a number of parallel machines. Workstations are connected by conveyors in a complex network.

In the next section, we begin by discussing the role of uptimes in Ford's current simulation model. We then propose a method to extend this, using the simulation approaches we have developed in Chapters 4 and 5. We conclude this chapter with a brief look at how these changes may affect simulation modelling at Ford.

## 6.2 Uptimes in Ford's model

To simulate a production line at Ford, the next time each machine will break down needs to be generated. Currently, Ford do this by picking one of the machine's observed uptimes at random.

As discussed in previous chapters, we have written code to simulate an ARMA-GARCH or MMPP model instead. Since each machine is modelled independently, this code can be applied to each machine individually.

In all, the changes Ford would need to make to implement our models are as follows:

1. First, Ford will need to fit models to their machines' uptimes, selecting a suitable ARMA-GARCH or MMPP model for each machine

2. Second, the parameters of these models will need to be included in the simulation

3. Third, in simulation, the state of the MMPP or the lagged values used in ARMA-GARCH need to be tracked for each machine

4. Fourth, when generating an uptime for a machine, Ford should sample from the MMPP or ARMA-GARCH, as discussed in past chapters.

## 6.3 The impact of our models on Ford's simulation

At the time of writing, Ford have not yet made the changes suggested in the previous section. As such, the impact of our models on Ford's simulations is as yet unknown. That said, we can briefly consider what this impact might be.

Both ARMA-GARCH and MMPP models allow us to model autocorrelated uptimes. In Ford's case, we have observed positive correlation between uptimes, indicating that short uptimes tend to be followed by further short uptimes. In other words, machines tend to break down repeatedly after short periods of work, effectively increasing the repair time needed before the machine can work again. In turn, this could cause machines that depend on the workstation's output to become starved more often, in turn starving other machines further down the line. Conversely, since long uptimes tend to be followed by long uptimes, Ford are likely to see long periods without breakdowns, where the production line is running efficiently.

## 6.4 Conclusion

In this chapter, we have briefly considered the changes Ford would need to make to incorporate our models into their simulations. We have indicated that the changes required are relatively modest and can be applied to machines individually, without affecting the rest of the simulation model. Though we expect our models to change the behaviour of the simulation, this cannot yet be confirmed as the required changes have not yet been made.

# Chapter 7

# Conclusion

To conclude, we will now briefly review the main problem of the thesis and the contributions of each chapter. In this thesis, we have discussed how Ford can model the times between breakdowns of machines in a production line, without assuming that the times are independent.

## 7.1 Obtaining uptimes from historical data

We began by considering the data Ford have available about their production line and how it was obtained. We noted that various issues may cause problems with the data collection process, leading to inaccurate data.

One of the main issues here was the presence of nonproductive periods, during which the production line was not operational. As we assumed that machines should not break down or be repaired during these times, the uptimes and downtimes need to be adjusted to take them into account.

After adjusting the uptimes, we noted that a large number of them are unrealistically small or large. This led us to exclude these outliers.

## 7.2 Detecting autocorrelation in uptimes

Second, we discussed how we may detect autocorrelation in the uptimes of Ford's machines. In the course of this, we needed to solve two problems:

1. How can we test for autocorrelation in the uptimes of a single machine?

2. Given the results of the tests for each machine, how can we decide which machines show autocorrelation, such that we do not make an overly large number of Type I errors?

To solve the first problem, we applied a Ljung-Box test [82]. This test allowed us to test whether the autocorrelation for a given time series was non-zero at a number of lags simultaneously.

Given the $p$-values of the Ljung-Box tests for all machines, we then needed to decide for which machines we should reject the null hypothesis. This *multiple comparison* problem can be solved in a number of ways, which we briefly reviewed. For the particular case of Ford's machines, we decided it may not be reasonable to assume the tests of different machines are independent. As such, we applied a Holm-Bonferroni [62] correction to compute adjusted $p$-values[132]. Comparing the adjusted $p$-values to the threshold of 0.05, we concluded that the uptimes are autocorrelated for most of Ford's machines.

## 7.3 ARMA and ARMA-GARCH models of uptimes

We then turned to applying time series analysis to the uptimes. First, we applied ARMA models to the uptimes. While, for many machines, these models provided an acceptable fit, the residuals of some machines appeared to vary over time.

To model uptimes with variance changing over time, we added GARCH terms to the ARMA models. The ARMA-GARCH models thus obtained provided a better fit for the uptimes of most machines. However, some of the machines still appeared to have residuals whose variance changed over time. For these machines, using higher-order GARCH terms might lead to an improved fit.

For both ARMA and ARMA-GARCH models, we also provided a procedure to generate uptimes using simulation.

## 7.4 Markov-Modulated Poisson Processes for uptimes

As an alternative to using ARMA and ARMA-GARCH models, we treated a sequence of times between breakdowns as arrival times in an arrival process. This allowed us to model the uptimes using a Markov-Modulated Poisson Process.

Using the EM algorithm of Rydén[114], we obtained fits of the MMPP to the uptimes of each of Ford's machines. Our fitting procedure chose MMPPs of order 2 and 3, suggesting that higher-order MMPPs do not provide a sufficiently large improvement in fit.

Our diagnostic plots for MMPPs showed that they were generally better than ARMA models at modelling the distribution of the uptimes. Despite this heuristic result, an objective comparison using AIC showed that the ARMA and ARMA-GARCH models are preferred in most cases.

As well as fitting MMPPs and providing diagnostics, we also provided a procedure to simulate uptimes from the MMPP.

## 7.5 Conclusion and future work

In this thesis, we have considered a number of models Ford can use to model the correlated times between breakdowns of their machines. These models enable Ford to improve their simulations.

As discussed in a previous section, most manufacturing models to date have assumed independence of all random variables in the system. In this thesis, we extend this literature by studying autocorrelation in machine uptimes in detail. Our work focuses on the practical aspects of detecting and modelling autocorrelated uptimes, as well as including them in simulations.

Our first contribution is a practical procedure to detect autocorrelation in uptimes. The procedure has very mild assumptions and compensates for the number of machines it is applied to, ensuring that the probability of a Type I error is kept low.

Second, we provide two ways to model autocorrelated uptimes. The first, an extension of the autoregressive and ARTA methods discussed previously, is to use ARMA models including GARCH terms. We also provide a method based on the Markov-Modulated Poisson Process, a special case of the Markov Arrival Process.

For both methods, we provide diagnostic plots and a quantitative way to select the most appropriate model for a given series of uptimes. This allows us to automatically select an appropriate model.

Finally, to enable Ford to use our methods in simulation, we provide a way to generate simulated uptimes from each of our models.

# Appendix A

# Diagnostic plots for ARMA fits

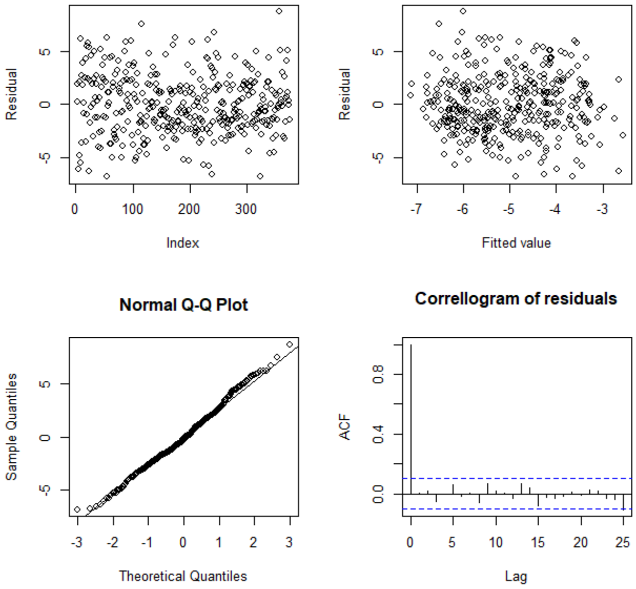


Figure A.1: ARMA diagnostic plot for machine DAG0160-0

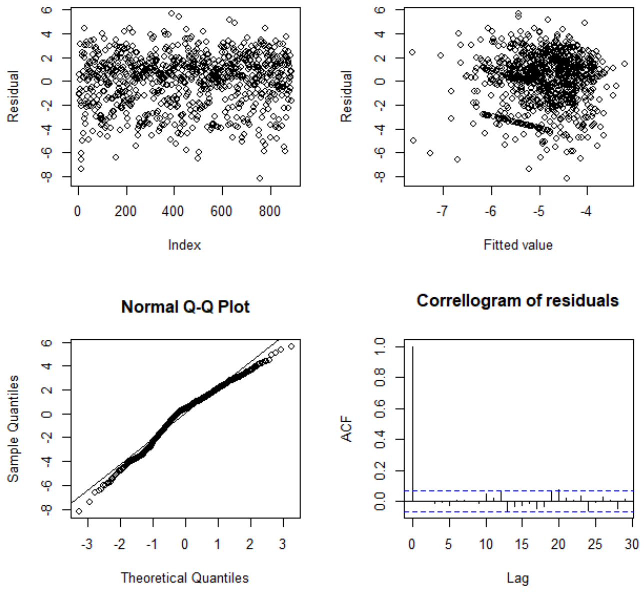


FIGURE A.2: ARMA diagnostic plot for machine OP 100-2

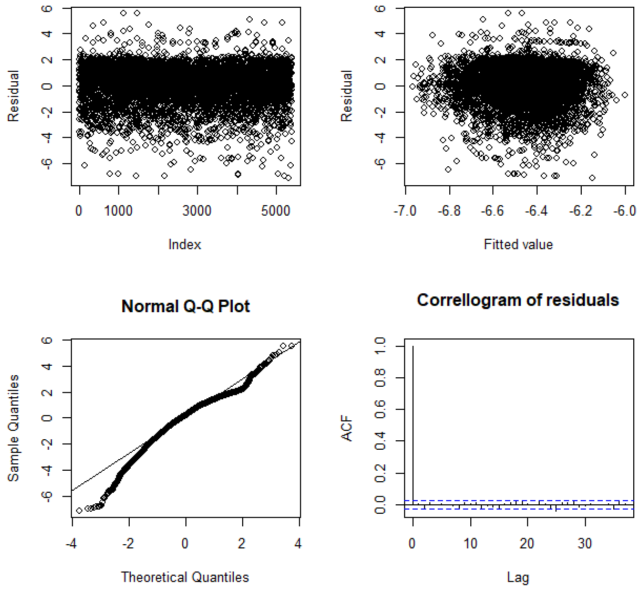


FIGURE A.3: ARMA diagnostic plot for machine OP 10A-0

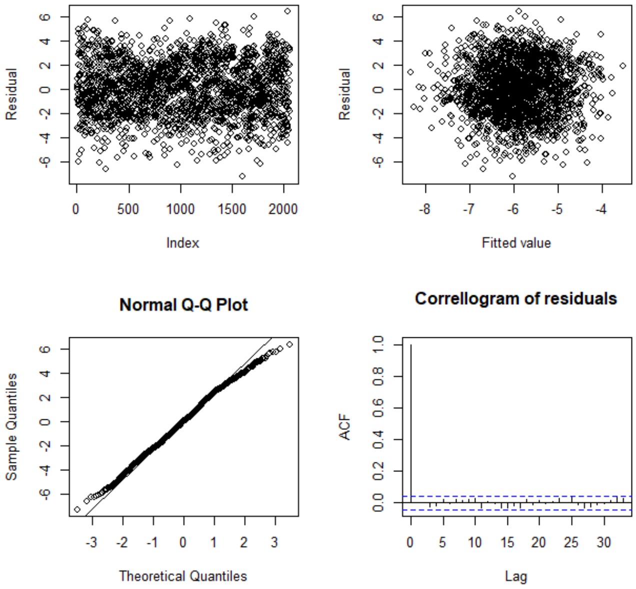


Figure A.4: ARMA diagnostic plot for machine OP 110-0

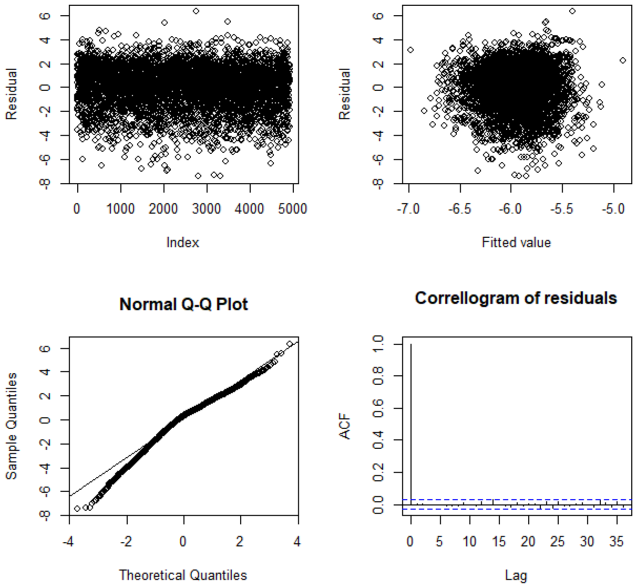


Figure A.5: ARMA diagnostic plot for machine OP 125-0

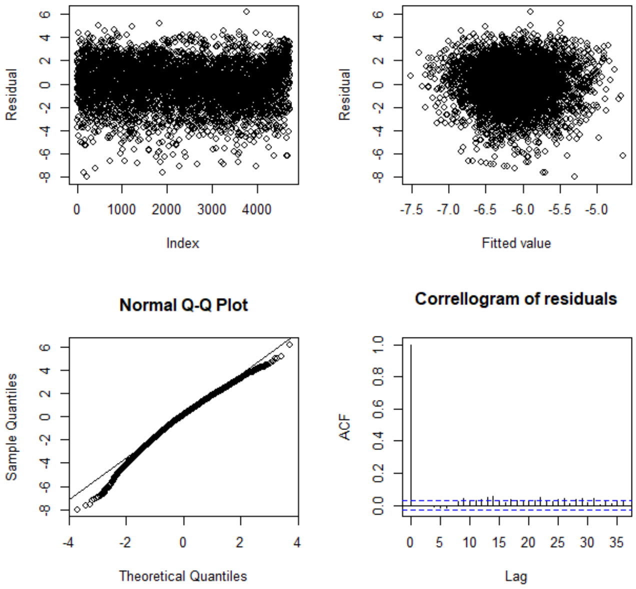


Figure A.6: ARMA diagnostic plot for machine OP 135-0

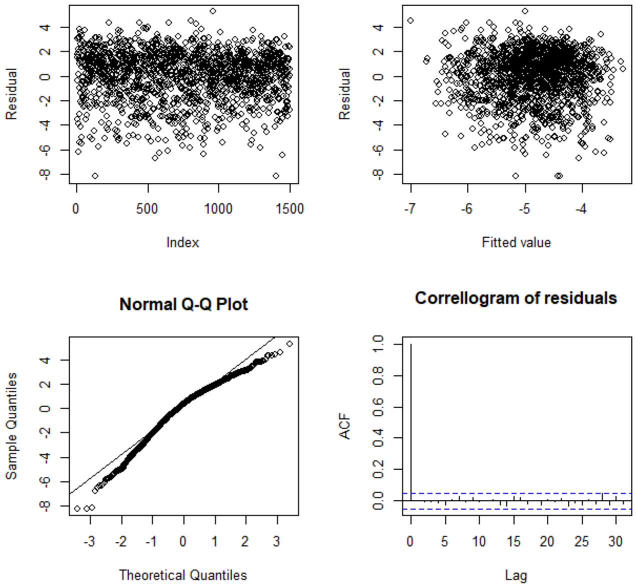


Figure A.7: ARMA diagnostic plot for machine OP 140-0

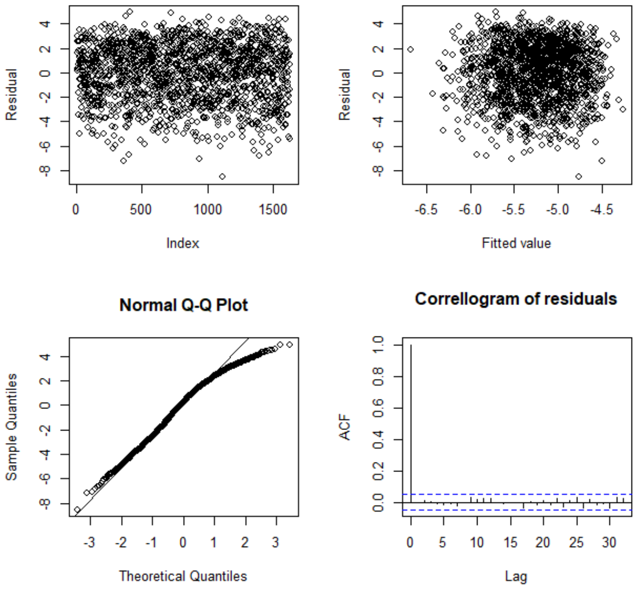


Figure A.8: ARMA diagnostic plot for machine OP 150-0

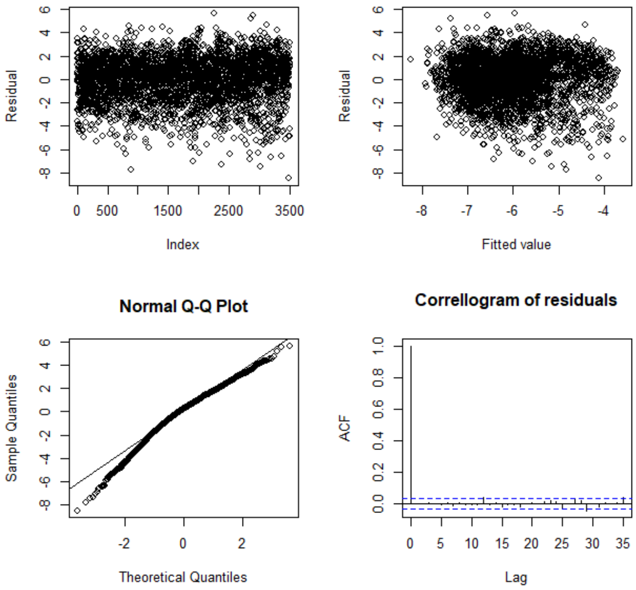


Figure A.9: ARMA diagnostic plot for machine OP 160-0

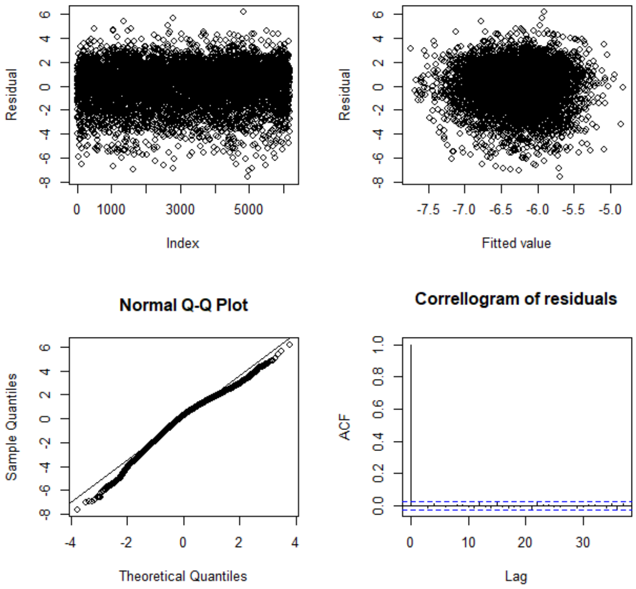


FIGURE A.10: ARMA diagnostic plot for machine OP 170_SCR-0

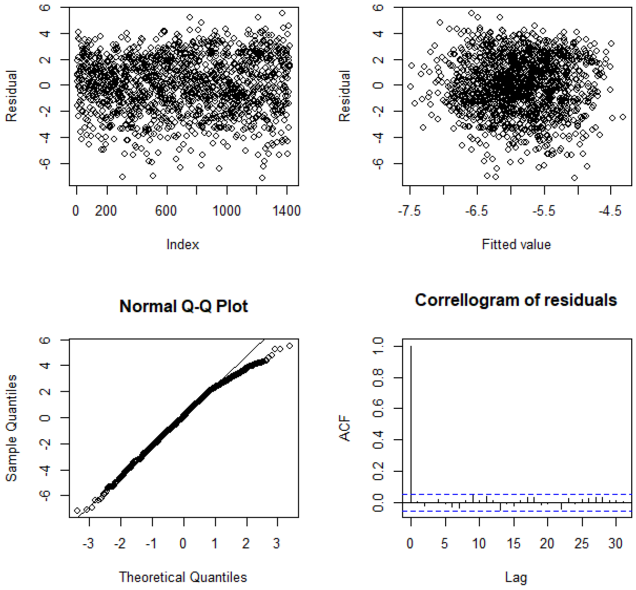


FIGURE A.11: ARMA diagnostic plot for machine OP 30A-0

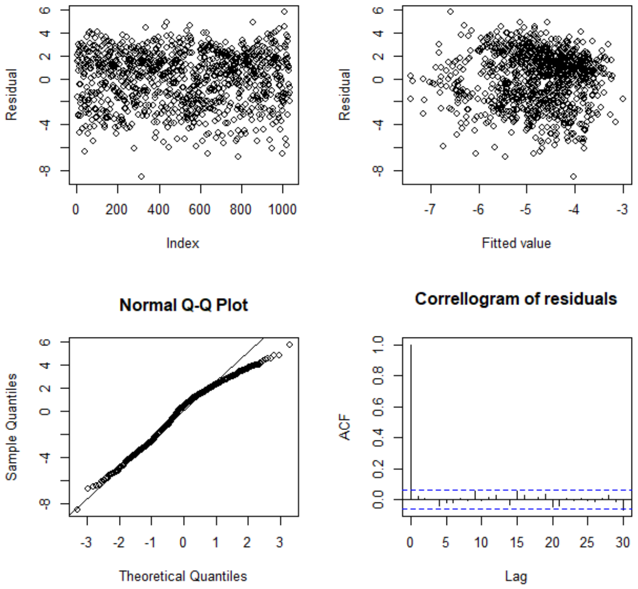


Figure A.12: ARMA diagnostic plot for machine OP 40-0

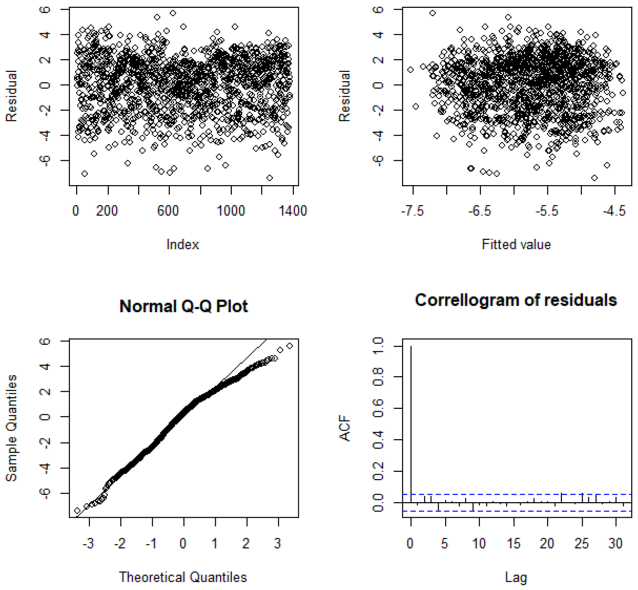


Figure A.13: ARMA diagnostic plot for machine OP 45A-0

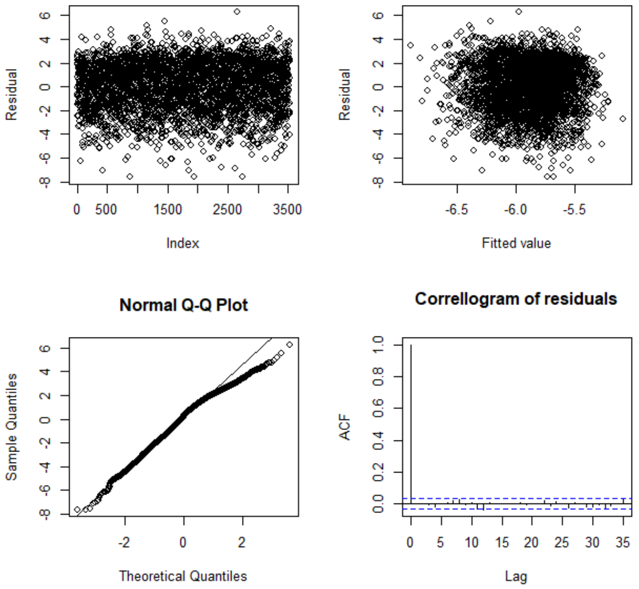


FIGURE A.14: ARMA diagnostic plot for machine OP 70-6

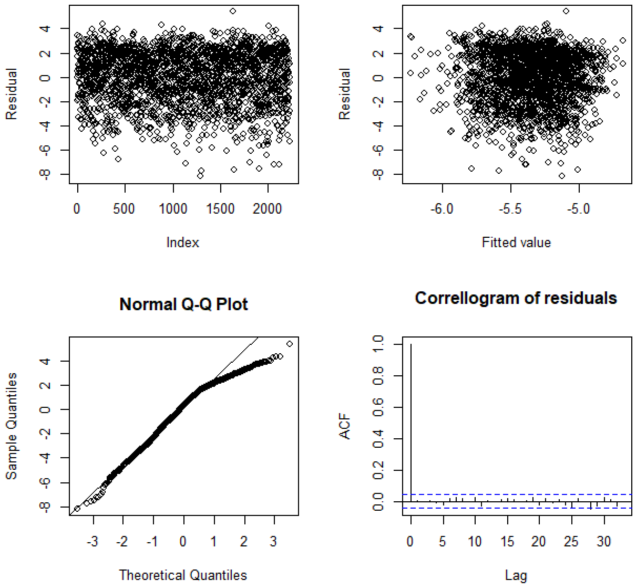


FIGURE A.15: ARMA diagnostic plot for machine OP10-1

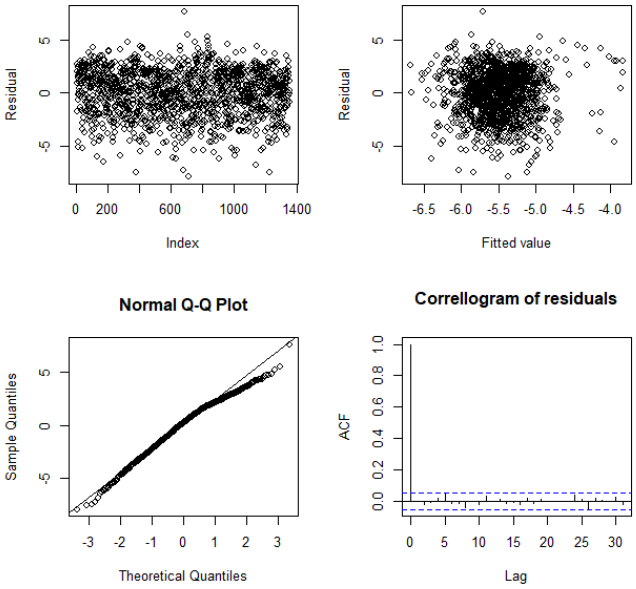


FIGURE A.16: ARMA diagnostic plot for machine OP10-2

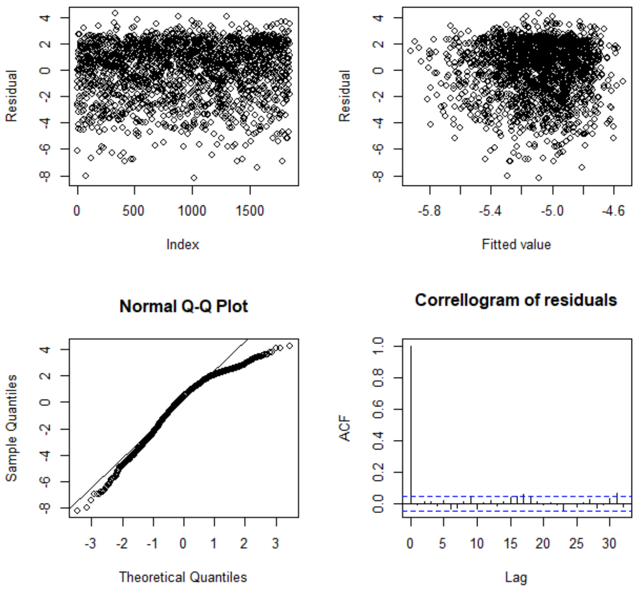


FIGURE A.17: ARMA diagnostic plot for machine OP10-3

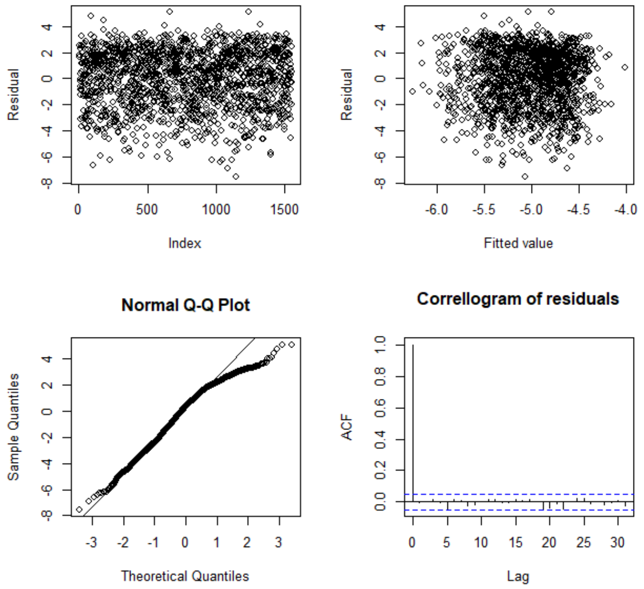


FIGURE A.18: ARMA diagnostic plot for machine OP10-4

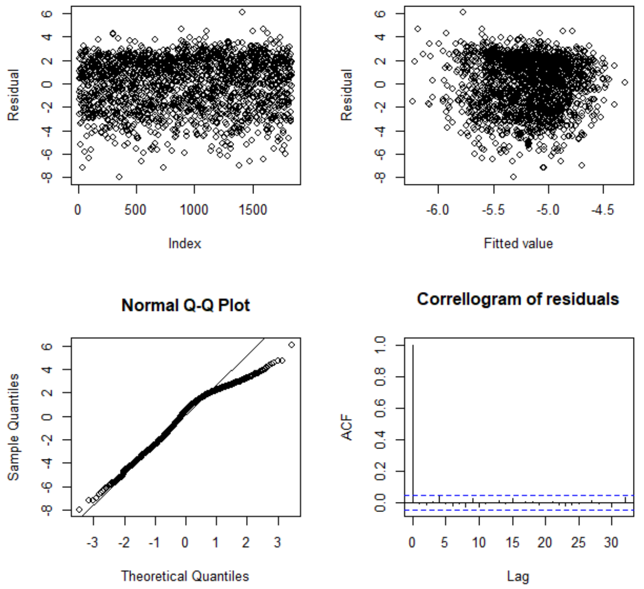


FIGURE A.19: ARMA diagnostic plot for machine OP10-5

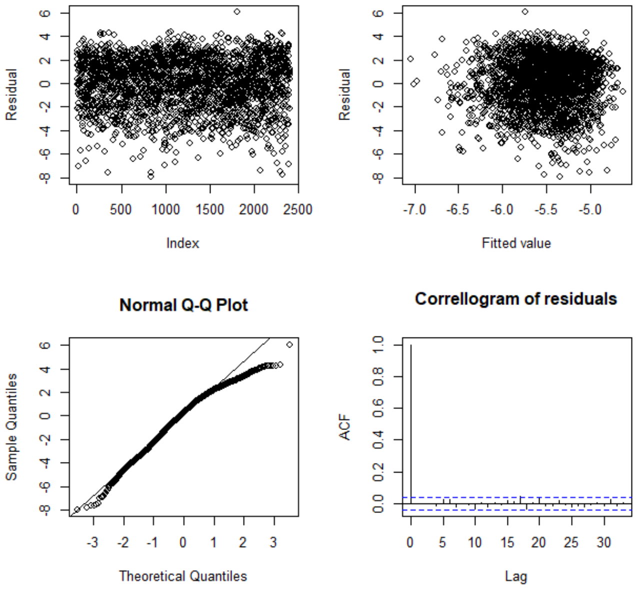


Figure A.20: ARMA diagnostic plot for machine OP10-6

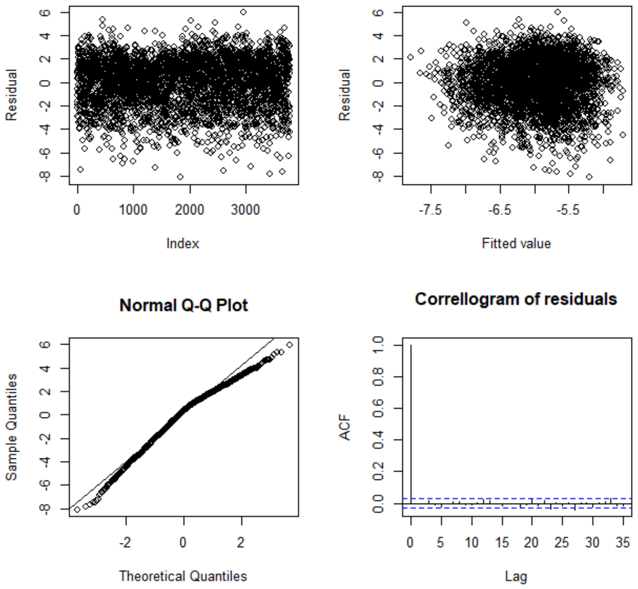


Figure A.21: ARMA diagnostic plot for machine OP100G-0

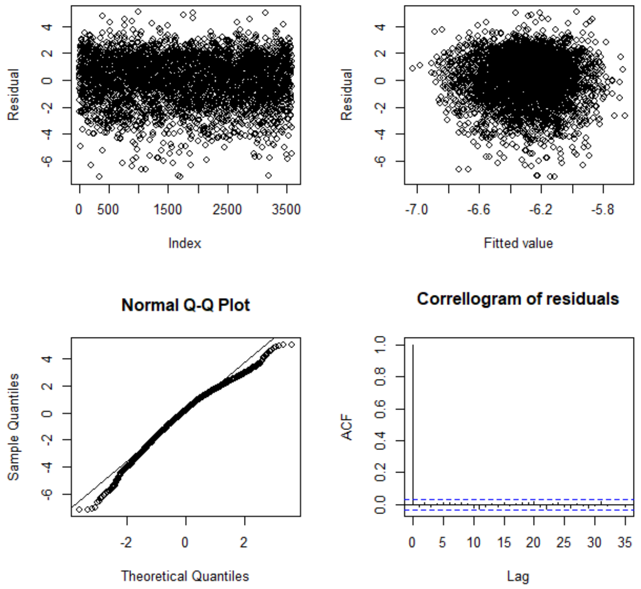


Figure A.22: ARMA diagnostic plot for machine OP10A-30AG-0

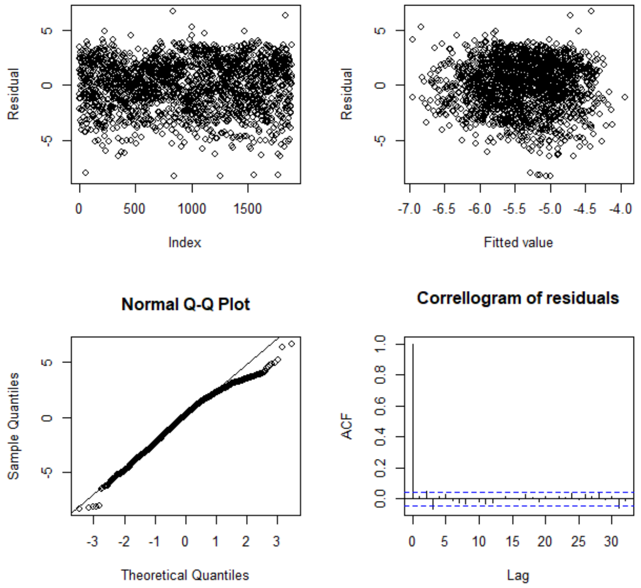


Figure A.23: ARMA diagnostic plot for machine OP10G-0

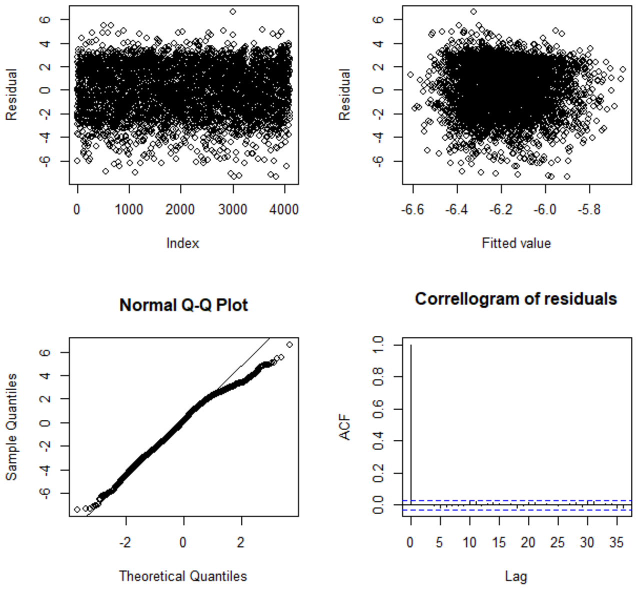


FIGURE A.24: ARMA diagnostic plot for machine OP120-1

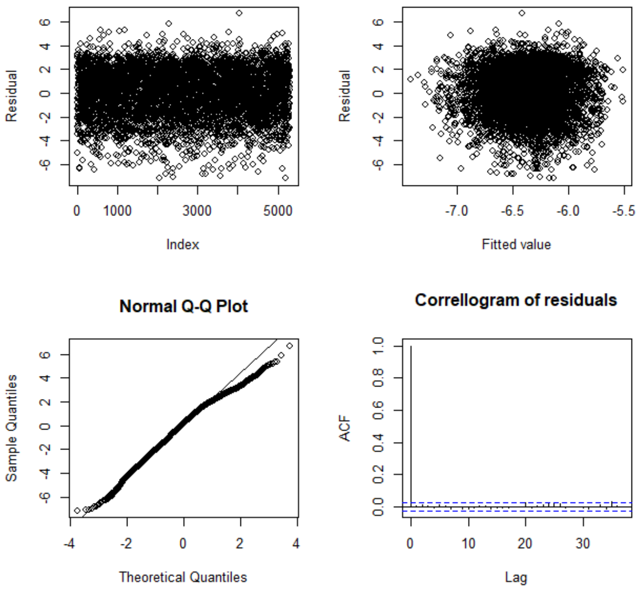


FIGURE A.25: ARMA diagnostic plot for machine OP120-2

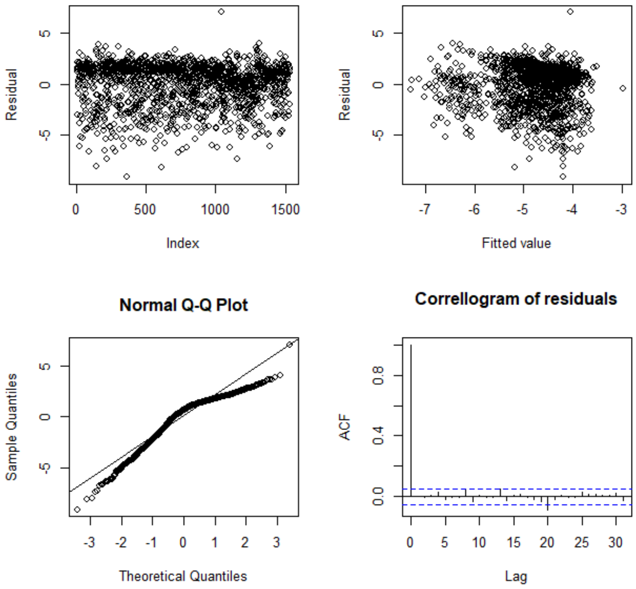


FIGURE A.26: ARMA diagnostic plot for machine OP130-1

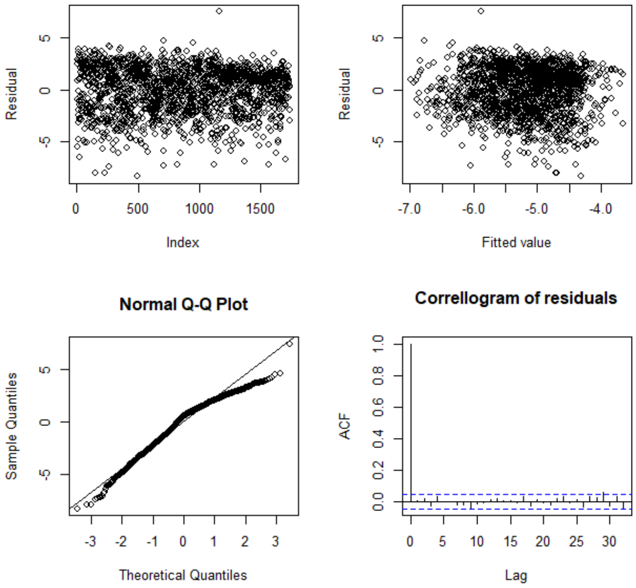


FIGURE A.27: ARMA diagnostic plot for machine OP130-2

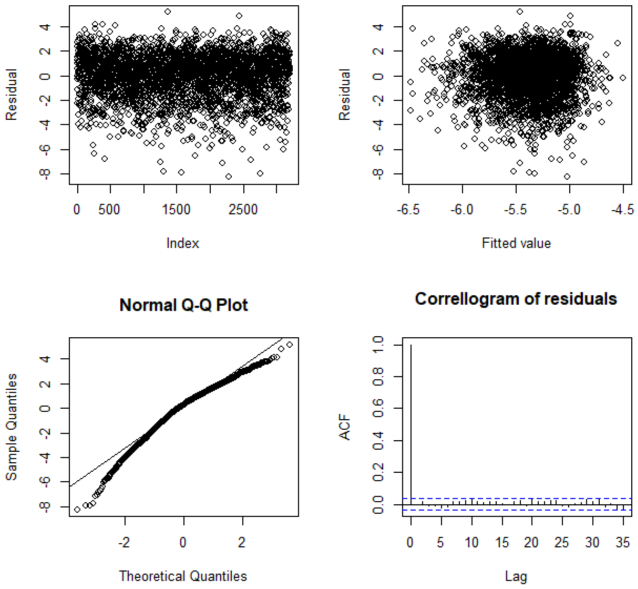


Figure A.28: ARMA diagnostic plot for machine OP130G-0

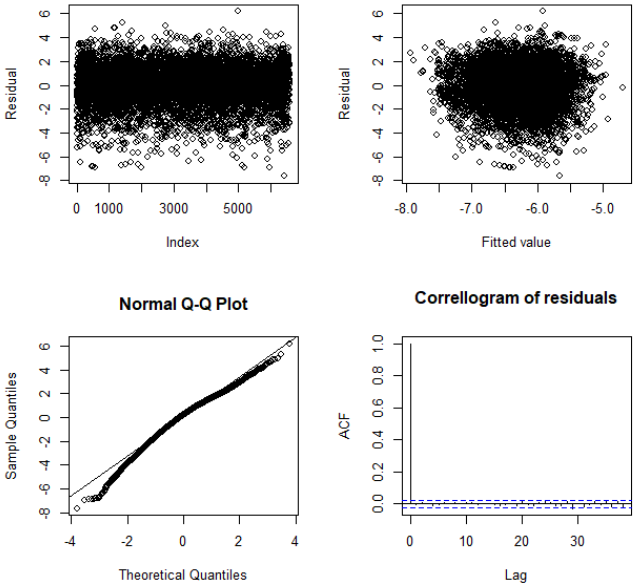


Figure A.29: ARMA diagnostic plot for machine OP170_LEAK-0

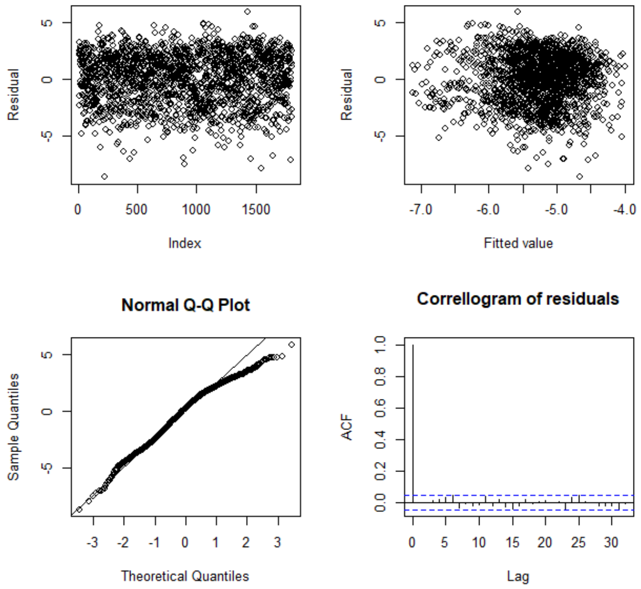


Figure A.30: ARMA diagnostic plot for machine OP20-1

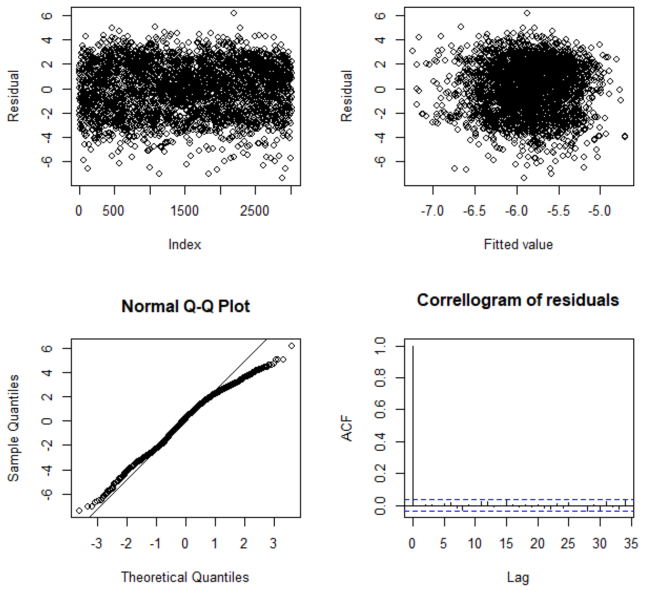


Figure A.31: ARMA diagnostic plot for machine OP20-10

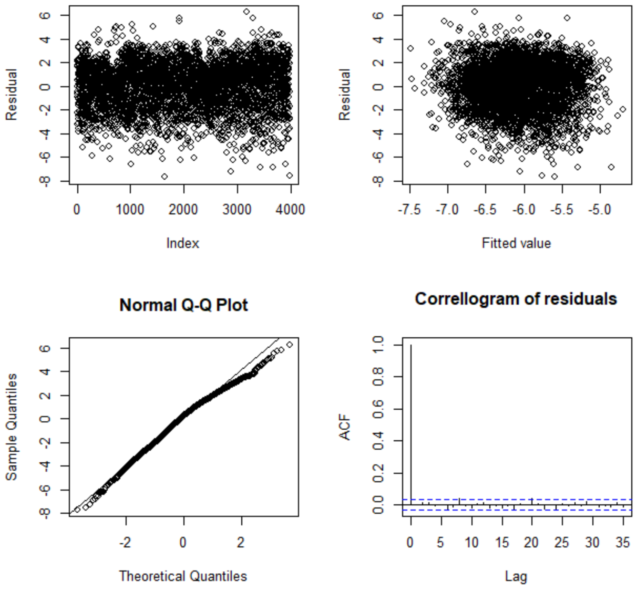


FIGURE A.32: ARMA diagnostic plot for machine OP20-11

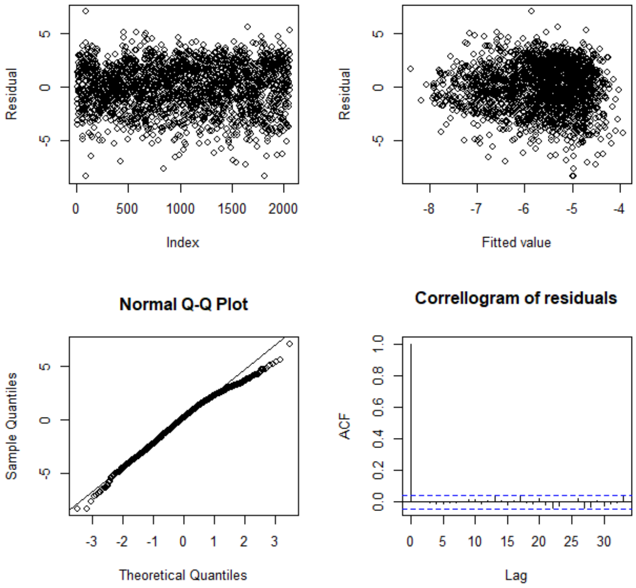


FIGURE A.33: ARMA diagnostic plot for machine OP20-12

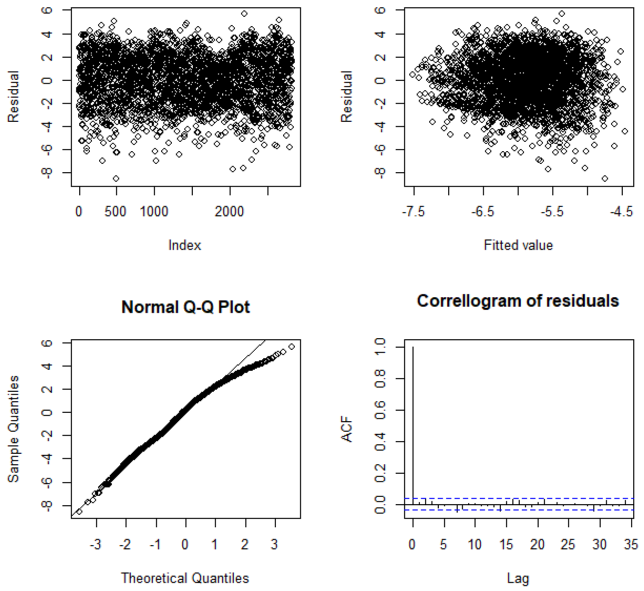


FIGURE A.34: ARMA diagnostic plot for machine OP20-2

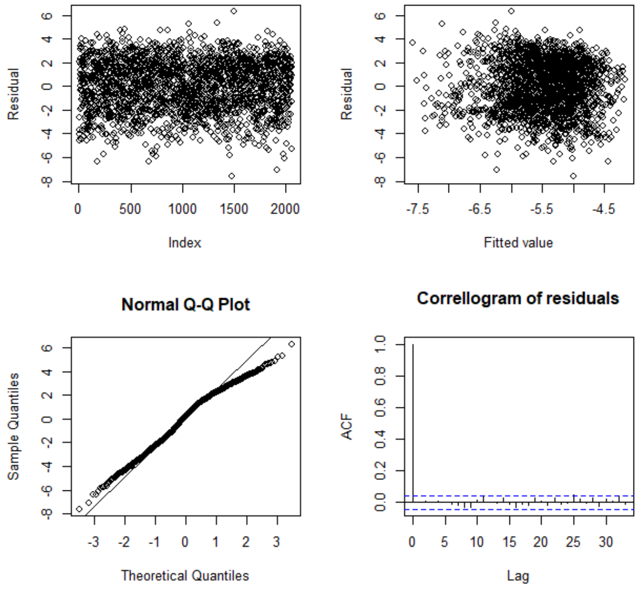


FIGURE A.35: ARMA diagnostic plot for machine OP20-3

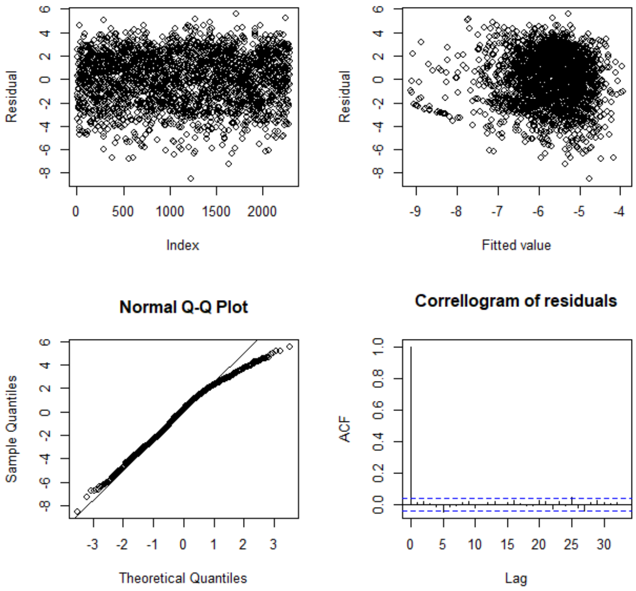


Figure A.36: ARMA diagnostic plot for machine OP20-4

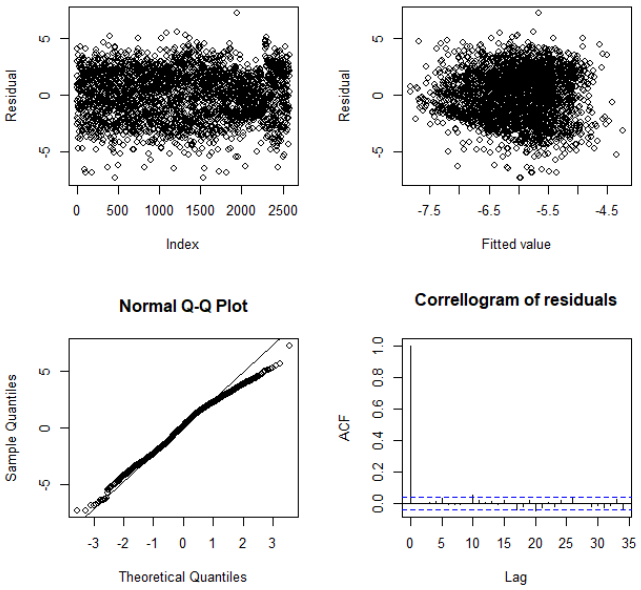


Figure A.37: ARMA diagnostic plot for machine OP20-5

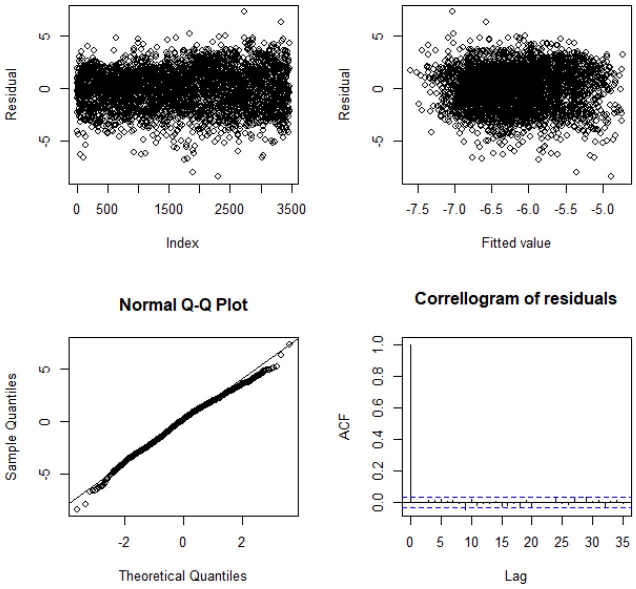


FIGURE A.38: ARMA diagnostic plot for machine OP20-6

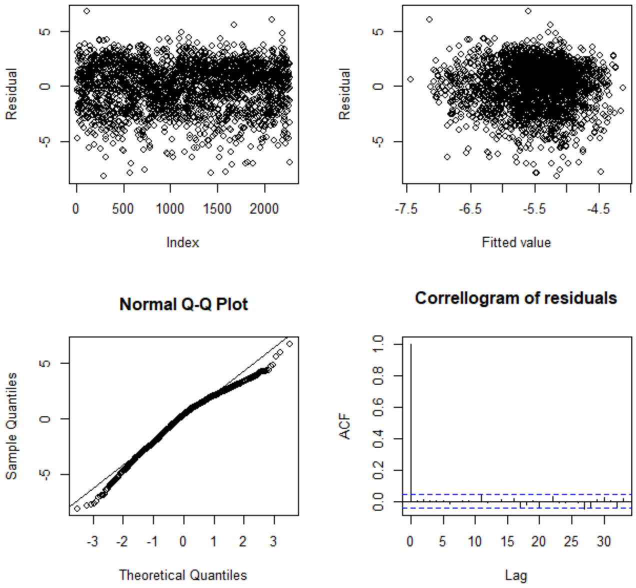


FIGURE A.39: ARMA diagnostic plot for machine OP20-7

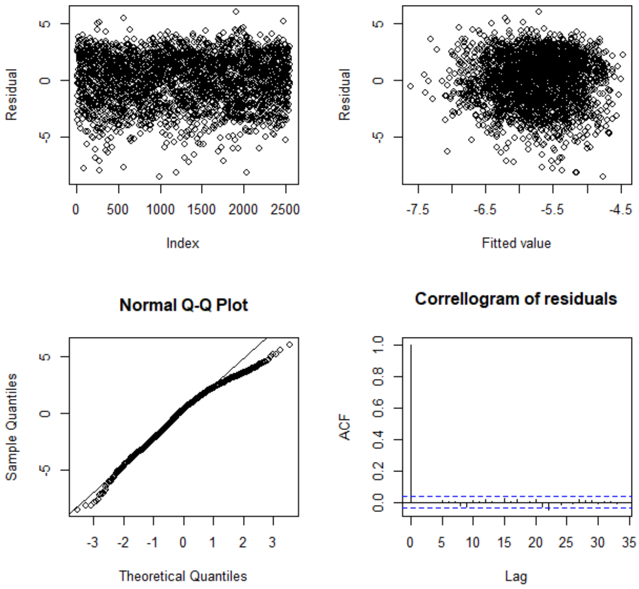


Figure A.40: ARMA diagnostic plot for machine OP20-8

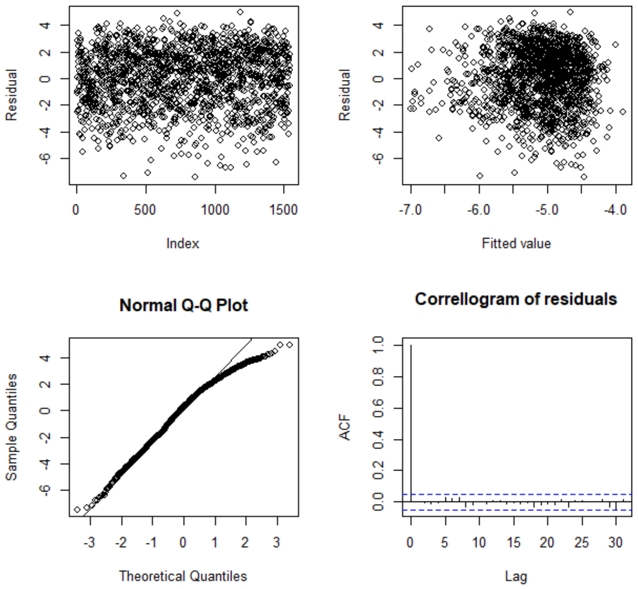


Figure A.41: ARMA diagnostic plot for machine OP20-9

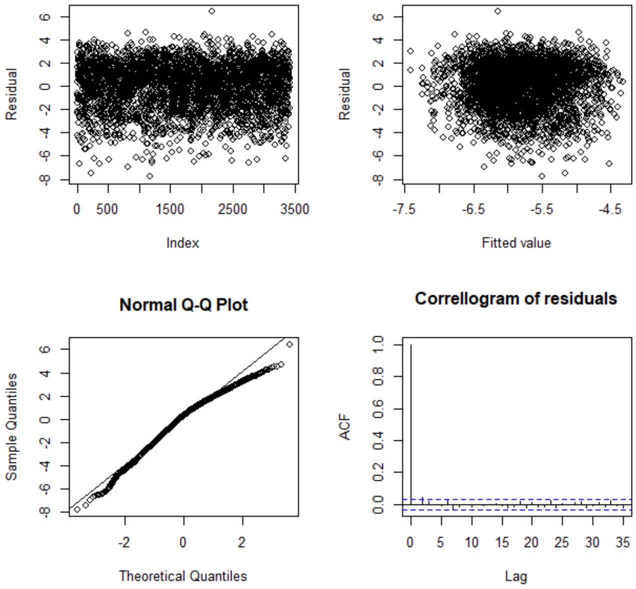


Figure A.42: ARMA diagnostic plot for machine OP20G-0

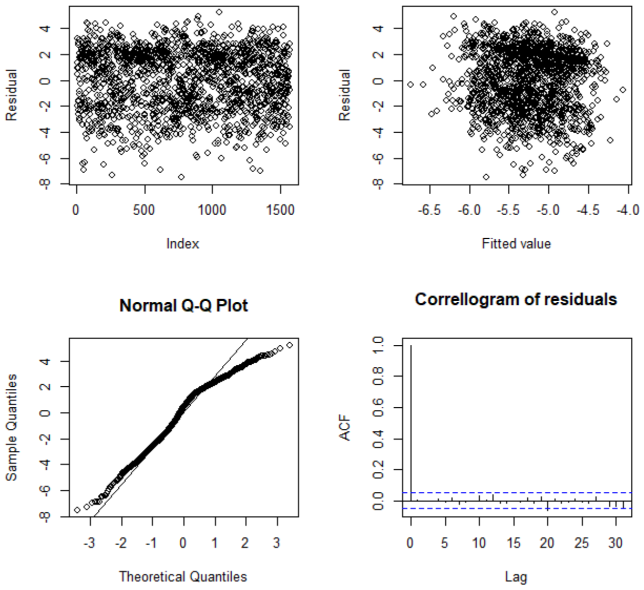


Figure A.43: ARMA diagnostic plot for machine OP30-40G-0

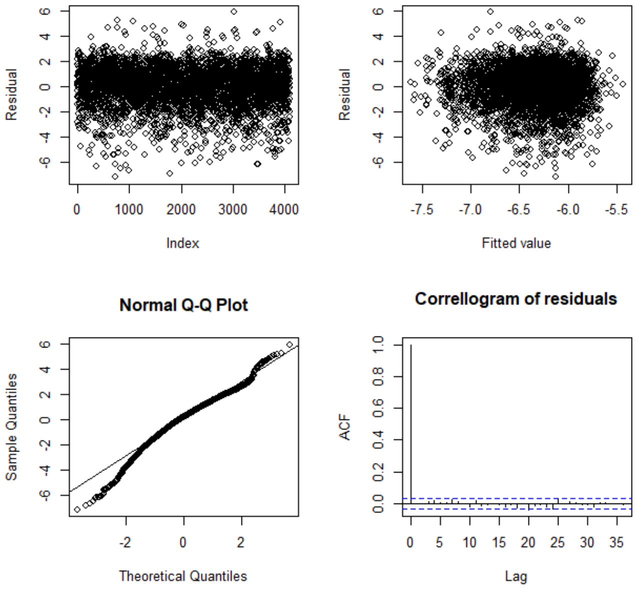


Figure A.44: ARMA diagnostic plot for machine OP45-50C-0

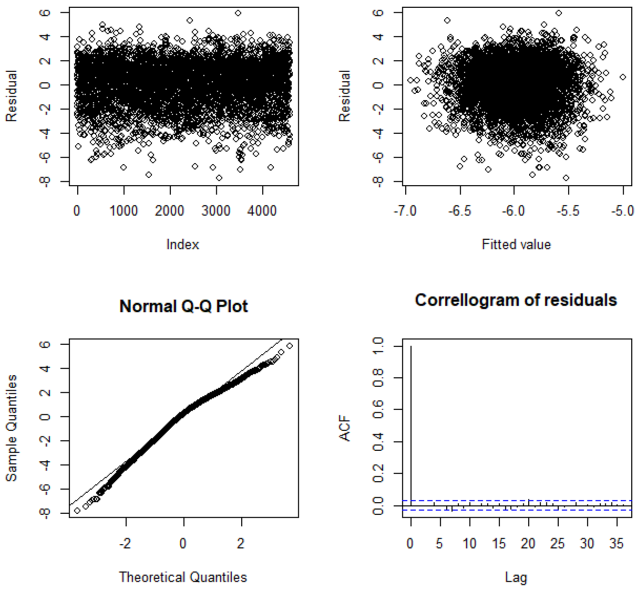


Figure A.45: ARMA diagnostic plot for machine OP50_BLTRD-0

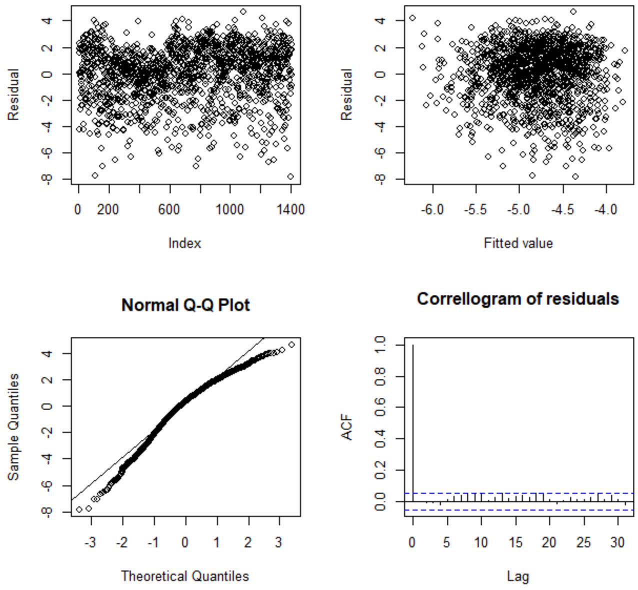


FIGURE A.46: ARMA diagnostic plot for machine OP55-0

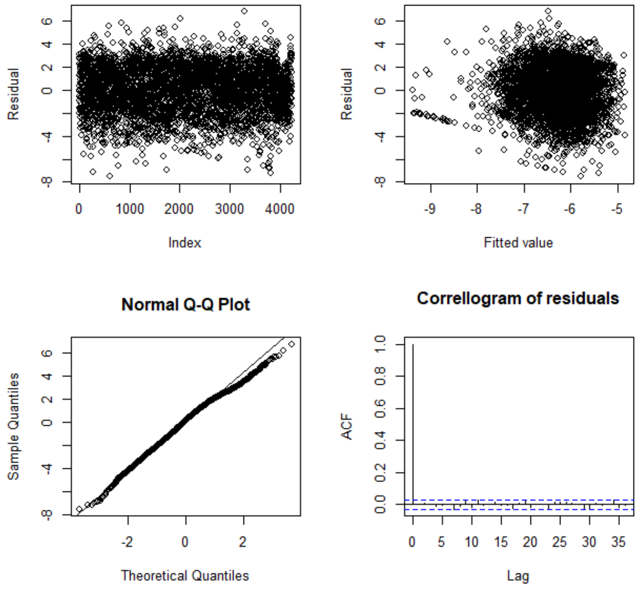


FIGURE A.47: ARMA diagnostic plot for machine OP70-1

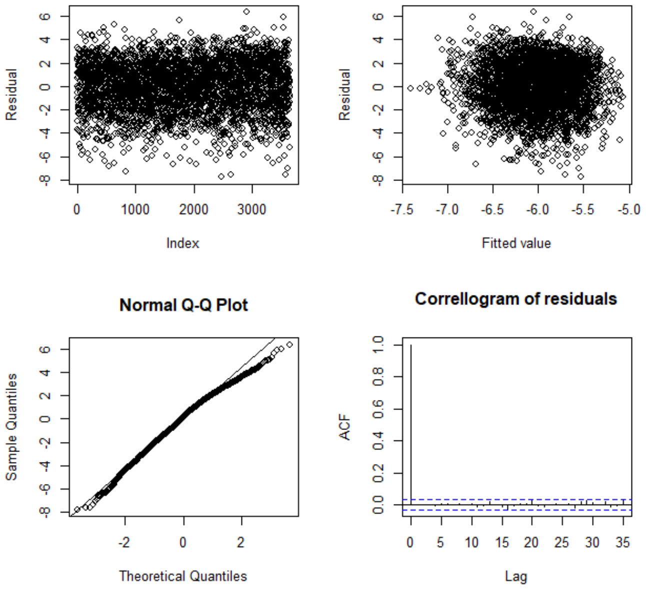


FIGURE A.48: ARMA diagnostic plot for machine OP70-10

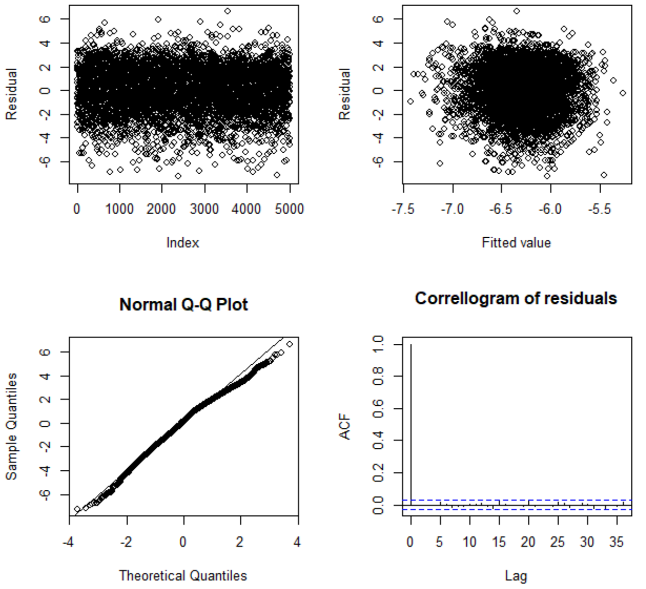


FIGURE A.49: ARMA diagnostic plot for machine OP70-12

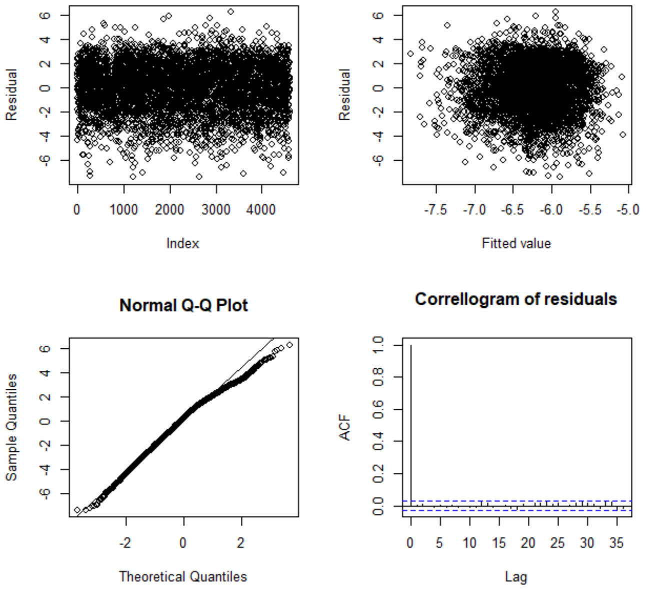


FIGURE A.50: ARMA diagnostic plot for machine OP70-2

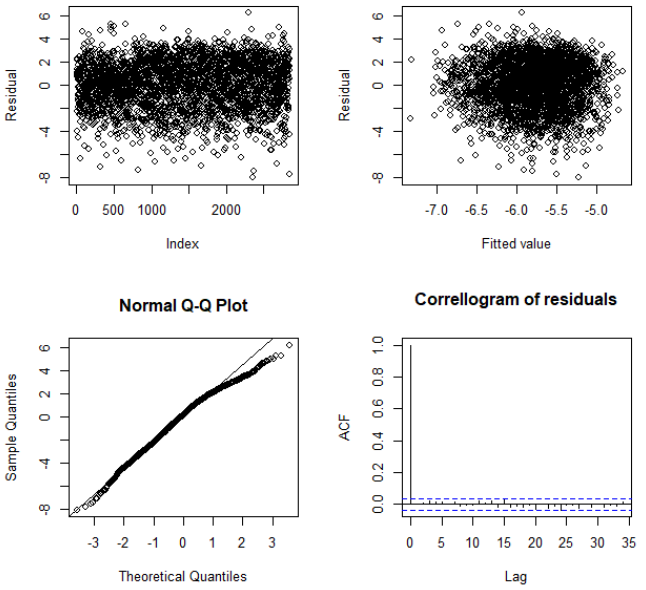


FIGURE A.51: ARMA diagnostic plot for machine OP70-3

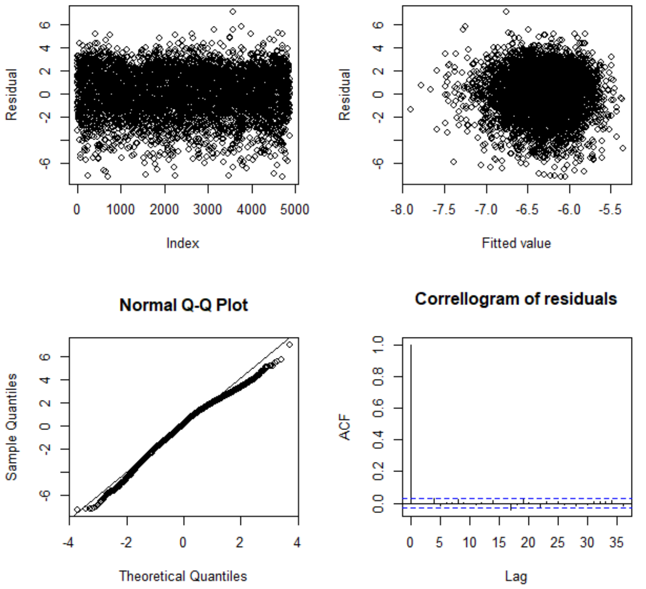


Figure A.52: ARMA diagnostic plot for machine OP70-4

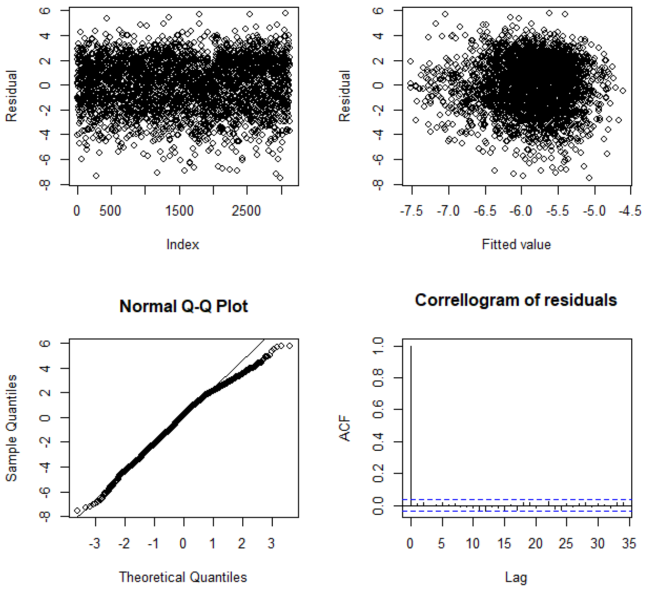


Figure A.53: ARMA diagnostic plot for machine OP70-5

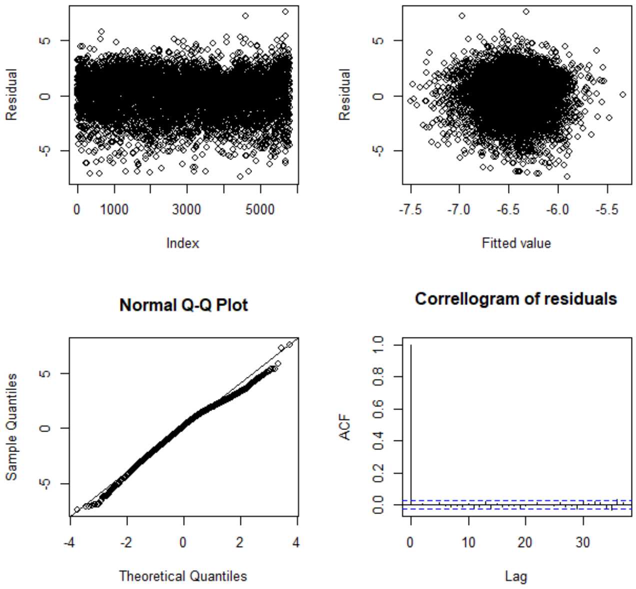


Figure A.54: ARMA diagnostic plot for machine OP70-7

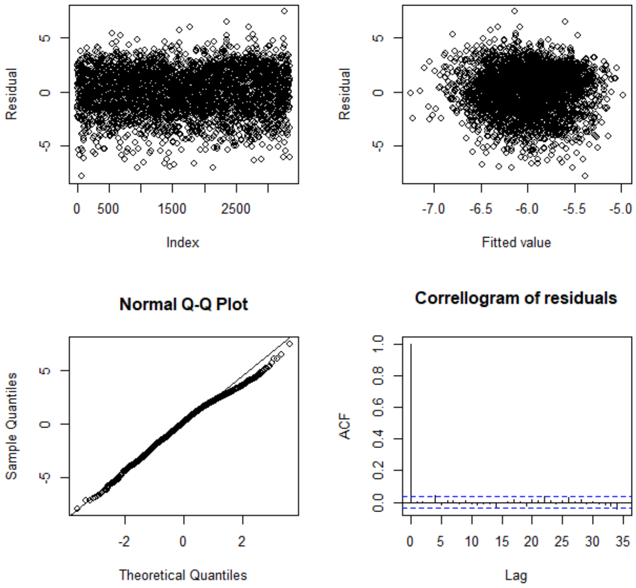


Figure A.55: ARMA diagnostic plot for machine OP70-8

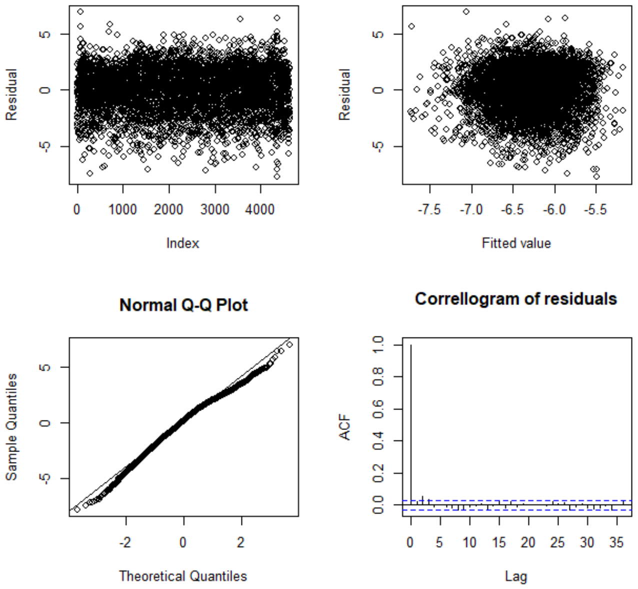


Figure A.56: ARMA diagnostic plot for machine OP70-9

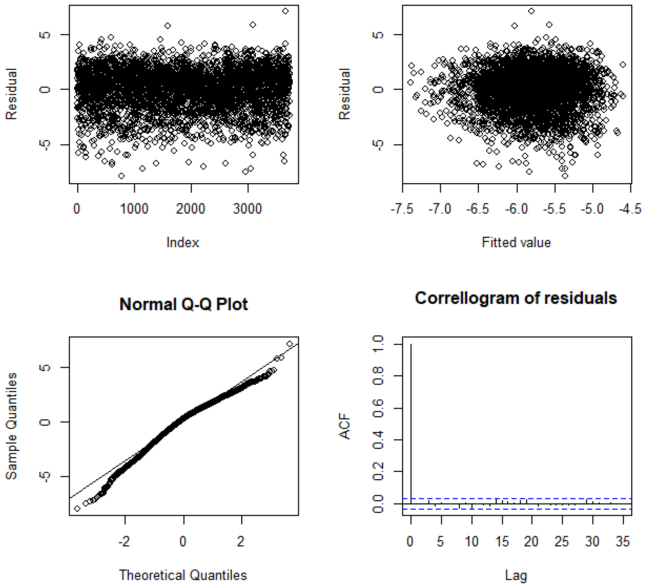


Figure A.57: ARMA diagnostic plot for machine OP70G-0

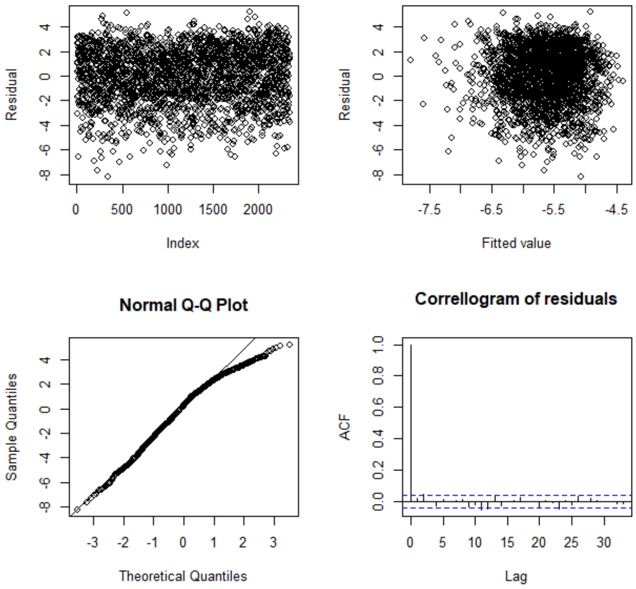


Figure A.58: ARMA diagnostic plot for machine OP80-1

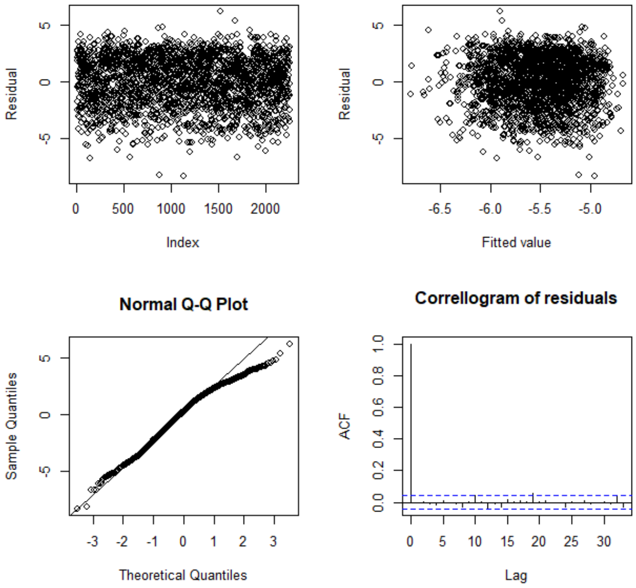


Figure A.59: ARMA diagnostic plot for machine OP80-10

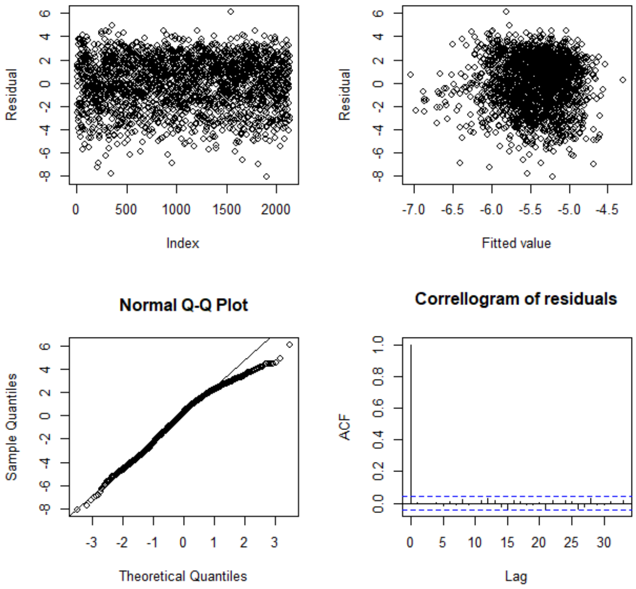


FIGURE A.60: ARMA diagnostic plot for machine OP80-2

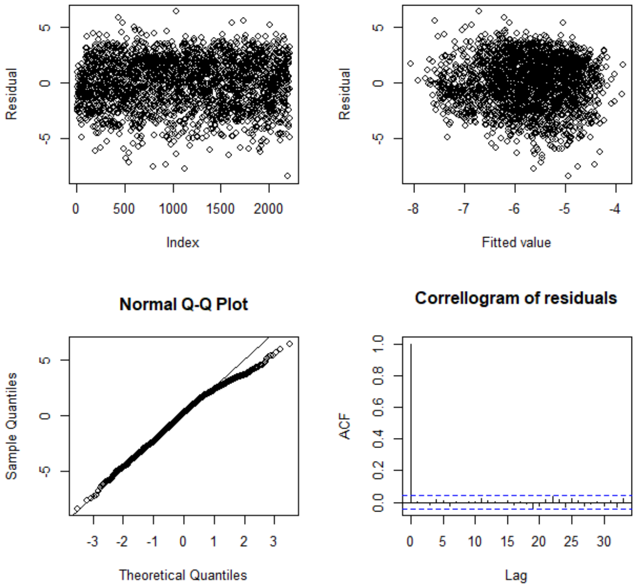


FIGURE A.61: ARMA diagnostic plot for machine OP80-3

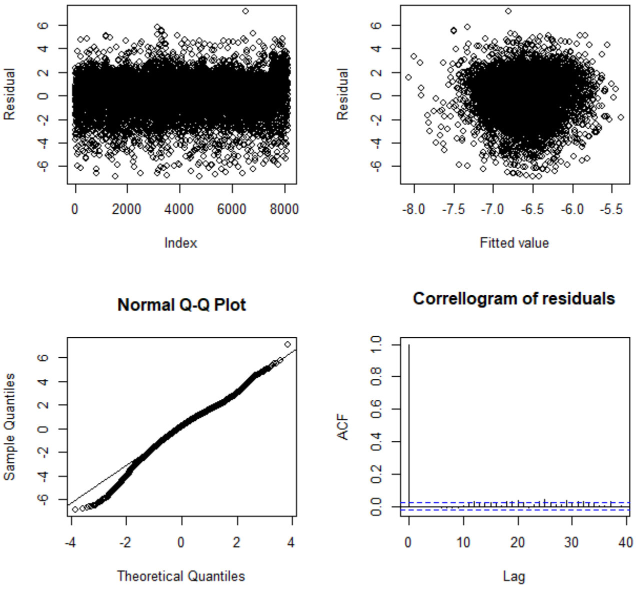


FIGURE A.62: ARMA diagnostic plot for machine OP80-4

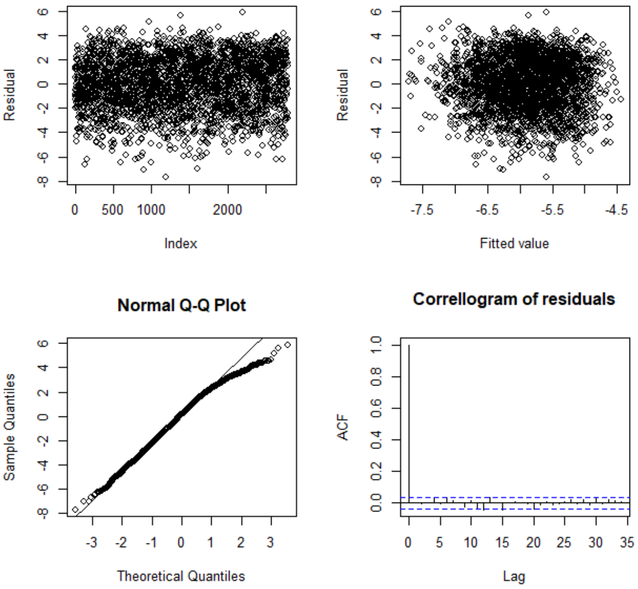


FIGURE A.63: ARMA diagnostic plot for machine OP80-5

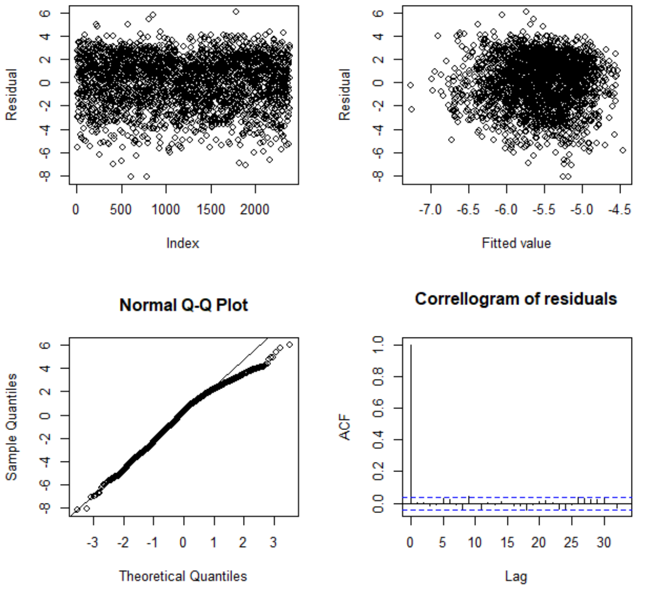


FIGURE A.64: ARMA diagnostic plot for machine OP80-6

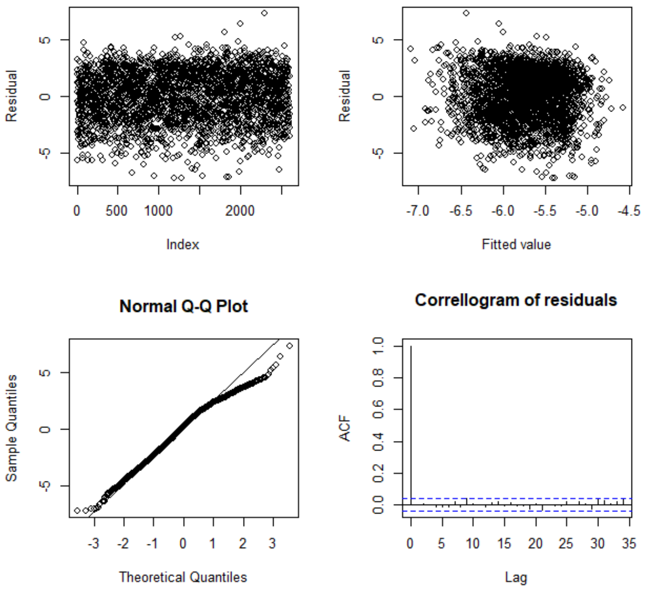


FIGURE A.65: ARMA diagnostic plot for machine OP80-7

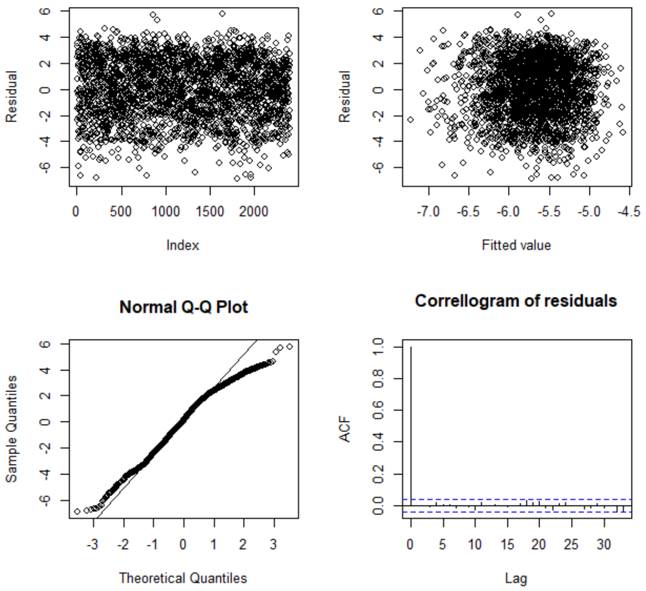


Figure A.66: ARMA diagnostic plot for machine OP80-8

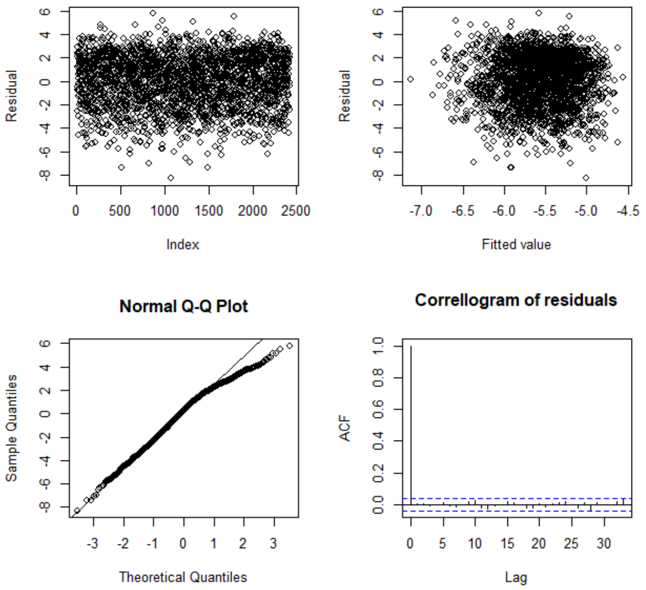


Figure A.67: ARMA diagnostic plot for machine OP80-9

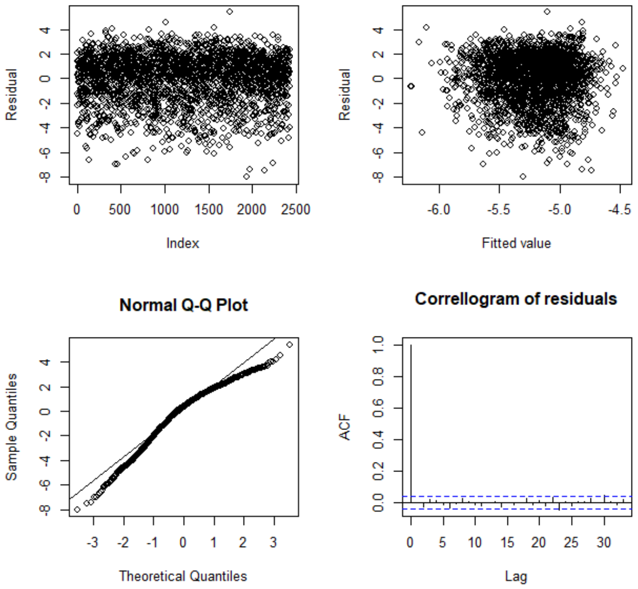


Figure A.68: ARMA diagnostic plot for machine OP80G-0

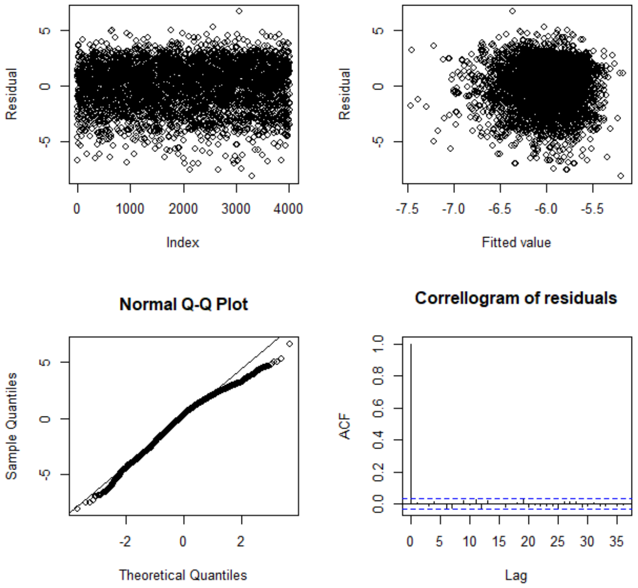


Figure A.69: ARMA diagnostic plot for machine OP90-1

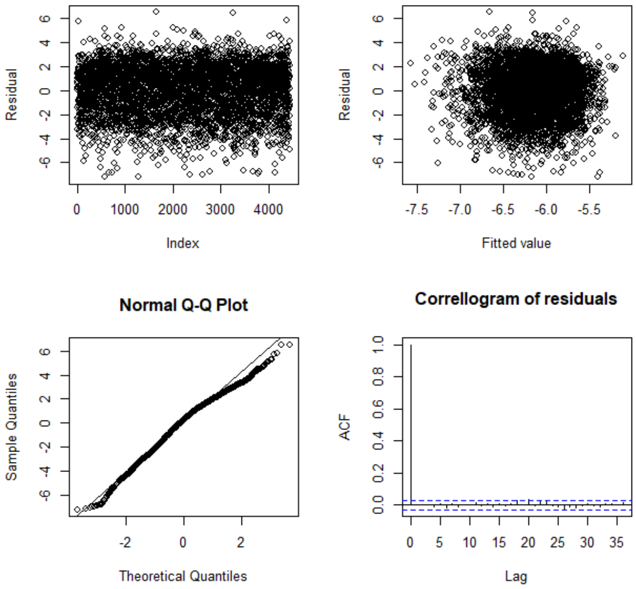


FIGURE A.70: ARMA diagnostic plot for machine OP90-10

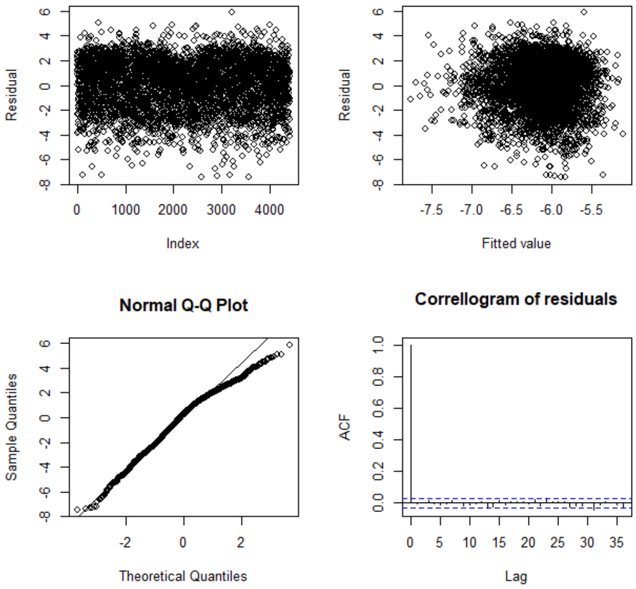


FIGURE A.71: ARMA diagnostic plot for machine OP90-12

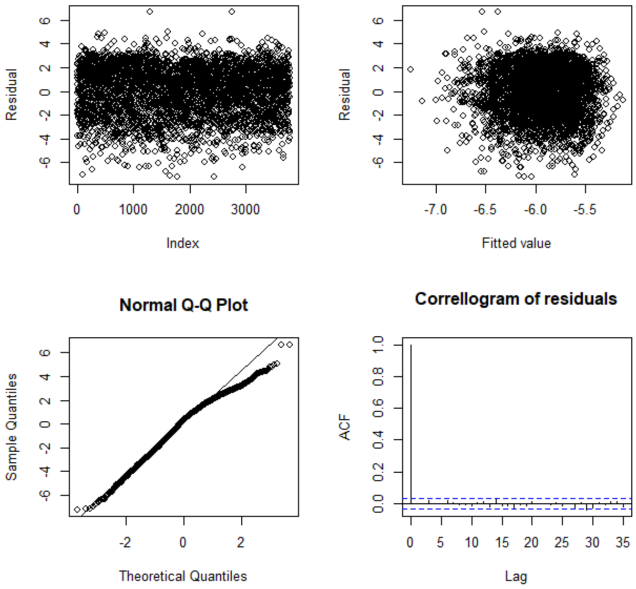


Figure A.72: ARMA diagnostic plot for machine OP90-2

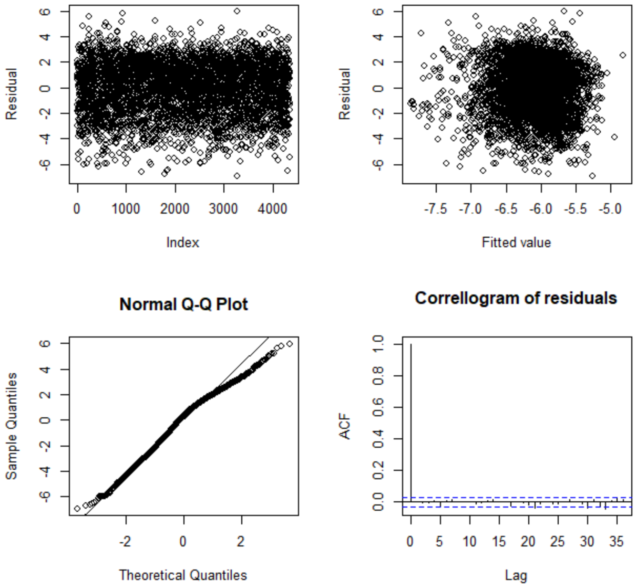


Figure A.73: ARMA diagnostic plot for machine OP90-3

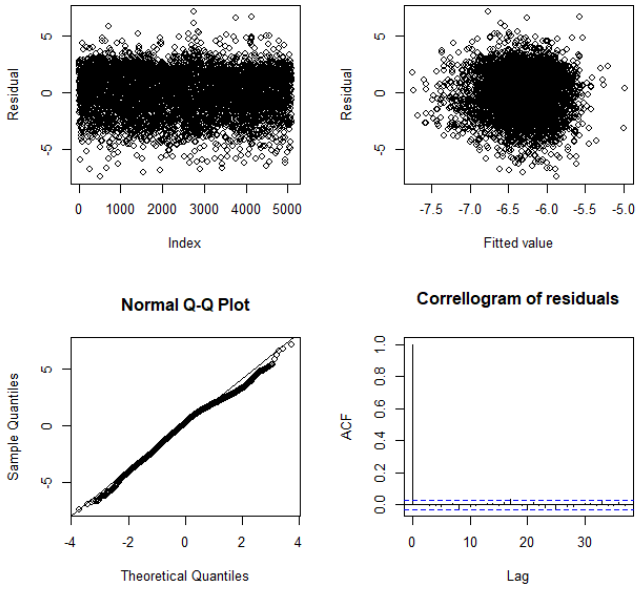


Figure A.74: ARMA diagnostic plot for machine OP90-4

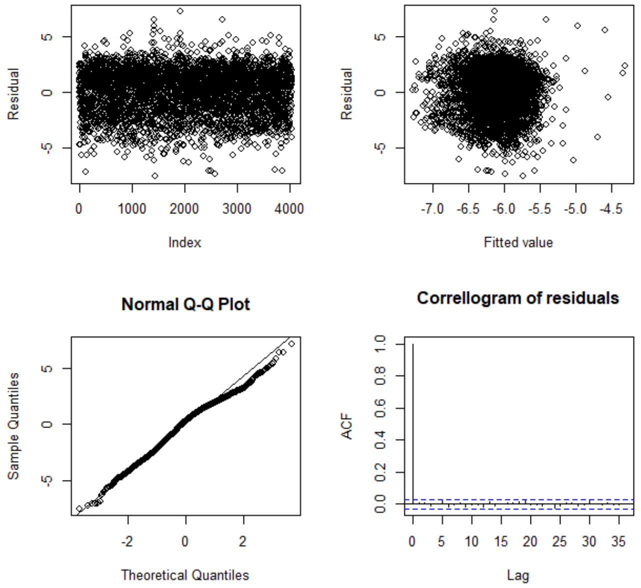


Figure A.75: ARMA diagnostic plot for machine OP90-5

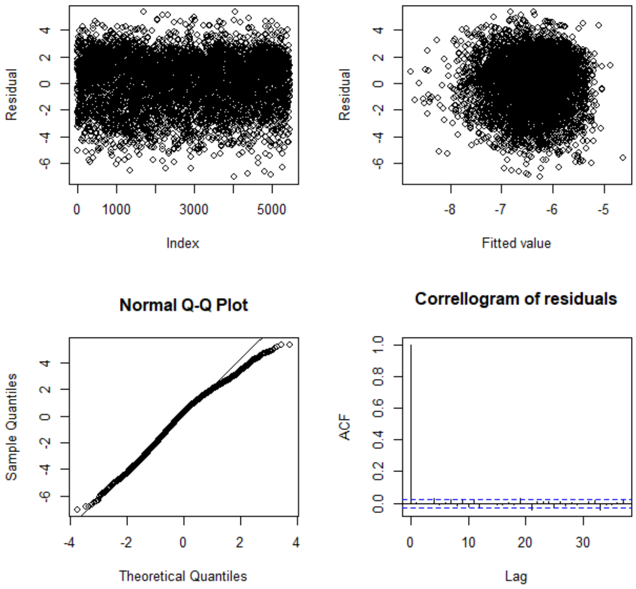


FIGURE A.76: ARMA diagnostic plot for machine OP90-6

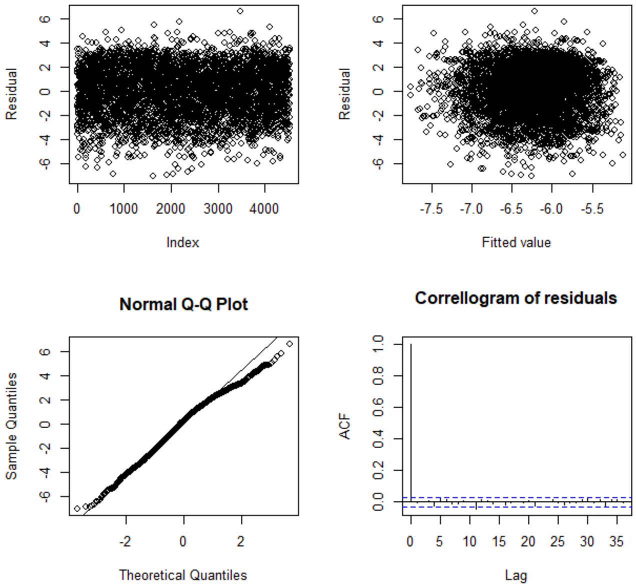


FIGURE A.77: ARMA diagnostic plot for machine OP90-7

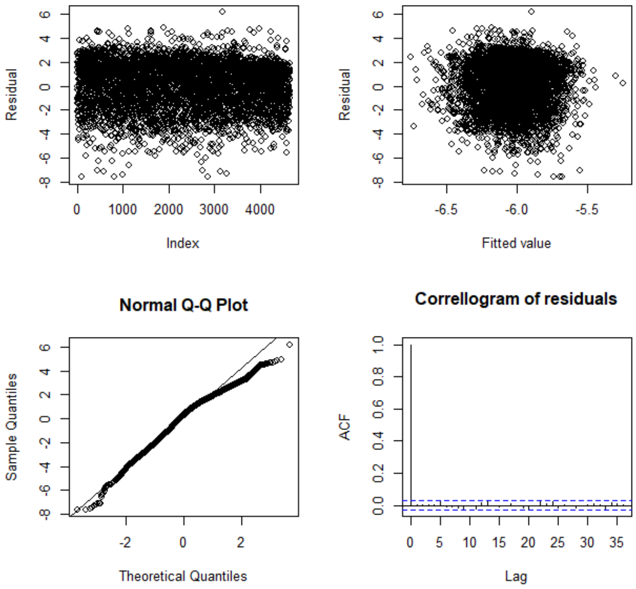


FIGURE A.78: ARMA diagnostic plot for machine OP90-8

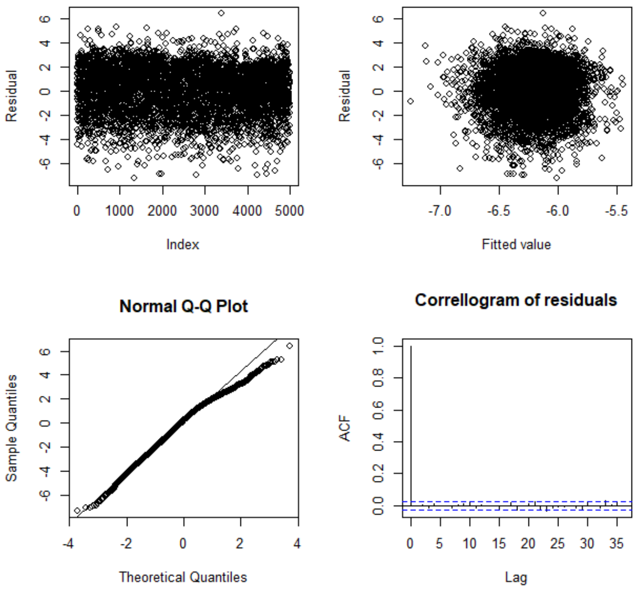


FIGURE A.79: ARMA diagnostic plot for machine OP90-9

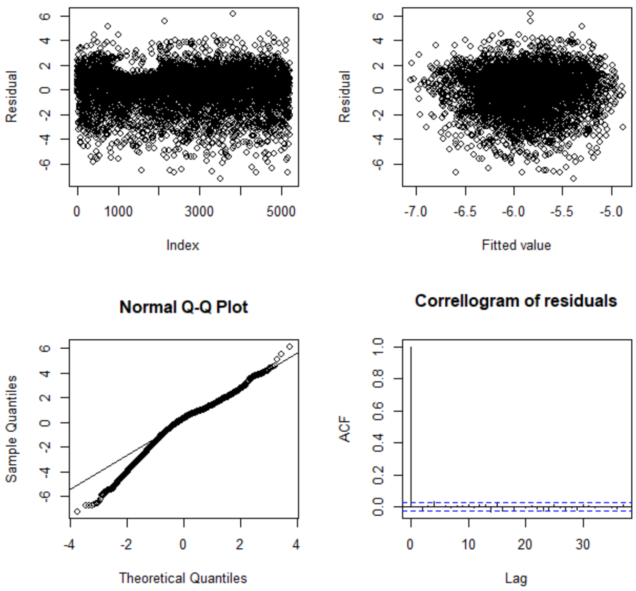


Figure A.80: ARMA diagnostic plot for machine OP90G-0

# Appendix B

# Diagnostic plots for ARMA-GARCH fits

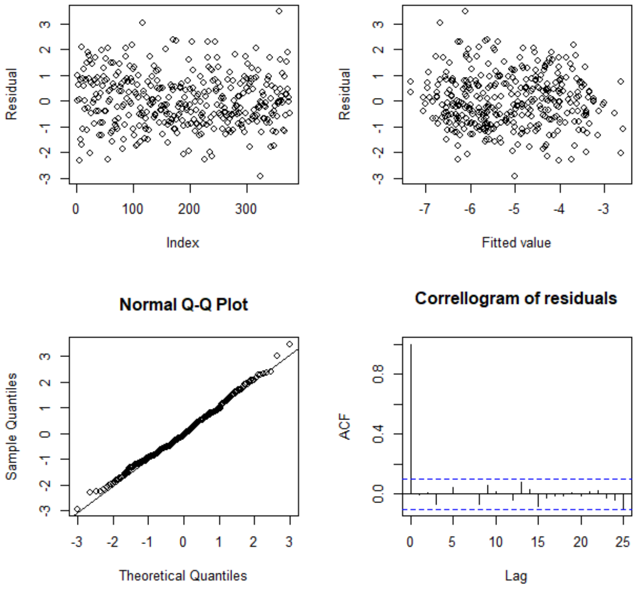


FIGURE B.1: ARMA-GARCH diagnostic plot for machine DAG0160-0

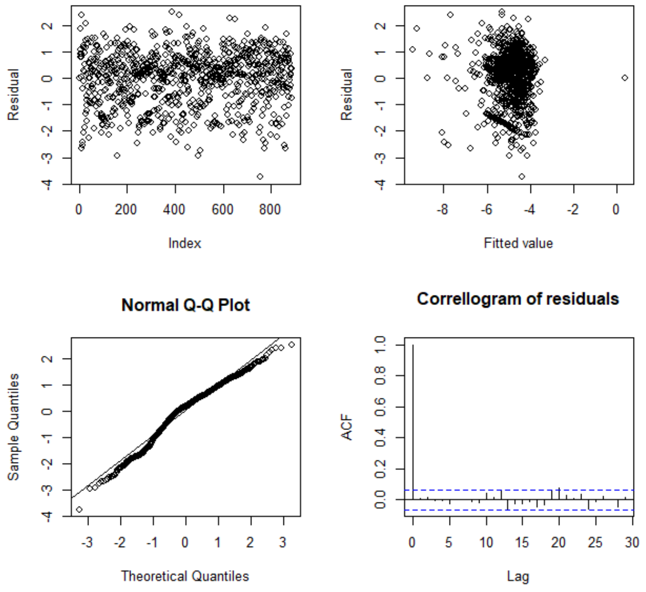


FIGURE B.2: ARMA-GARCH diagnostic plot for machine OP 100-2

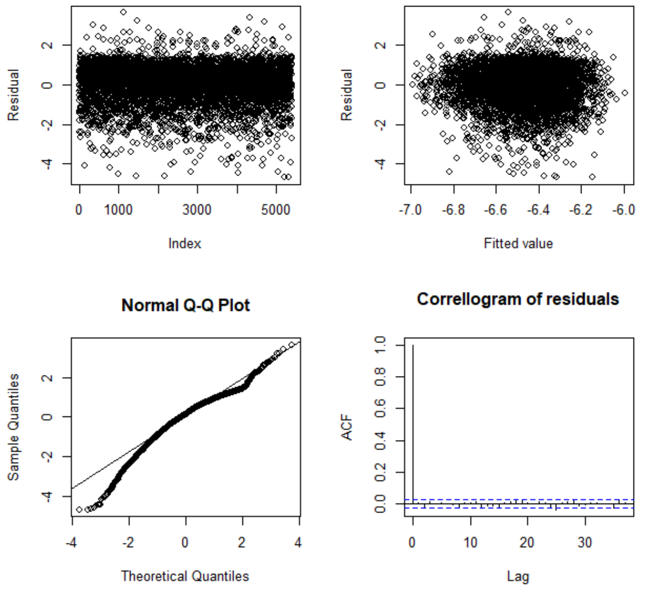


FIGURE B.3: ARMA-GARCH diagnostic plot for machine OP 10A-0

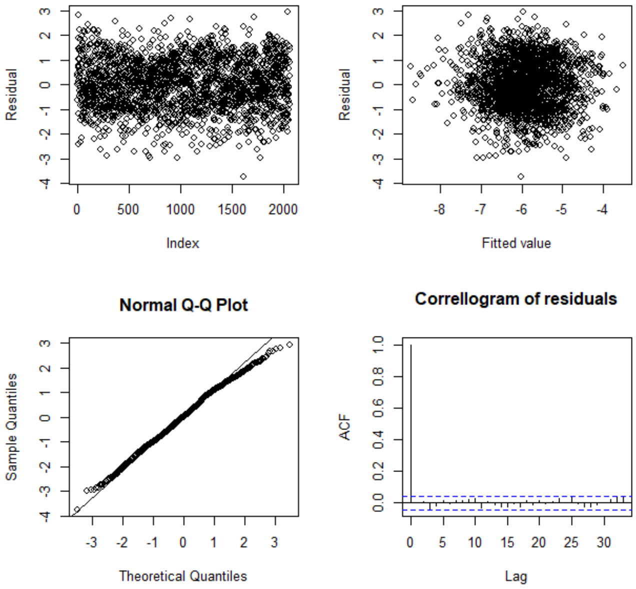


Figure B.4: ARMA-GARCH diagnostic plot for machine OP 110-0

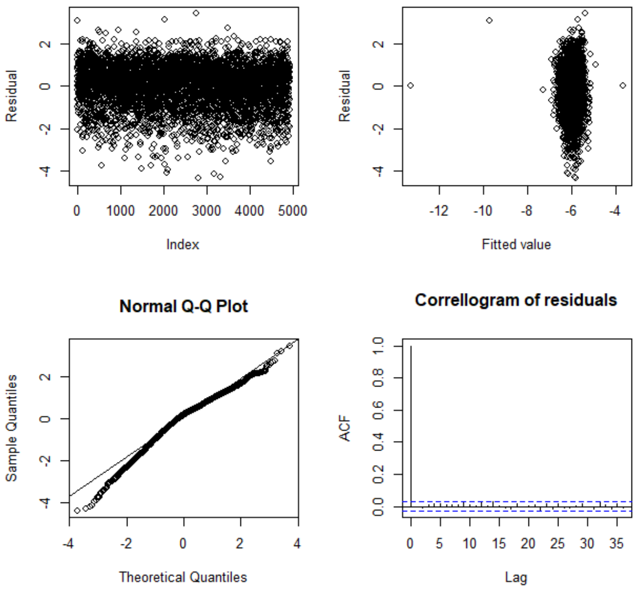


Figure B.5: ARMA-GARCH diagnostic plot for machine OP 125-0

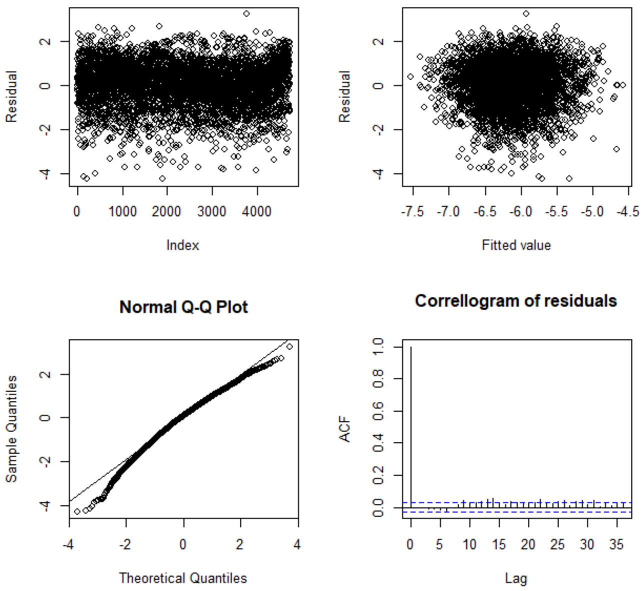


FIGURE B.6: ARMA-GARCH diagnostic plot for machine OP 135-0

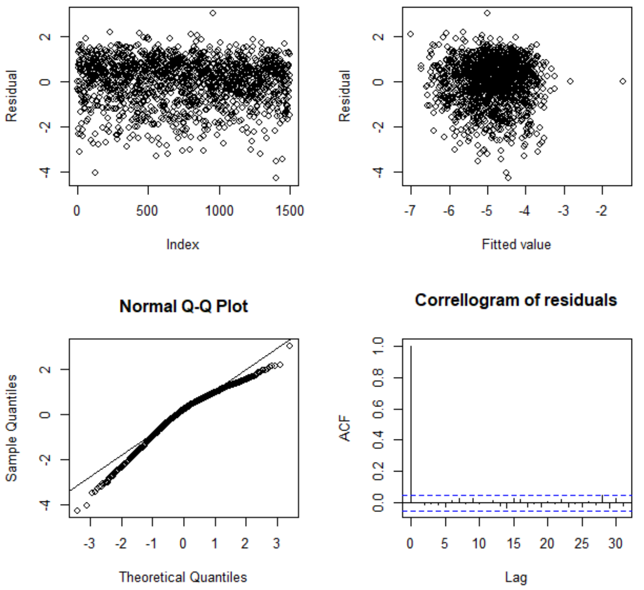


FIGURE B.7: ARMA-GARCH diagnostic plot for machine OP 140-0

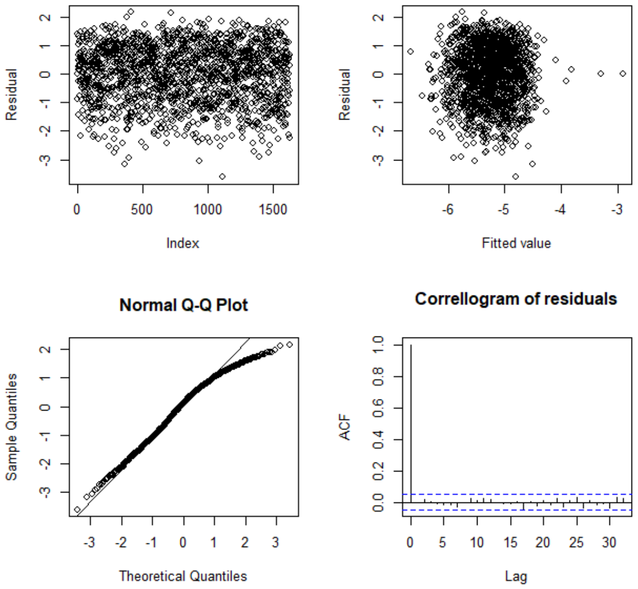


Figure B.8: ARMA-GARCH diagnostic plot for machine OP 150-0

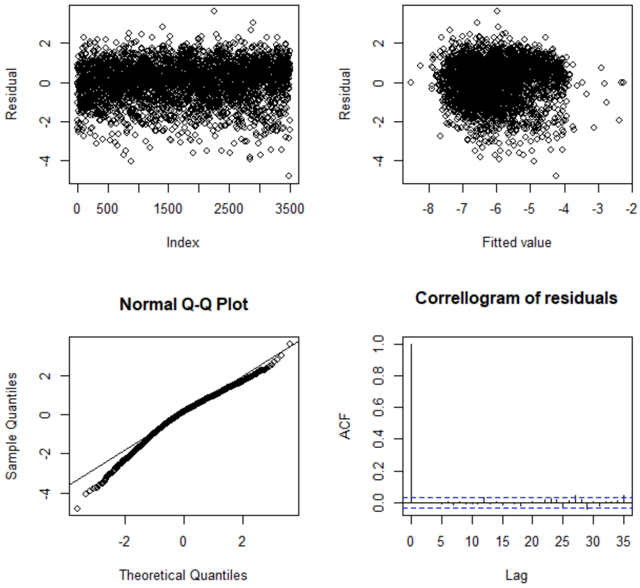


Figure B.9: ARMA-GARCH diagnostic plot for machine OP 160-0

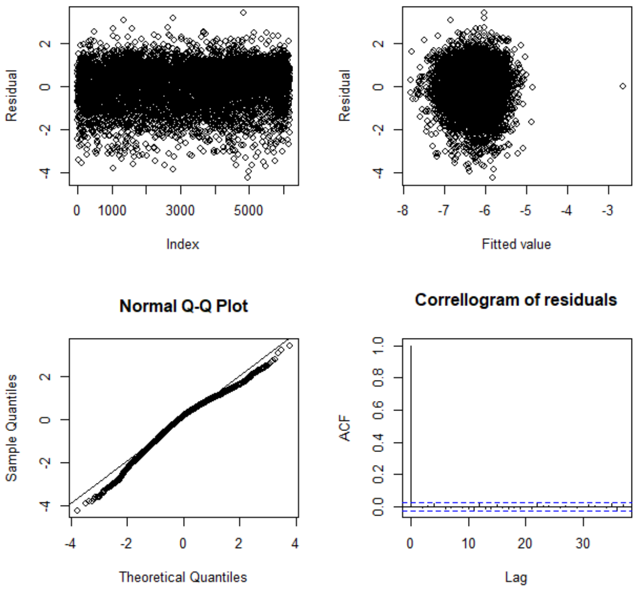


Figure B.10: ARMA-GARCH diagnostic plot for machine OP 170_SCR-0

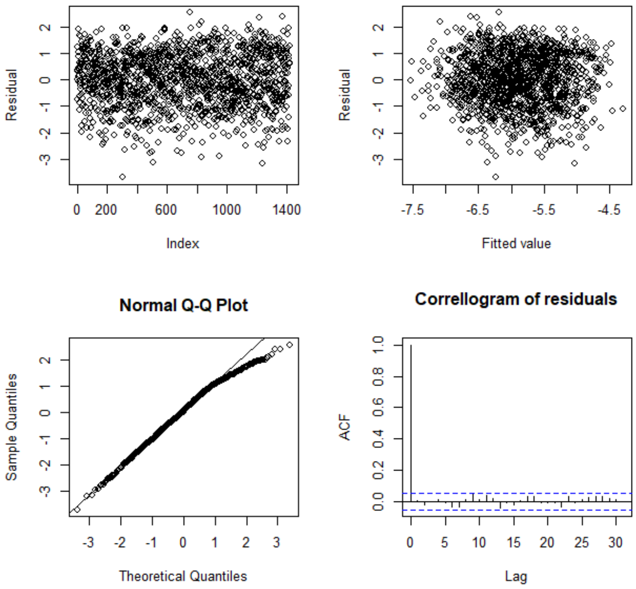


Figure B.11: ARMA-GARCH diagnostic plot for machine OP 30A-0

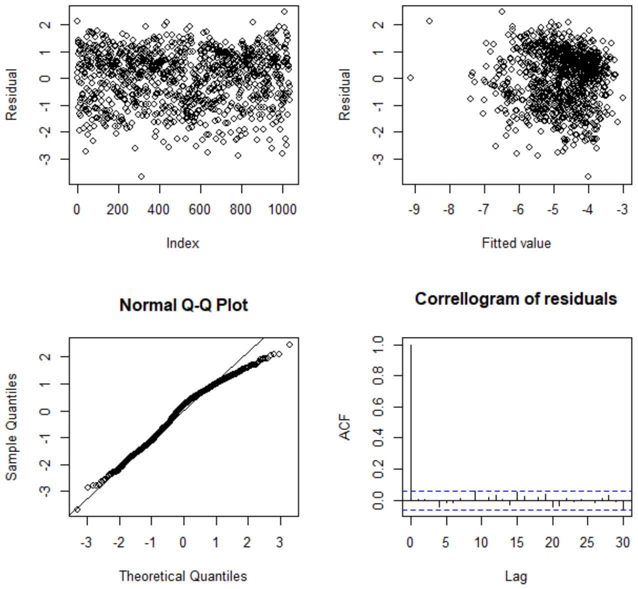


Figure B.12: ARMA-GARCH diagnostic plot for machine OP 40-0

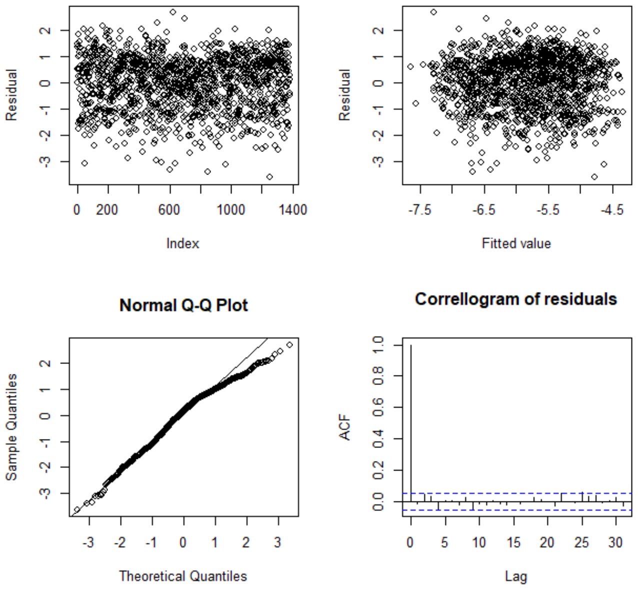


Figure B.13: ARMA-GARCH diagnostic plot for machine OP 45A-0

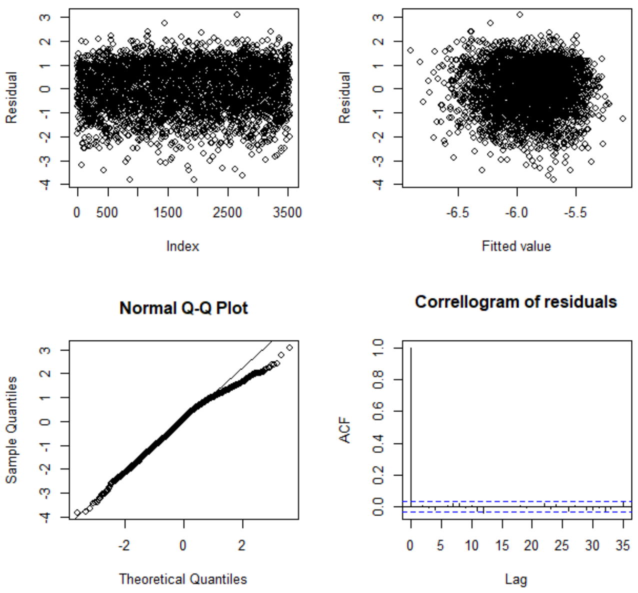


FIGURE B.14: ARMA-GARCH diagnostic plot for machine OP 70-6

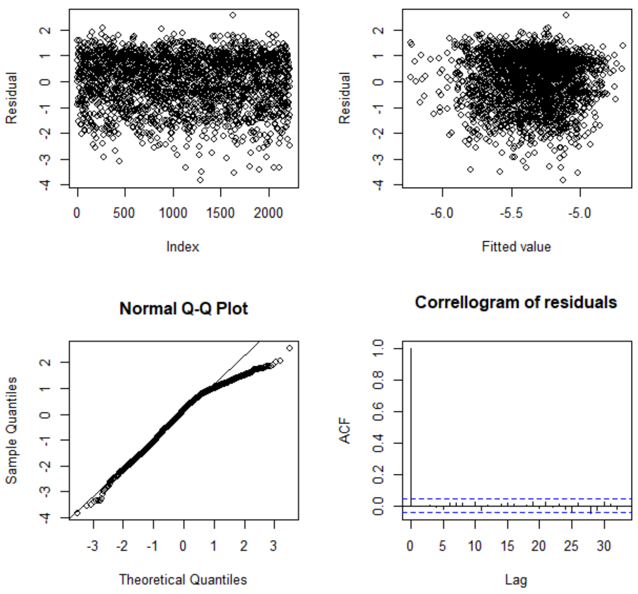


FIGURE B.15: ARMA-GARCH diagnostic plot for machine OP10-1

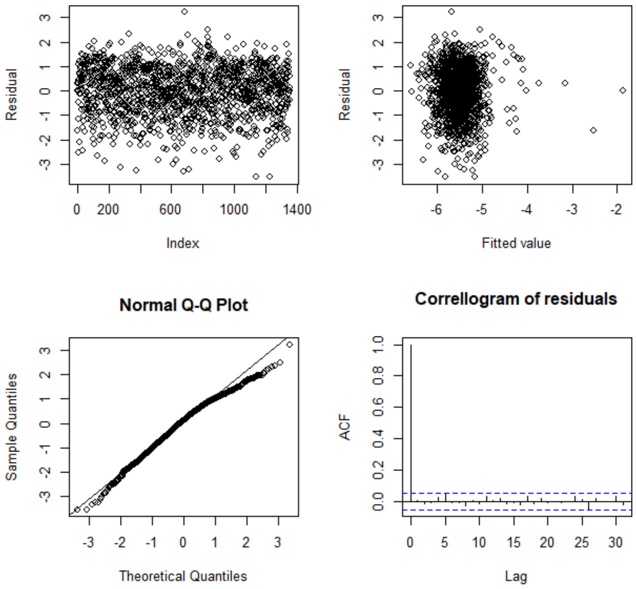


Figure B.16: ARMA-GARCH diagnostic plot for machine OP10-2

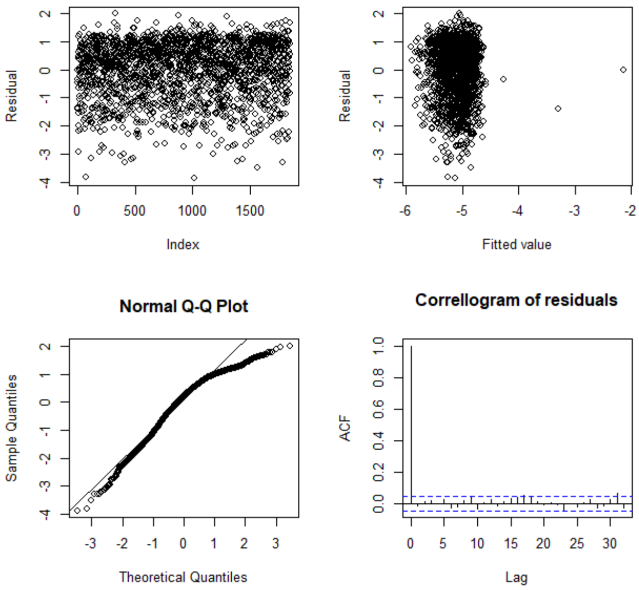


Figure B.17: ARMA-GARCH diagnostic plot for machine OP10-3

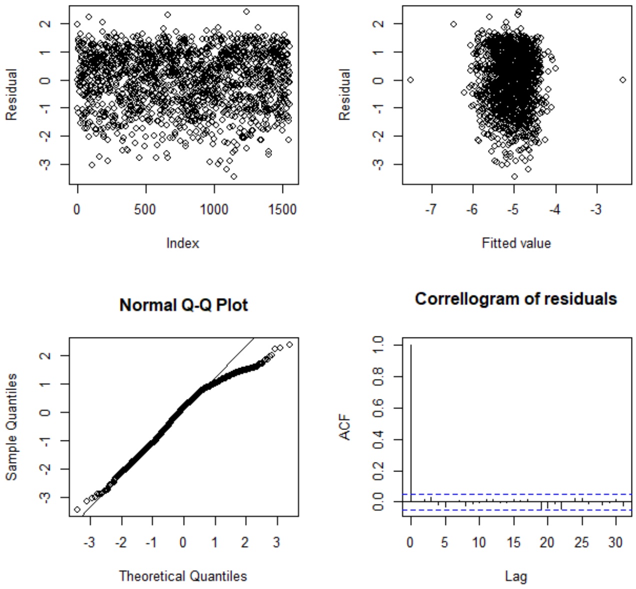


FIGURE B.18: ARMA-GARCH diagnostic plot for machine OP10-4

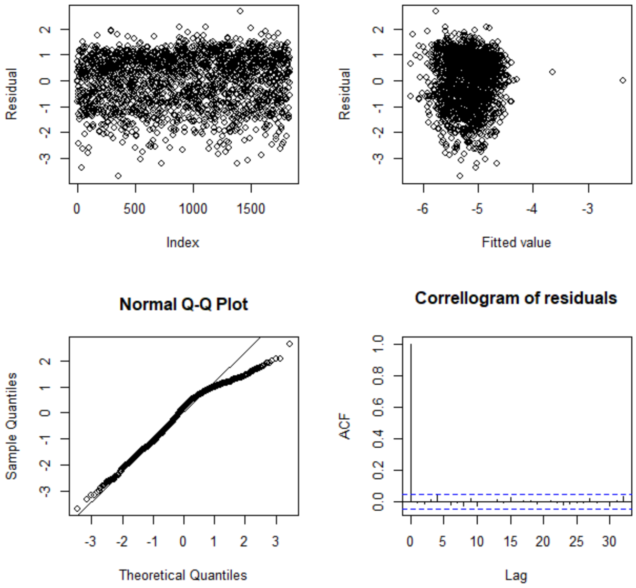


FIGURE B.19: ARMA-GARCH diagnostic plot for machine OP10-5

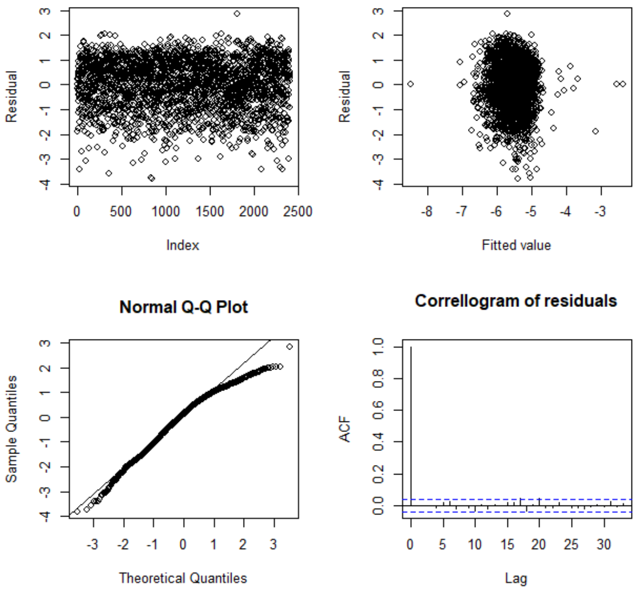


Figure B.20: ARMA-GARCH diagnostic plot for machine OP10-6

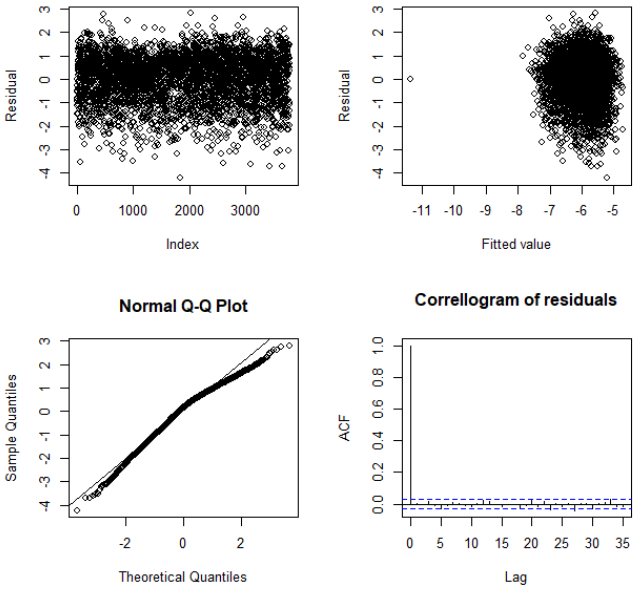


Figure B.21: ARMA-GARCH diagnostic plot for machine OP100G-0

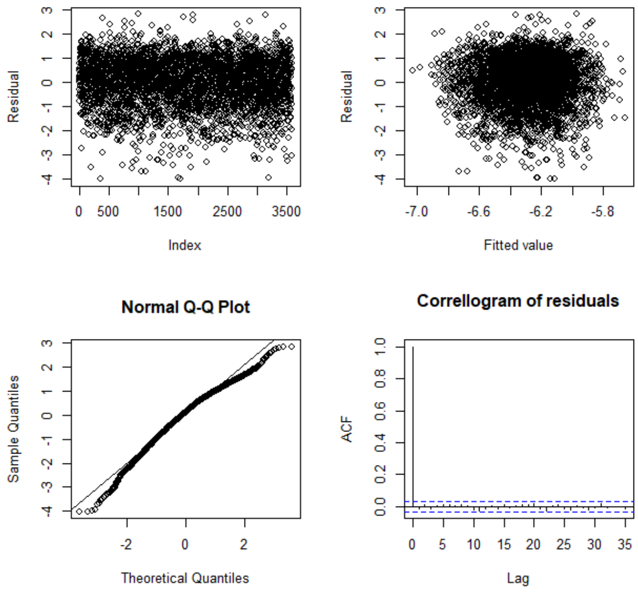


FIGURE B.22: ARMA-GARCH diagnostic plot for machine OP10A-30AG-0

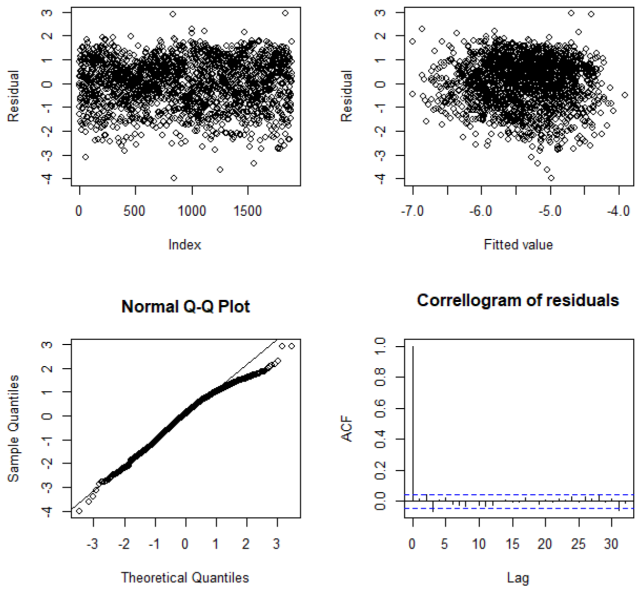


FIGURE B.23: ARMA-GARCH diagnostic plot for machine OP10G-0

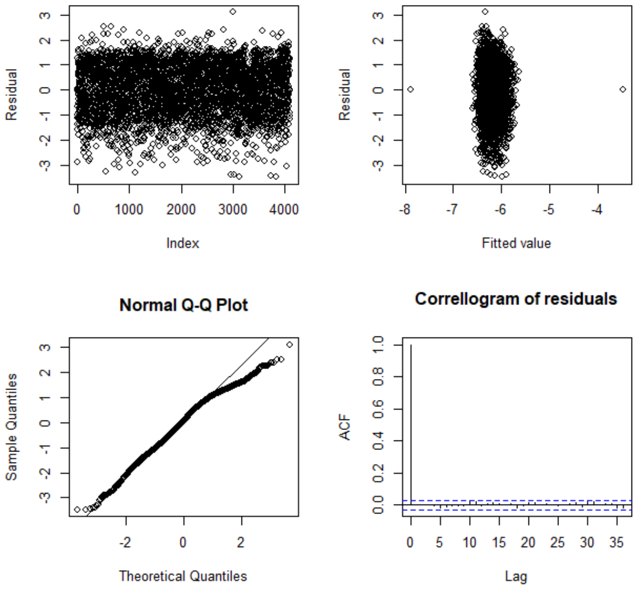


FIGURE B.24: ARMA-GARCH diagnostic plot for machine OP120-1

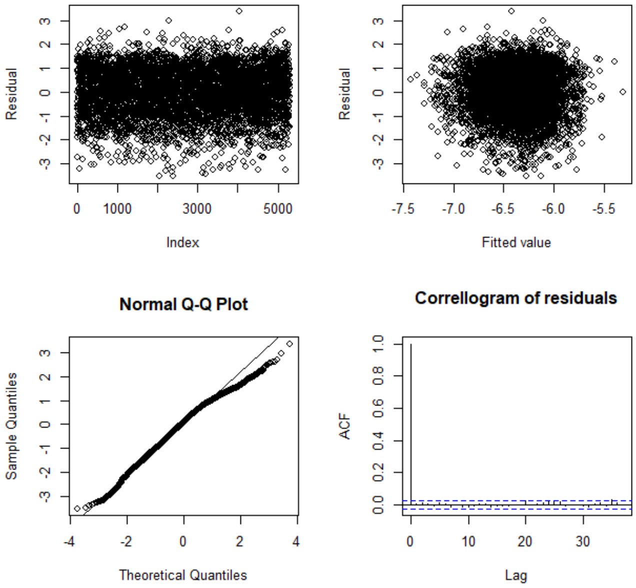


FIGURE B.25: ARMA-GARCH diagnostic plot for machine OP120-2

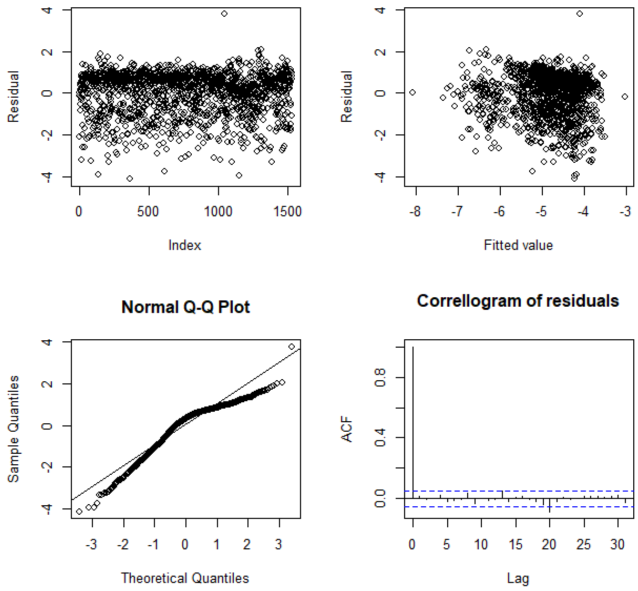


Figure B.26: ARMA-GARCH diagnostic plot for machine OP130-1

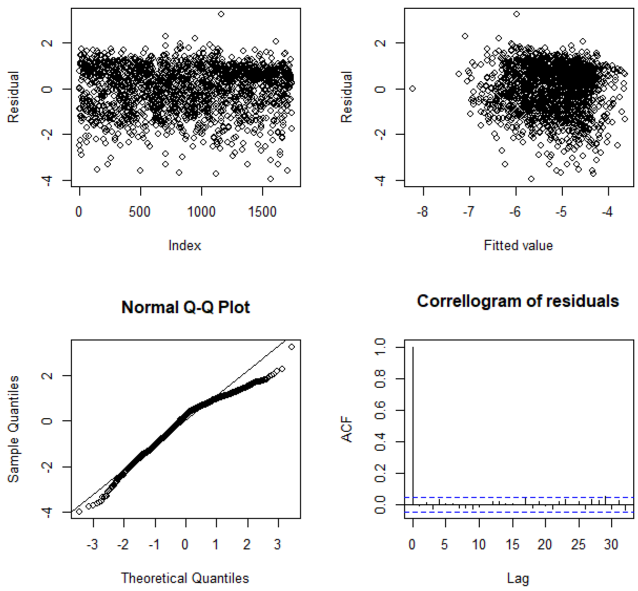


Figure B.27: ARMA-GARCH diagnostic plot for machine OP130-2

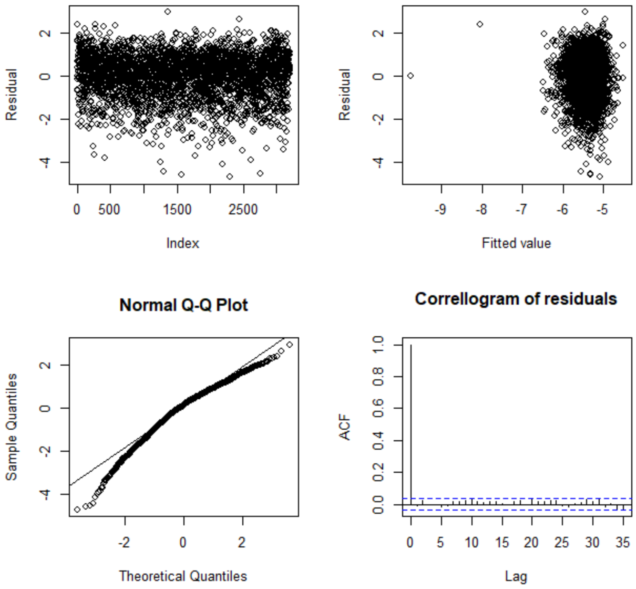


Figure B.28: ARMA-GARCH diagnostic plot for machine OP130G-0

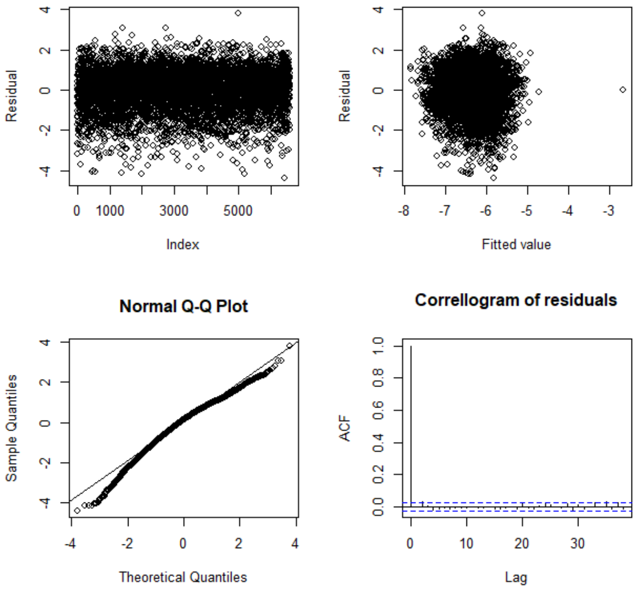


Figure B.29: ARMA-GARCH diagnostic plot for machine OP170_LEAK-0

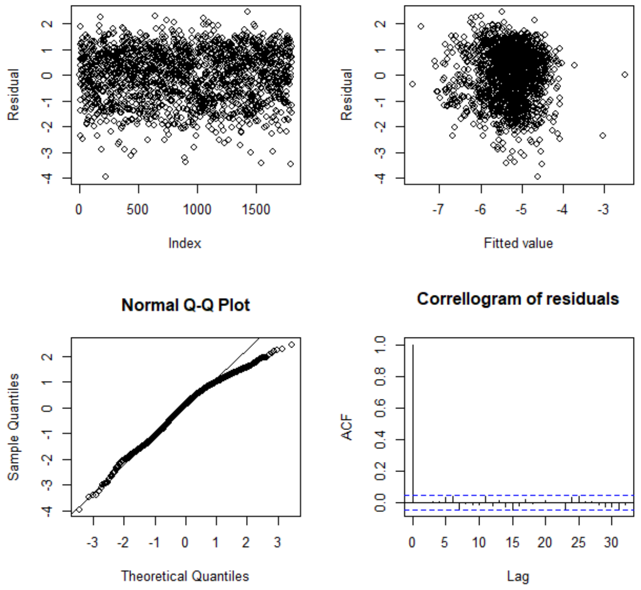


Figure B.30: ARMA-GARCH diagnostic plot for machine OP20-1

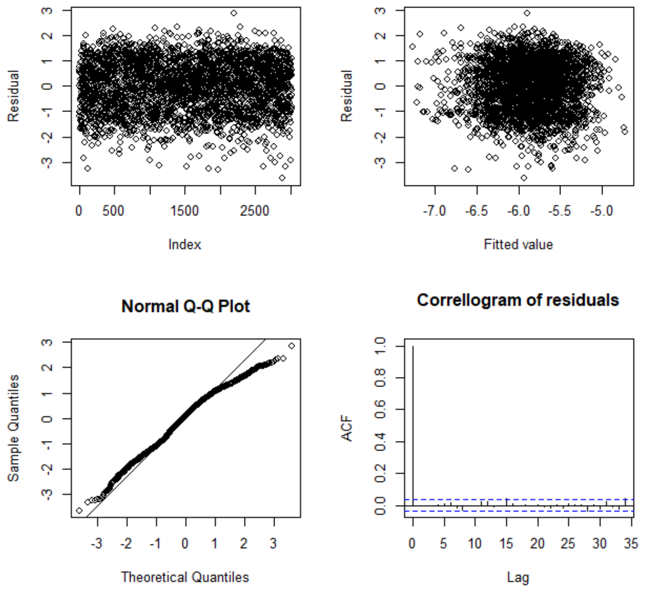


Figure B.31: ARMA-GARCH diagnostic plot for machine OP20-10

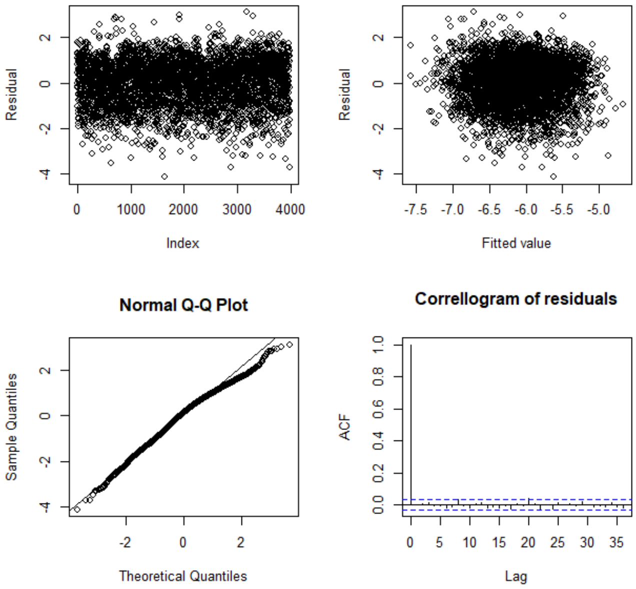


Figure B.32: ARMA-GARCH diagnostic plot for machine OP20-11

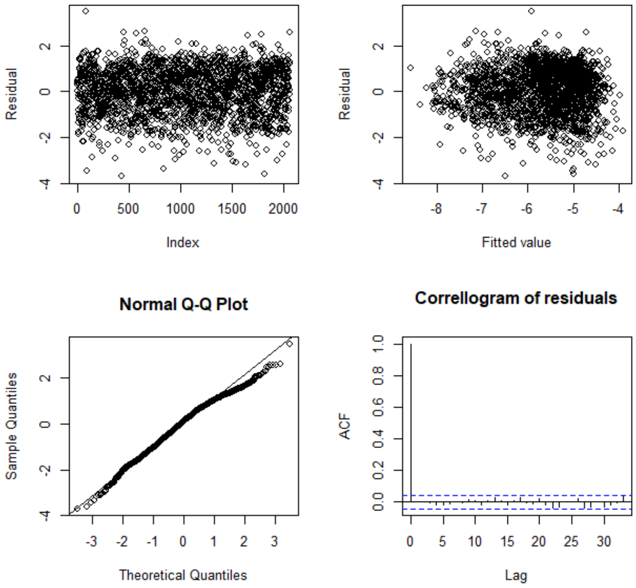


Figure B.33: ARMA-GARCH diagnostic plot for machine OP20-12

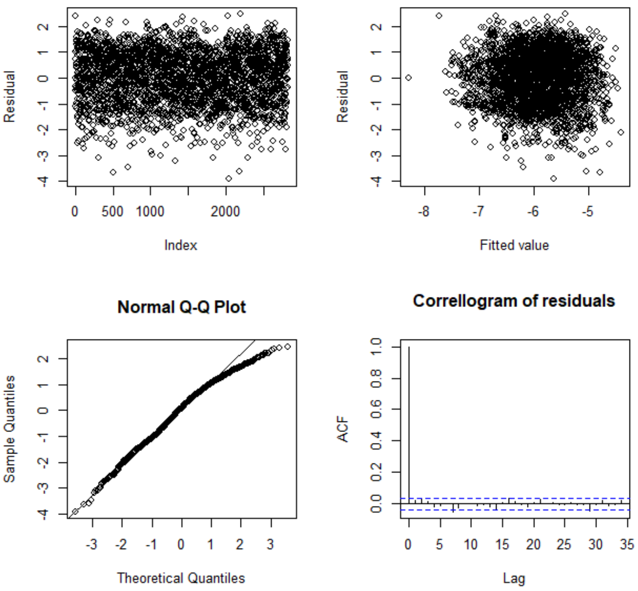


FIGURE B.34: ARMA-GARCH diagnostic plot for machine OP20-2

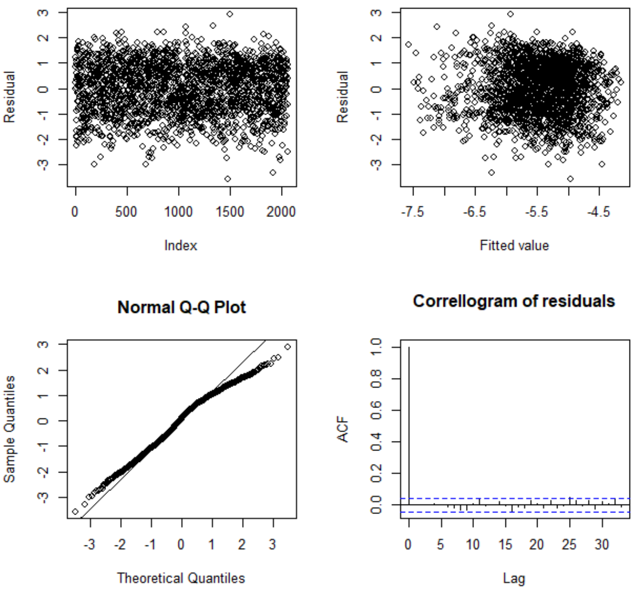


FIGURE B.35: ARMA-GARCH diagnostic plot for machine OP20-3

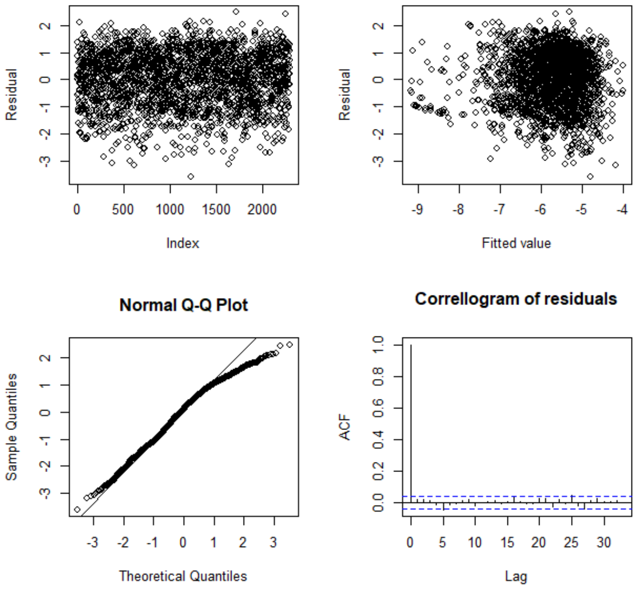


Figure B.36: ARMA-GARCH diagnostic plot for machine OP20-4

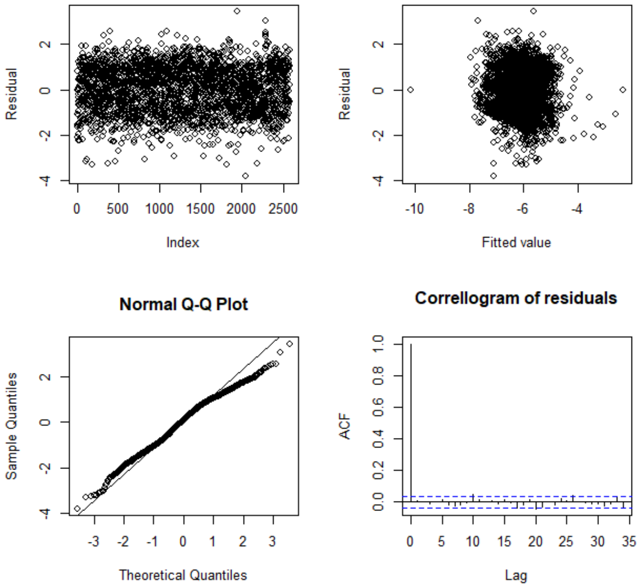


Figure B.37: ARMA-GARCH diagnostic plot for machine OP20-5

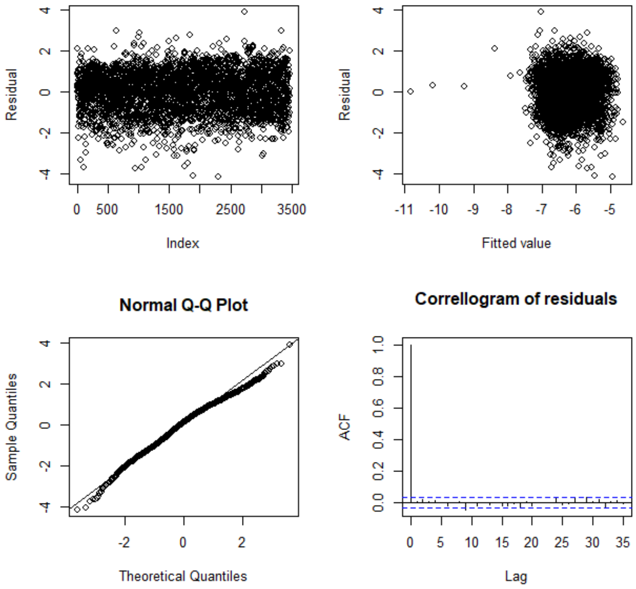


Figure B.38: ARMA-GARCH diagnostic plot for machine OP20-6

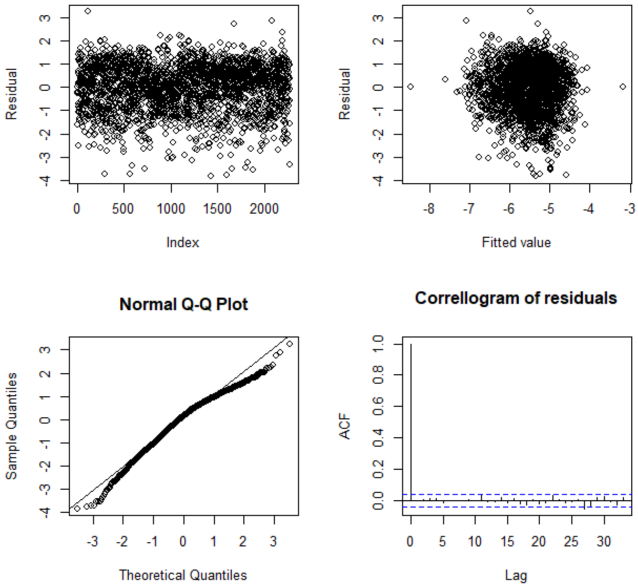


Figure B.39: ARMA-GARCH diagnostic plot for machine OP20-7

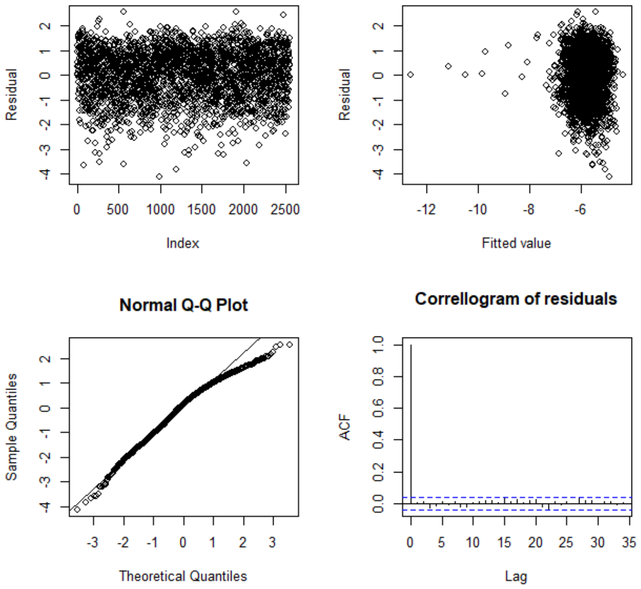


FIGURE B.40: ARMA-GARCH diagnostic plot for machine OP20-8

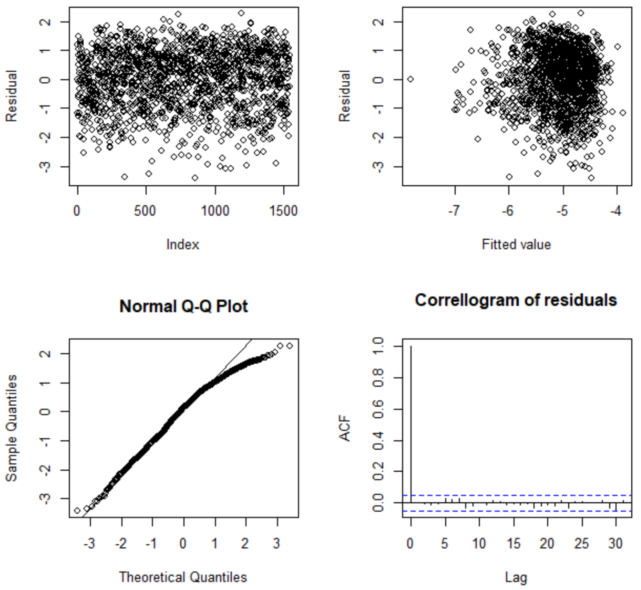


FIGURE B.41: ARMA-GARCH diagnostic plot for machine OP20-9

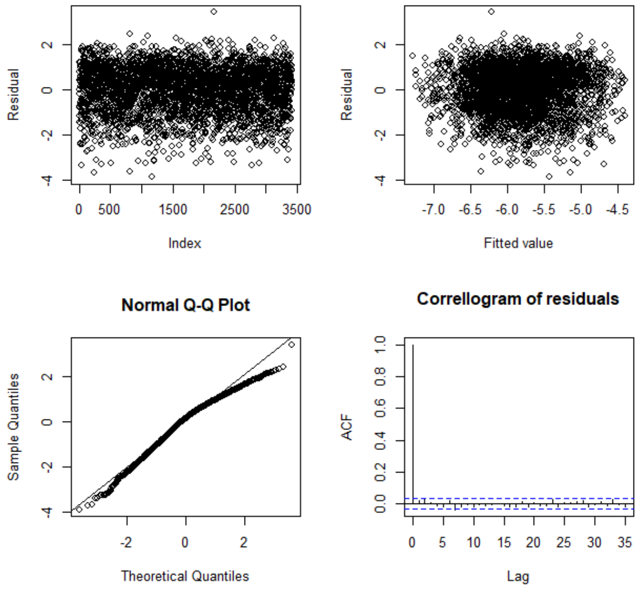


Figure B.42: ARMA-GARCH diagnostic plot for machine OP20G-0

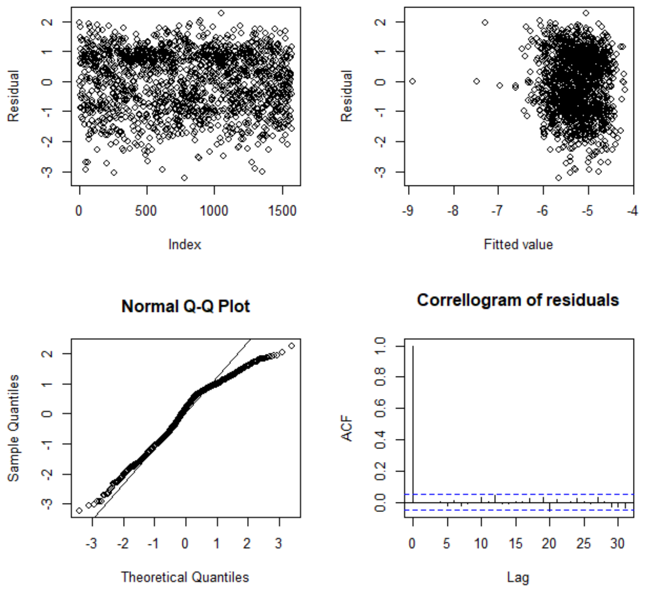


Figure B.43: ARMA-GARCH diagnostic plot for machine OP30-40G-0

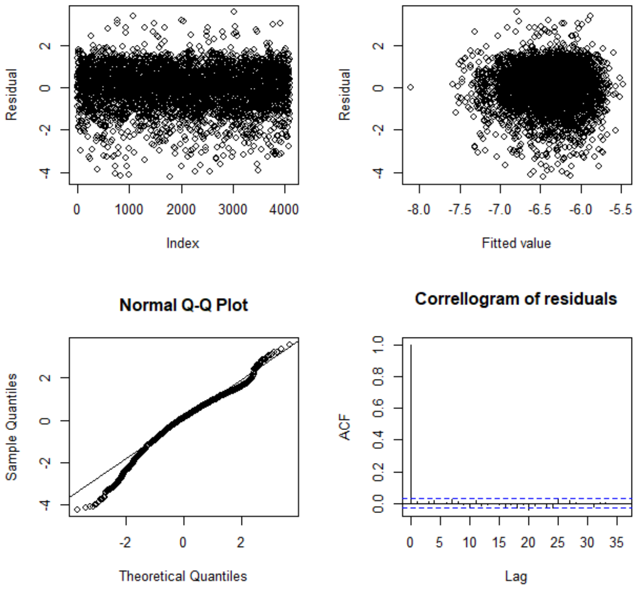


Figure B.44: ARMA-GARCH diagnostic plot for machine OP45-50C-0

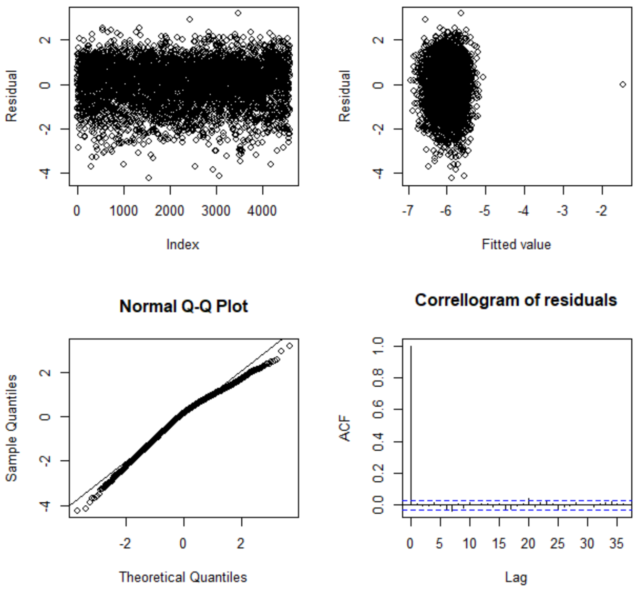


Figure B.45: ARMA-GARCH diagnostic plot for machine OP50_BLTRD-0

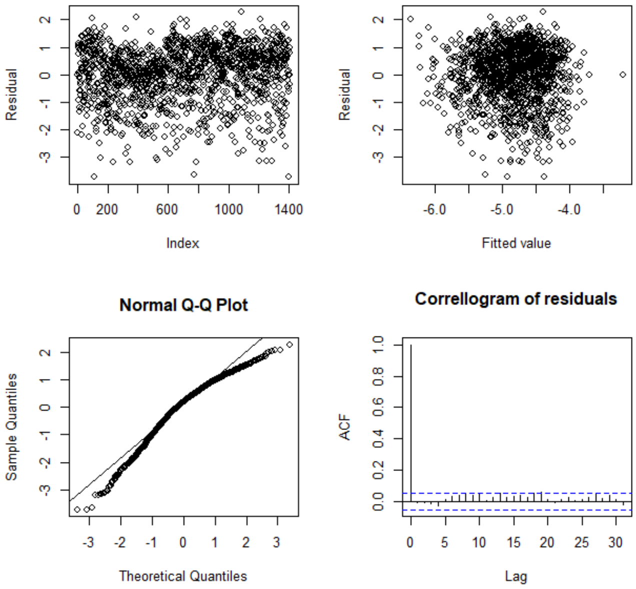


Figure B.46: ARMA-GARCH diagnostic plot for machine OP55-0

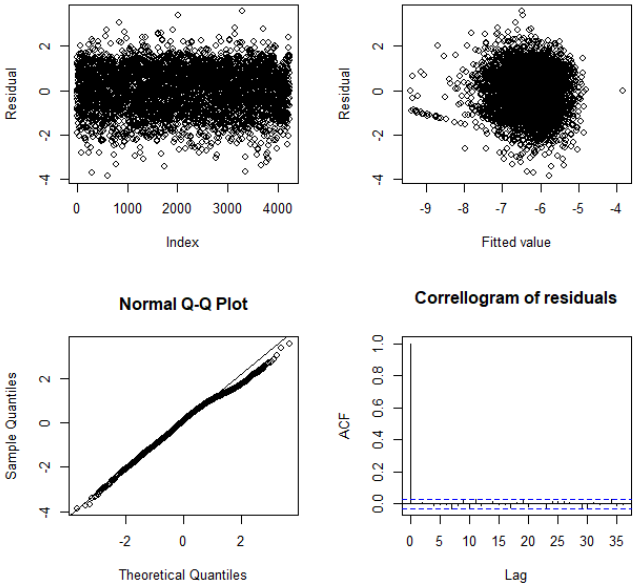


Figure B.47: ARMA-GARCH diagnostic plot for machine OP70-1

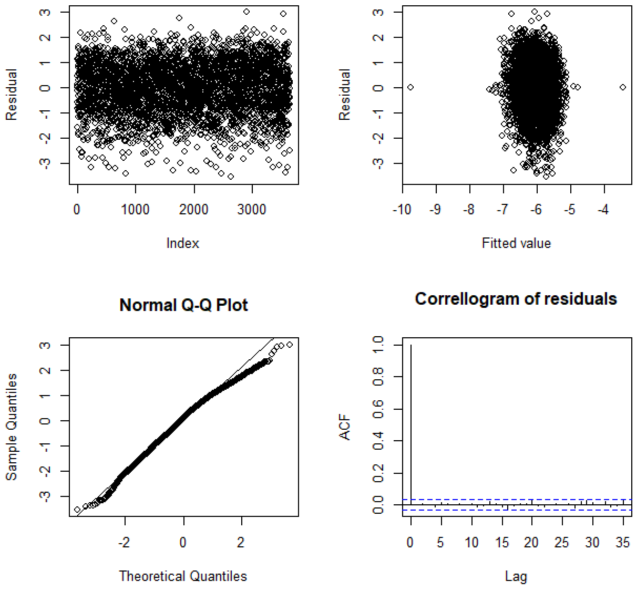


Figure B.48: ARMA-GARCH diagnostic plot for machine OP70-10

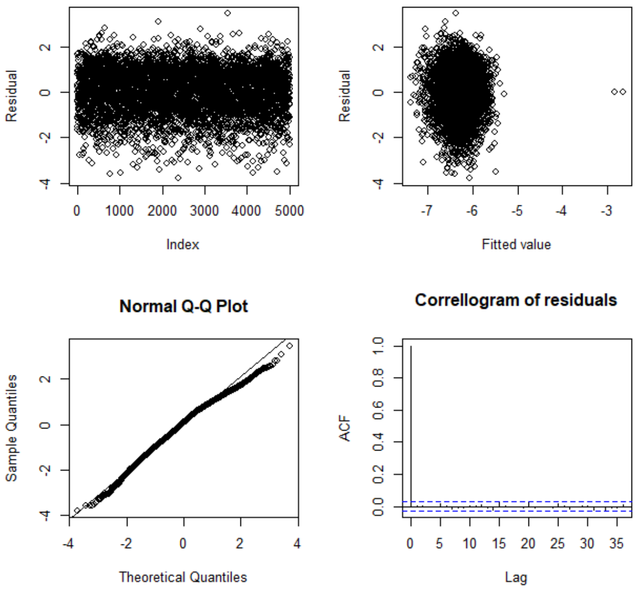


Figure B.49: ARMA-GARCH diagnostic plot for machine OP70-12

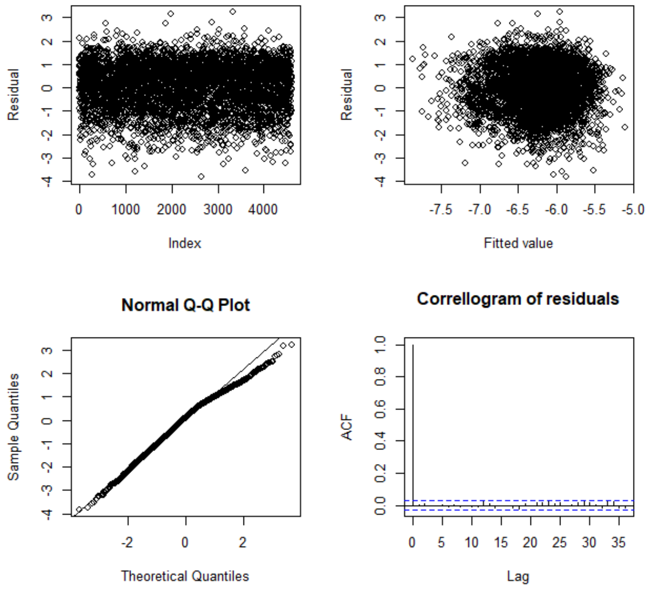


Figure B.50: ARMA-GARCH diagnostic plot for machine OP70-2

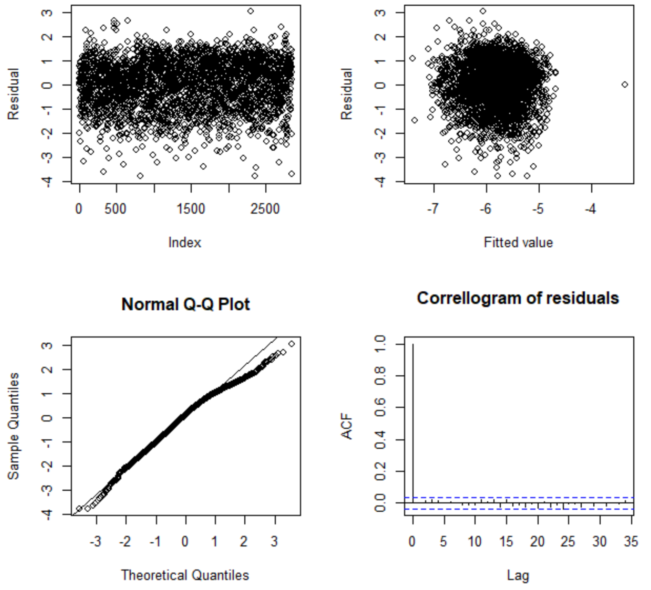


Figure B.51: ARMA-GARCH diagnostic plot for machine OP70-3

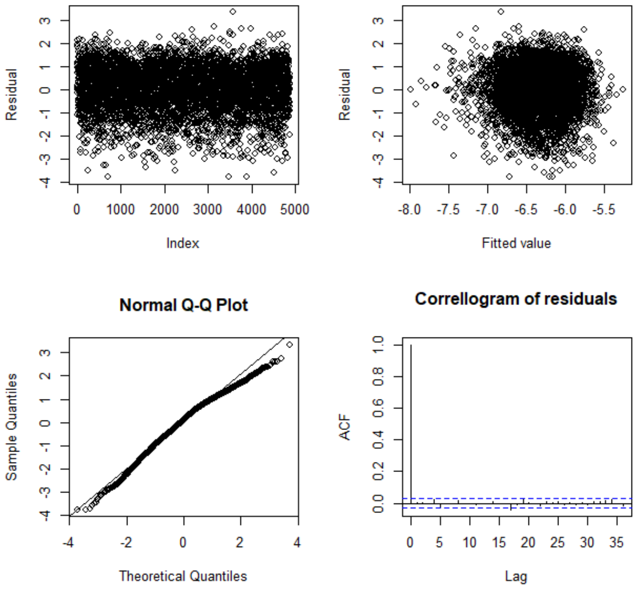


Figure B.52: ARMA-GARCH diagnostic plot for machine OP70-4

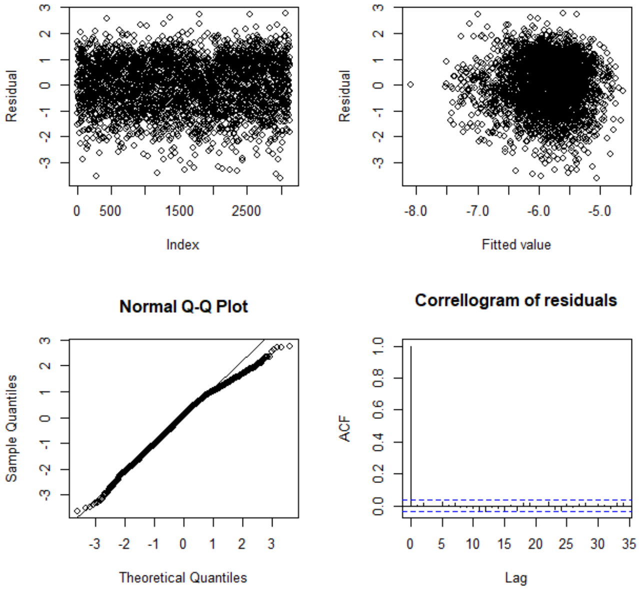


Figure B.53: ARMA-GARCH diagnostic plot for machine OP70-5

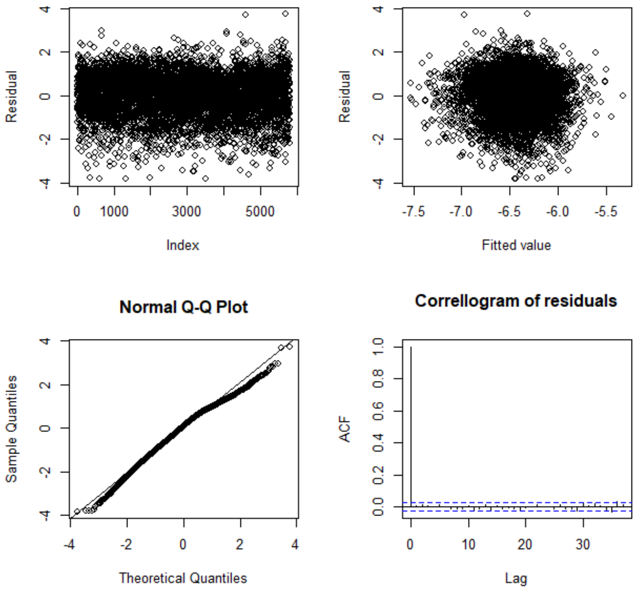


Figure B.54: ARMA-GARCH diagnostic plot for machine OP70-7

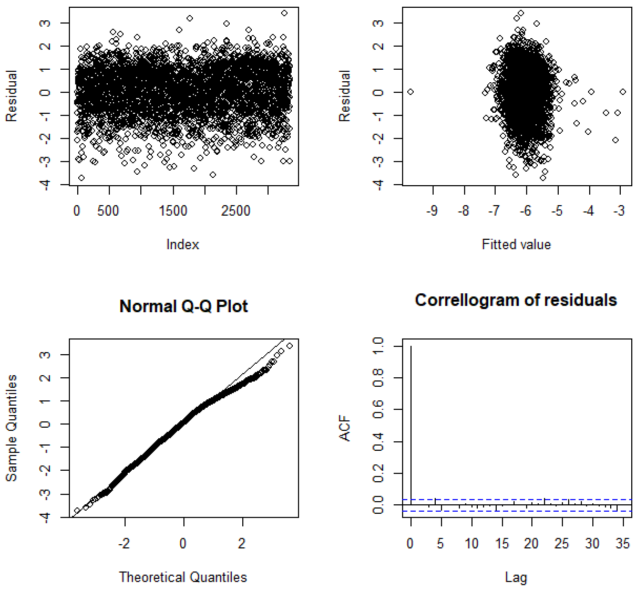


Figure B.55: ARMA-GARCH diagnostic plot for machine OP70-8

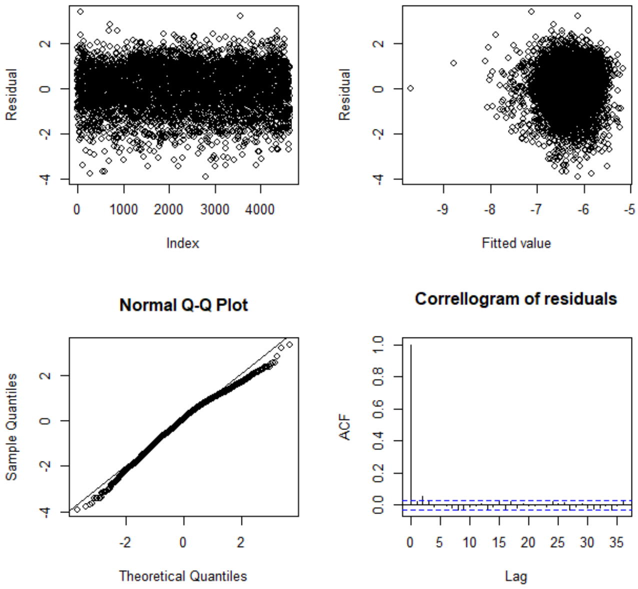


FIGURE B.56: ARMA-GARCH diagnostic plot for machine OP70-9

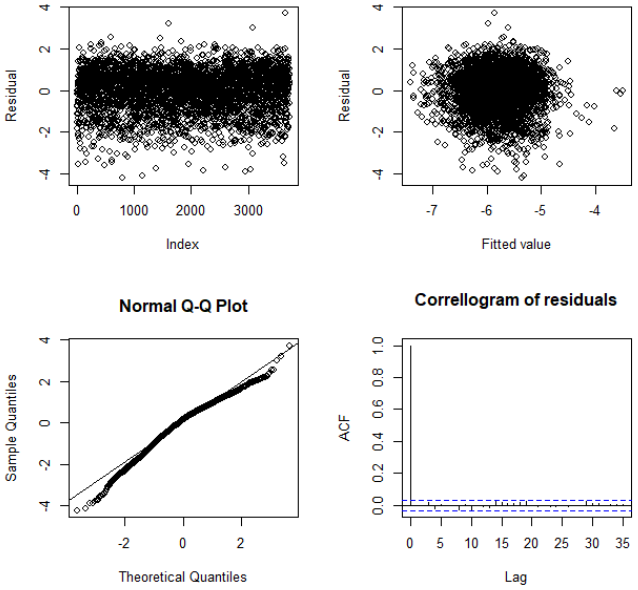


FIGURE B.57: ARMA-GARCH diagnostic plot for machine OP70G-0

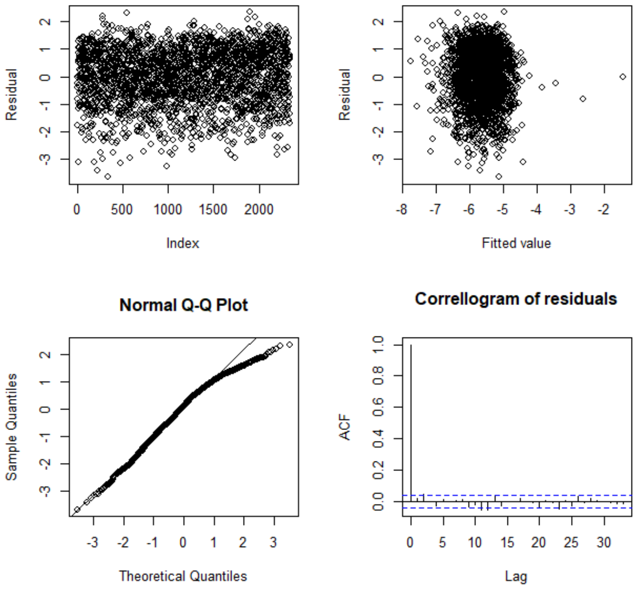


Figure B.58: ARMA-GARCH diagnostic plot for machine OP80-1

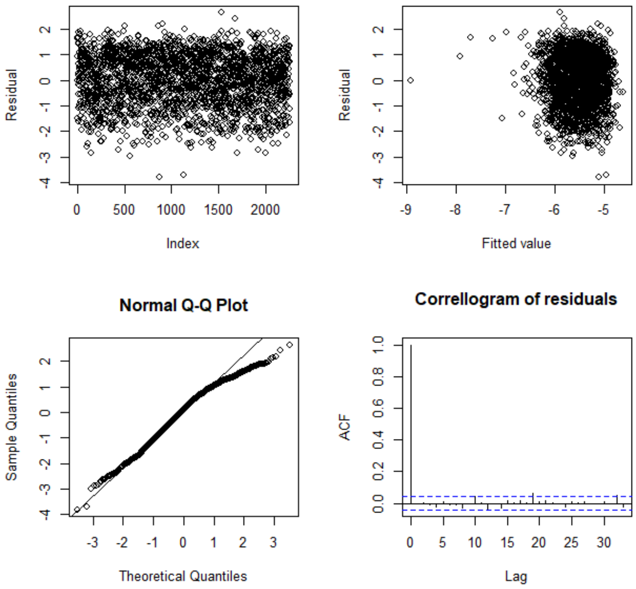


Figure B.59: ARMA-GARCH diagnostic plot for machine OP80-10

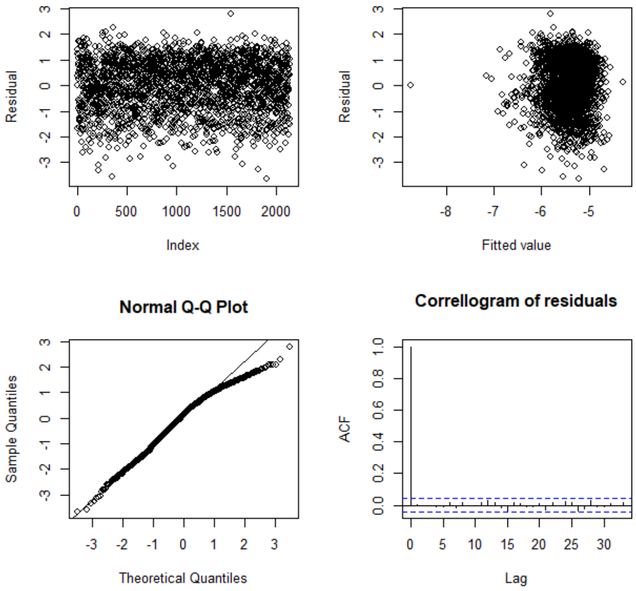


Figure B.60: ARMA-GARCH diagnostic plot for machine OP80-2

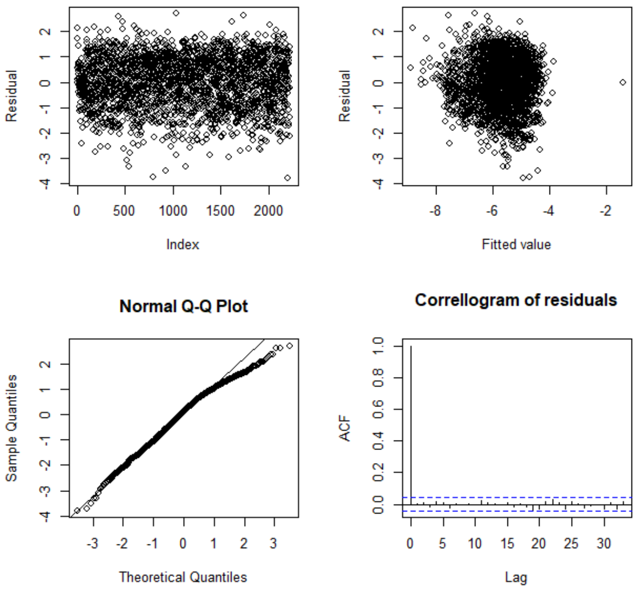


Figure B.61: ARMA-GARCH diagnostic plot for machine OP80-3

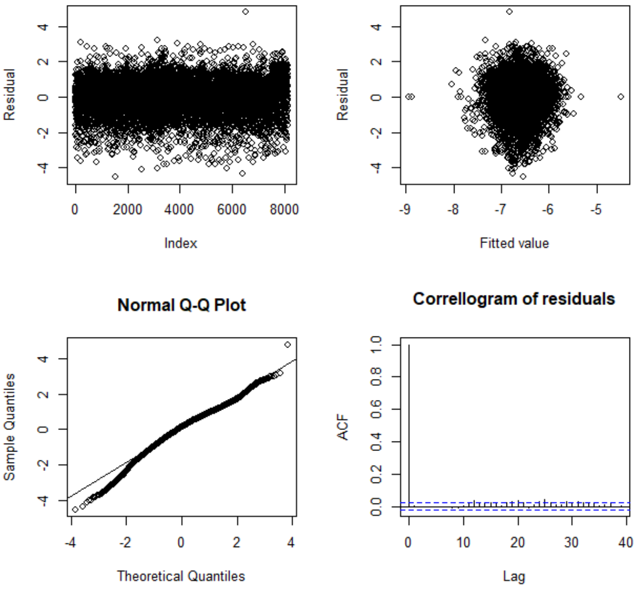


Figure B.62: ARMA-GARCH diagnostic plot for machine OP80-4

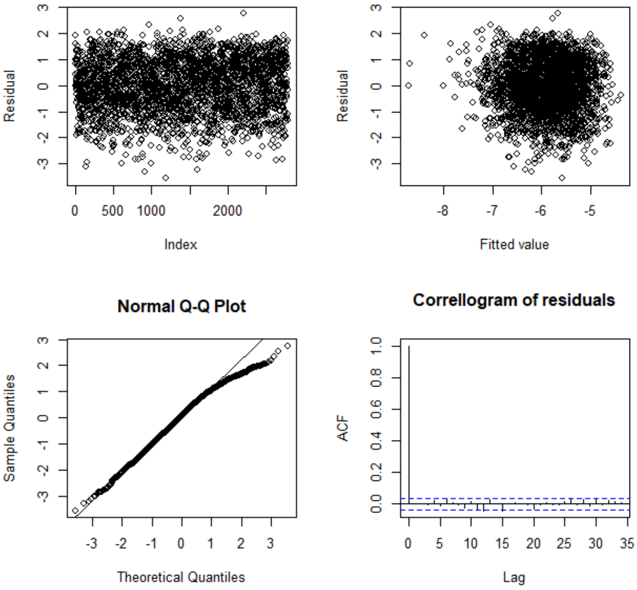


Figure B.63: ARMA-GARCH diagnostic plot for machine OP80-5

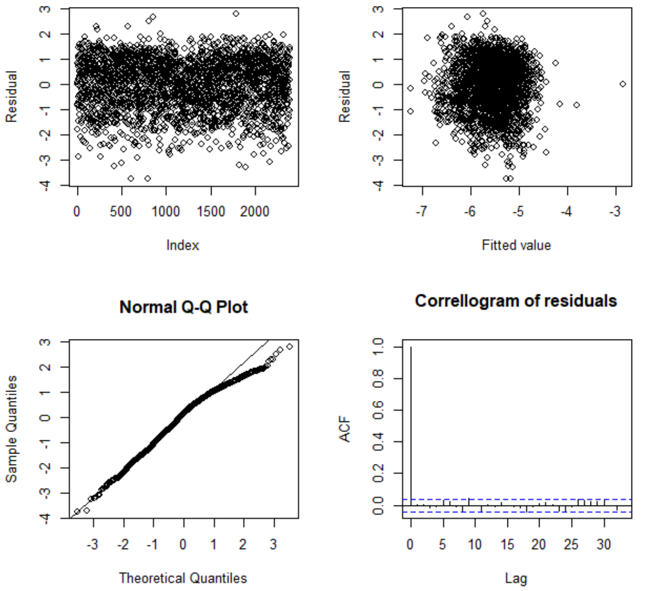


Figure B.64: ARMA-GARCH diagnostic plot for machine OP80-6

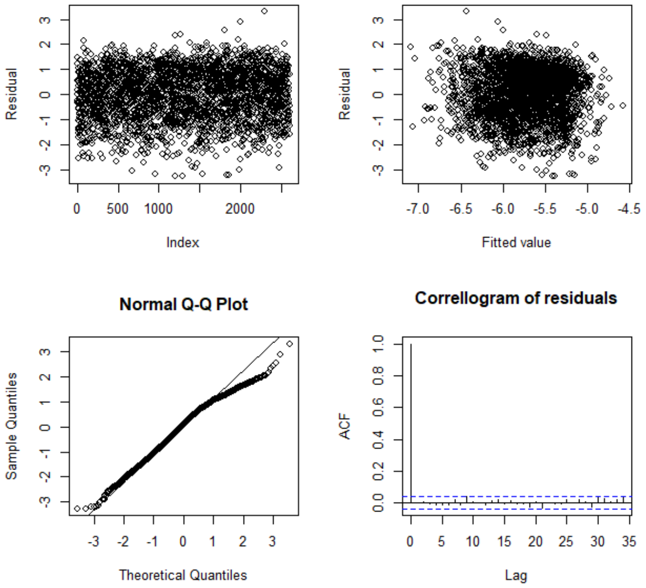


Figure B.65: ARMA-GARCH diagnostic plot for machine OP80-7

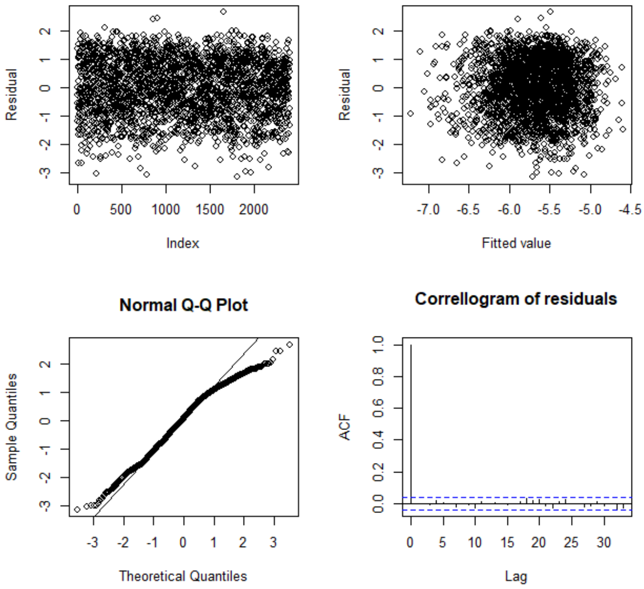


Figure B.66: ARMA-GARCH diagnostic plot for machine OP80-8

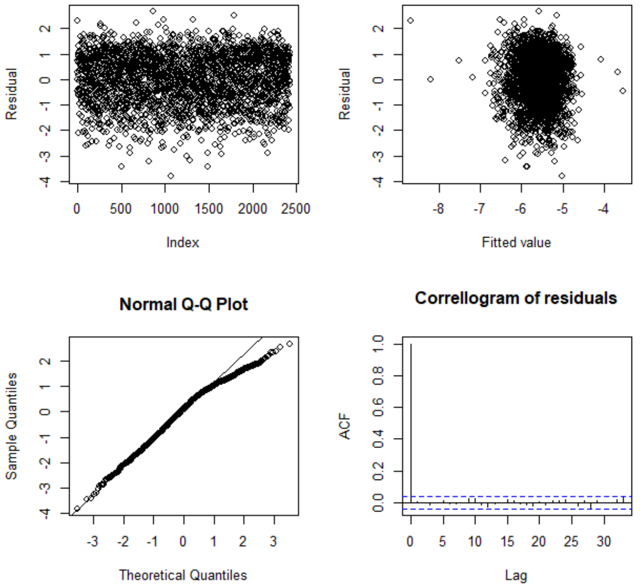


Figure B.67: ARMA-GARCH diagnostic plot for machine OP80-9

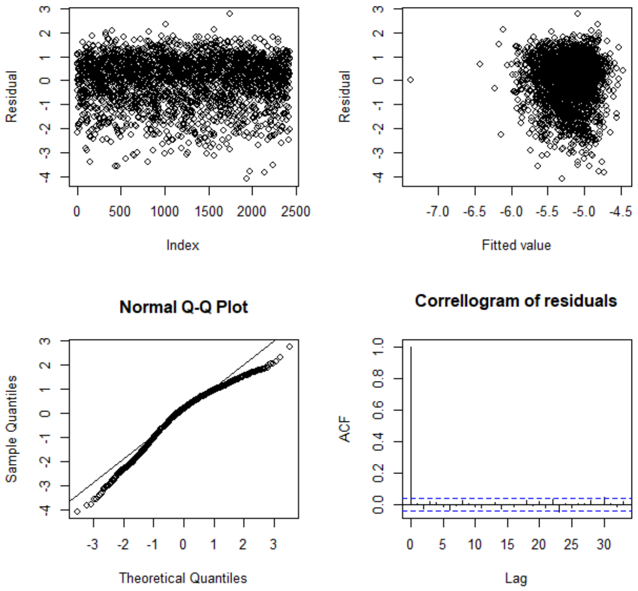


Figure B.68: ARMA-GARCH diagnostic plot for machine OP80G-0

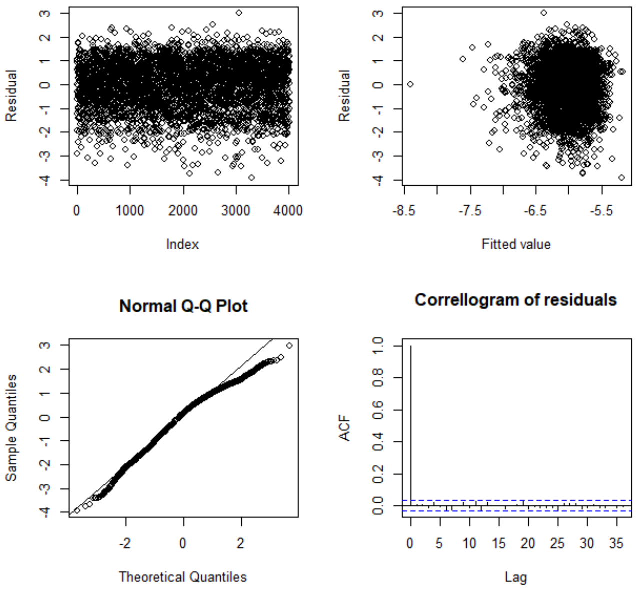


Figure B.69: ARMA-GARCH diagnostic plot for machine OP90-1

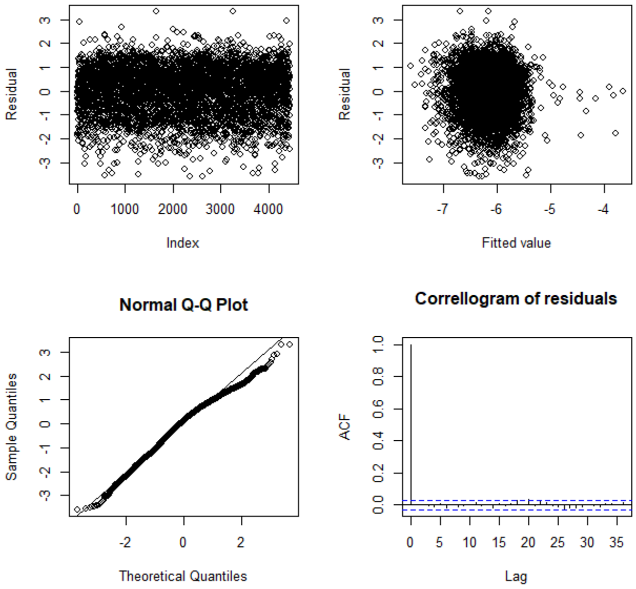


FIGURE B.70: ARMA-GARCH diagnostic plot for machine OP90-10

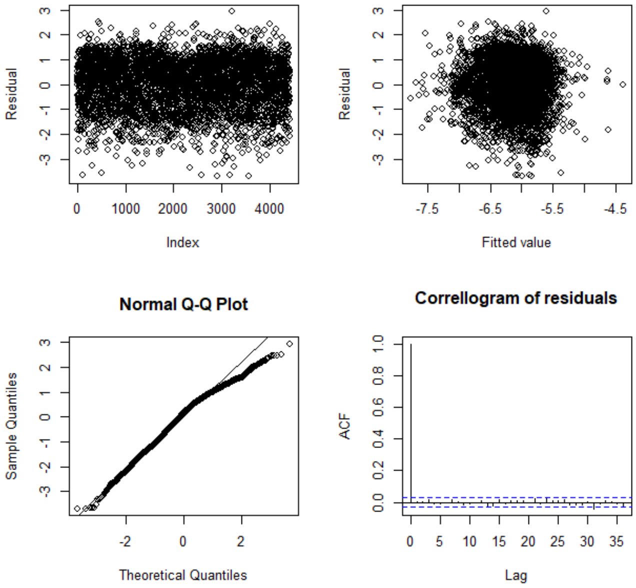


FIGURE B.71: ARMA-GARCH diagnostic plot for machine OP90-12

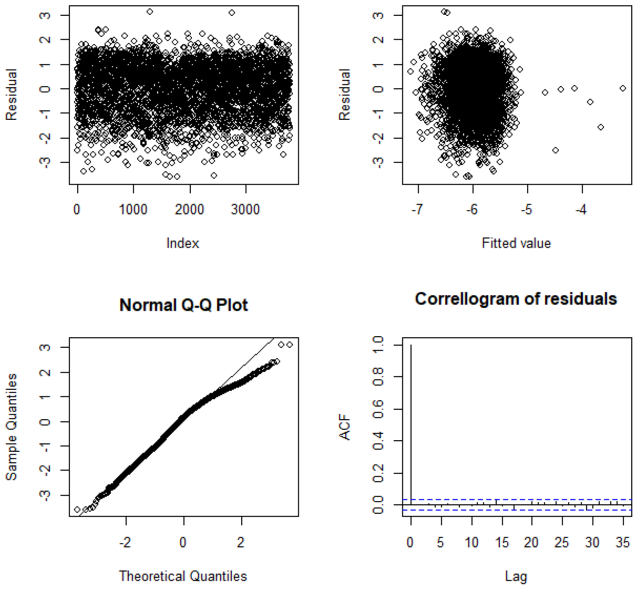


FIGURE B.72: ARMA-GARCH diagnostic plot for machine OP90-2

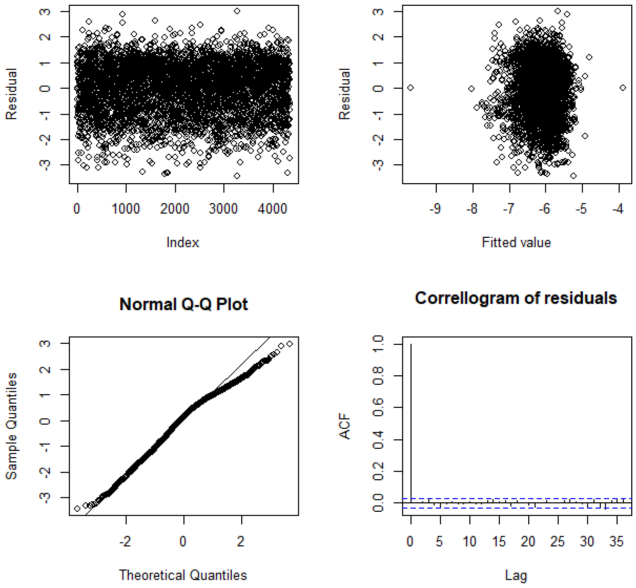


FIGURE B.73: ARMA-GARCH diagnostic plot for machine OP90-3

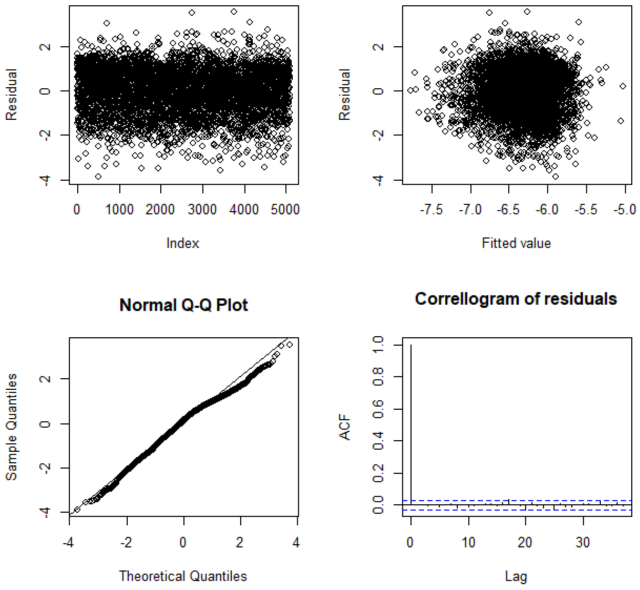


Figure B.74: ARMA-GARCH diagnostic plot for machine OP90-4

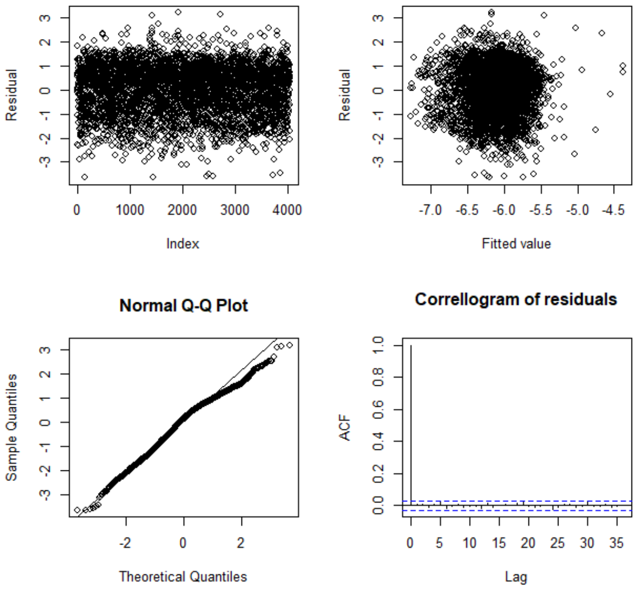


Figure B.75: ARMA-GARCH diagnostic plot for machine OP90-5

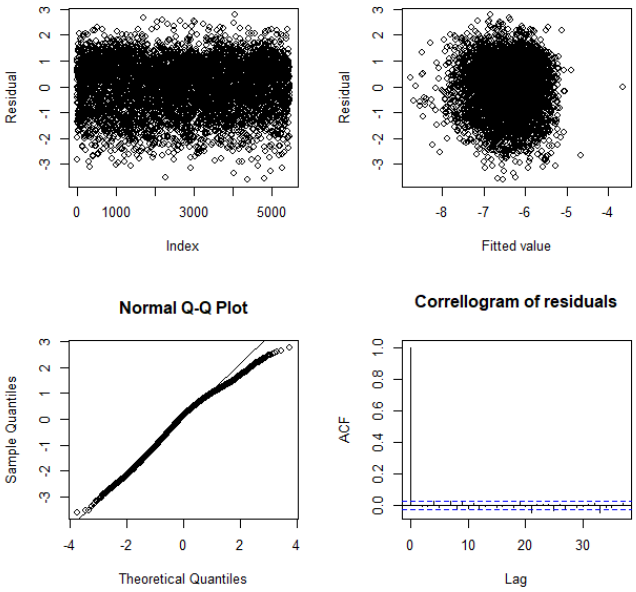


FIGURE B.76: ARMA-GARCH diagnostic plot for machine OP90-6

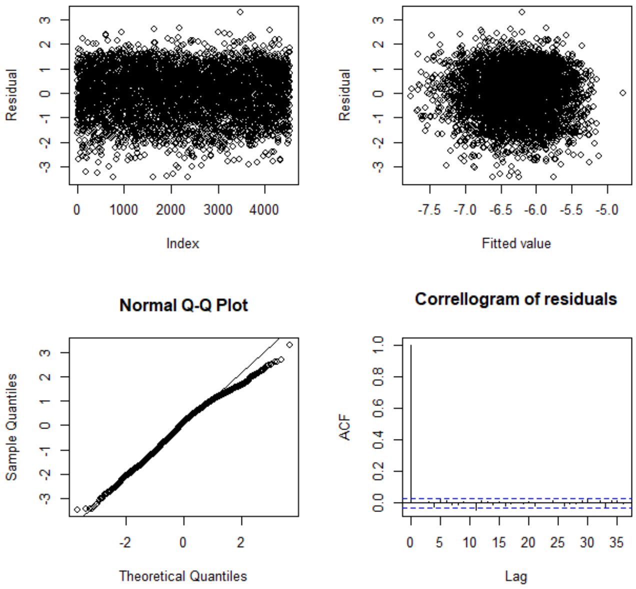


FIGURE B.77: ARMA-GARCH diagnostic plot for machine OP90-7

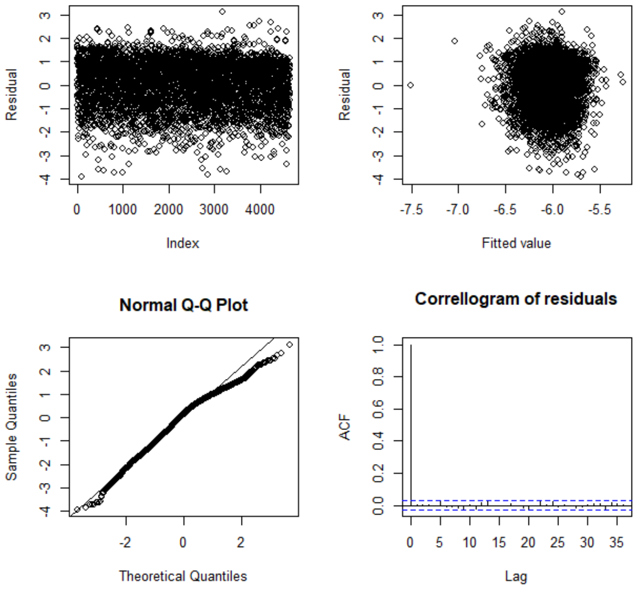


FIGURE B.78: ARMA-GARCH diagnostic plot for machine OP90-8

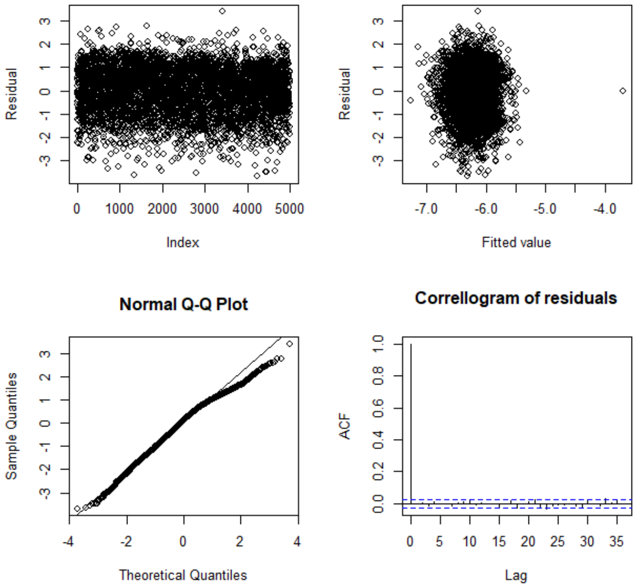


FIGURE B.79: ARMA-GARCH diagnostic plot for machine OP90-9

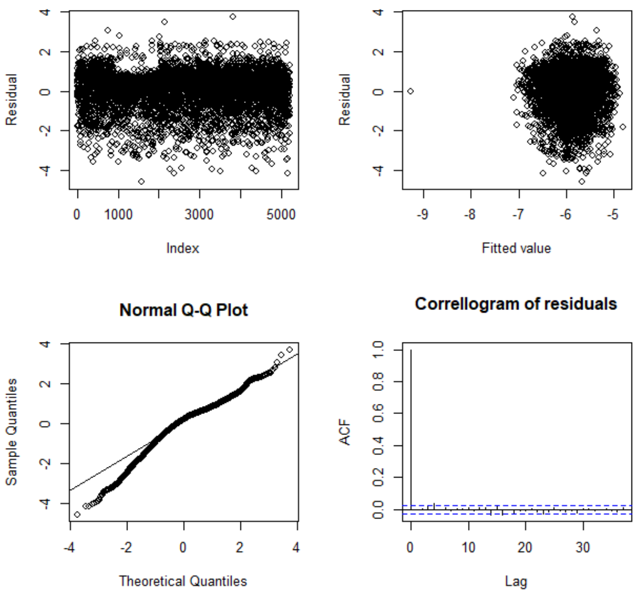


FIGURE B.80: ARMA-GARCH diagnostic plot for machine OP90G-0

# Appendix C

# Diagnostic plots for MMPP fits

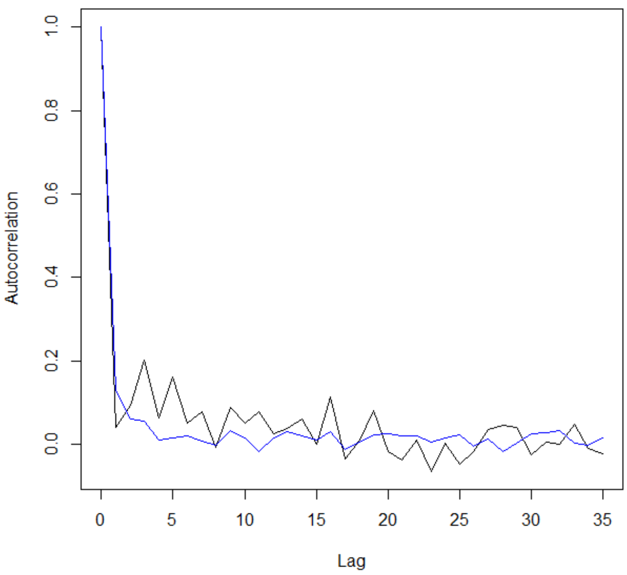


FIGURE C.1: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine DAG0160-0

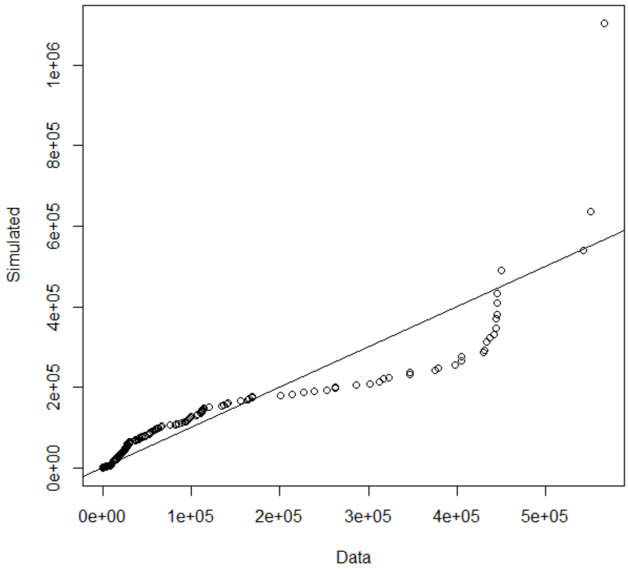


Figure C.2: QQ-plot of simulated and actual uptimes of MMPP fit for machine DAG0160-0

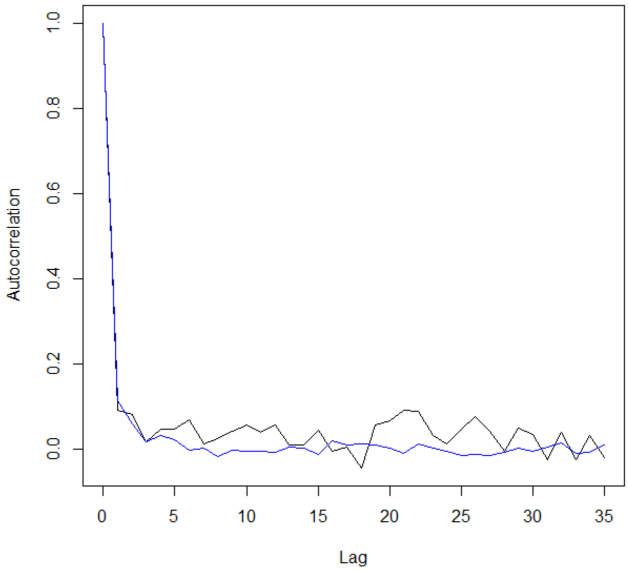


Figure C.3: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP 100-2

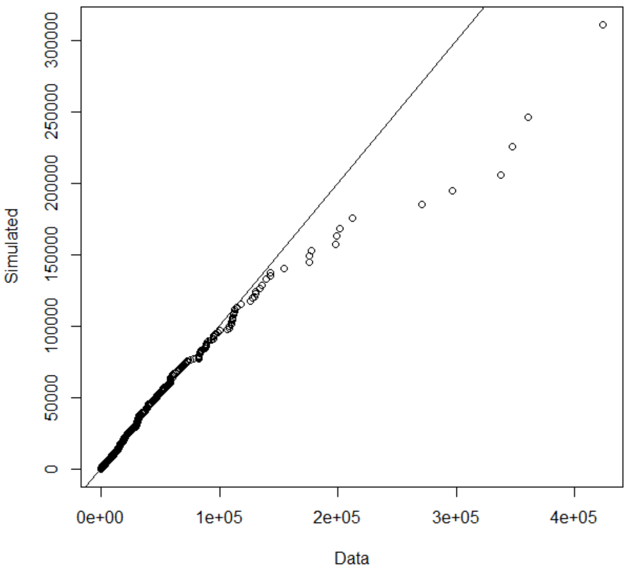


Figure C.4: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP 100-2

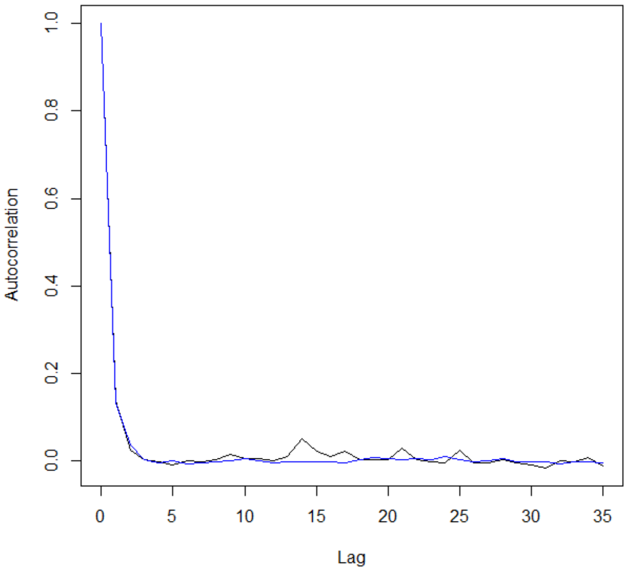


Figure C.5: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP 10A-0

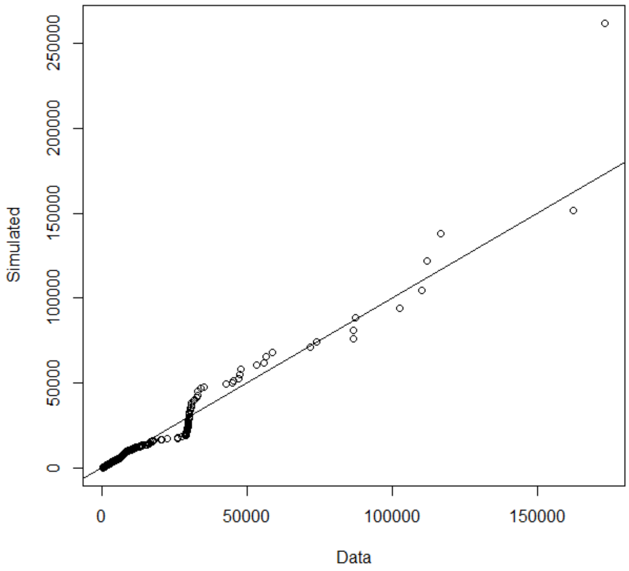


Figure C.6: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP 10A-0

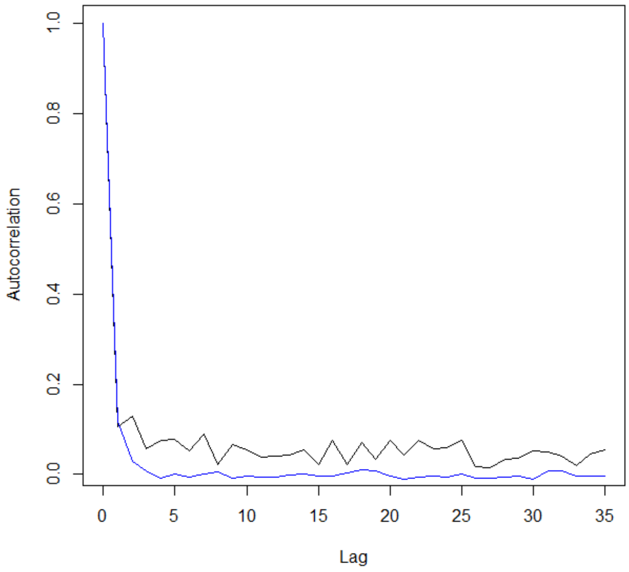


Figure C.7: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP 110-0

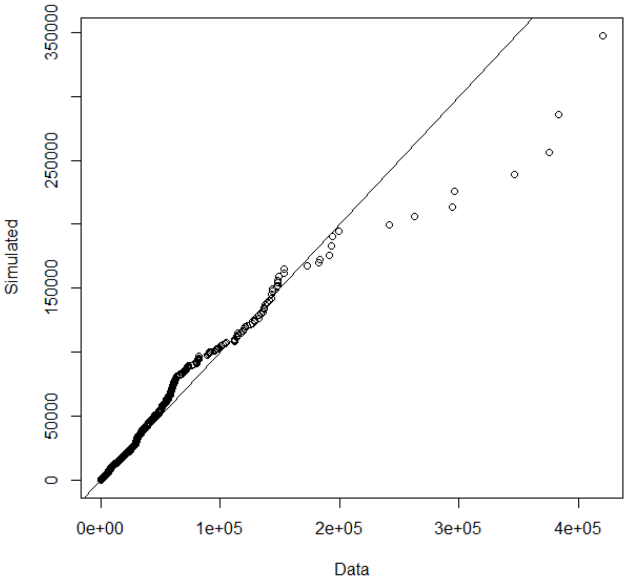


FIGURE C.8: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP 110-0

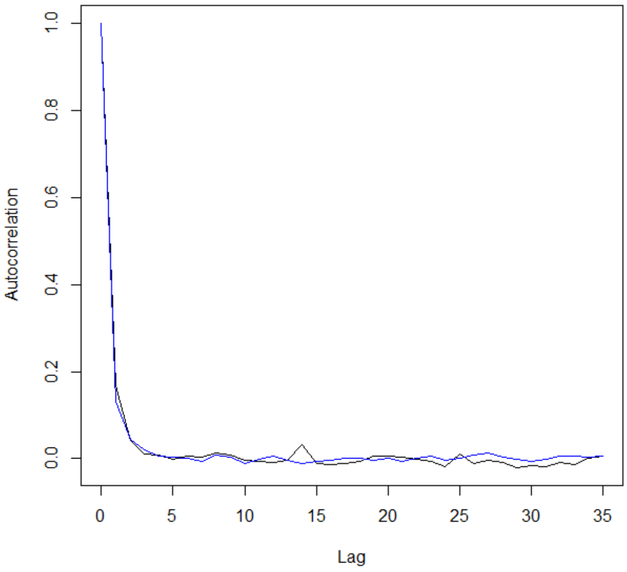


FIGURE C.9: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP 125-0

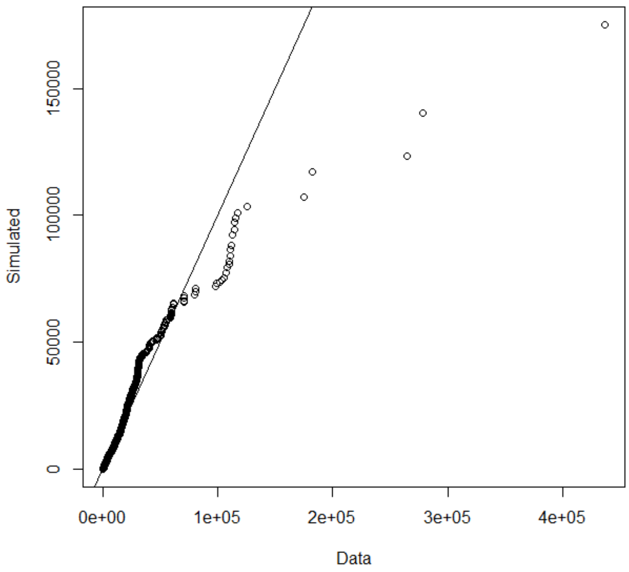


FIGURE C.10: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP 125-0

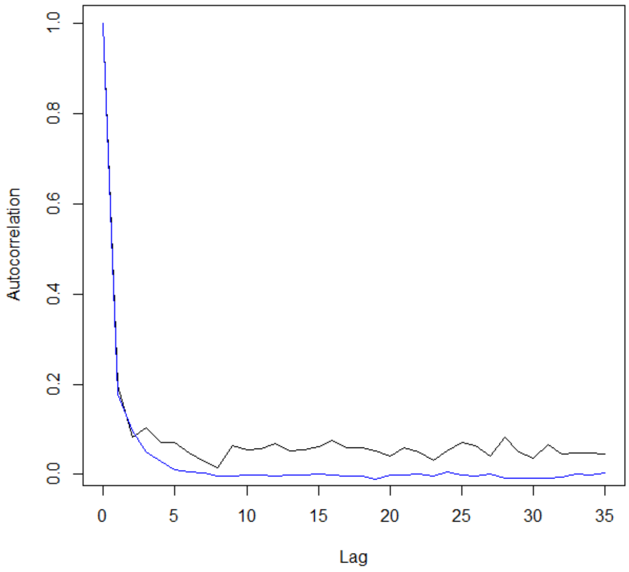


FIGURE C.11: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP 135-0

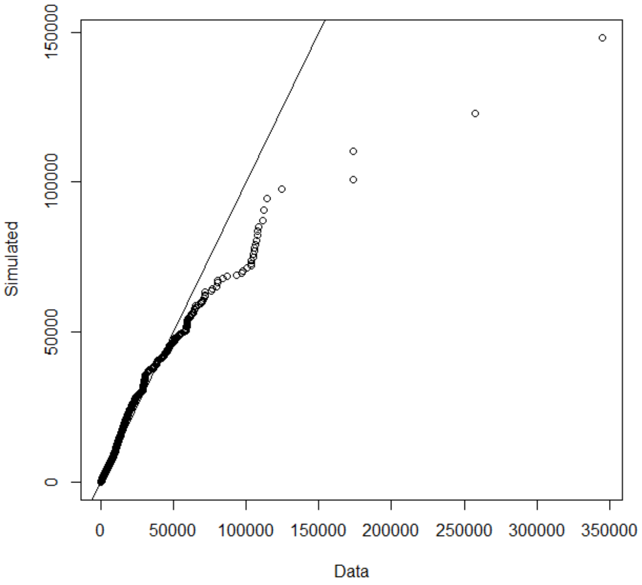


FIGURE C.12: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP 135-0

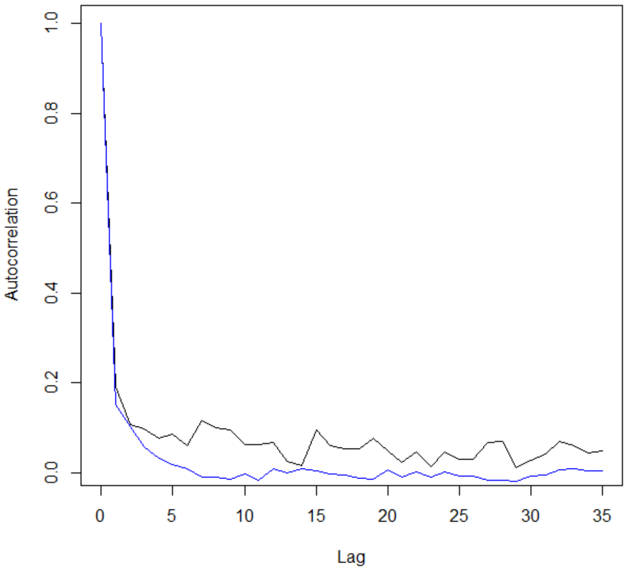


FIGURE C.13: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP 140-0

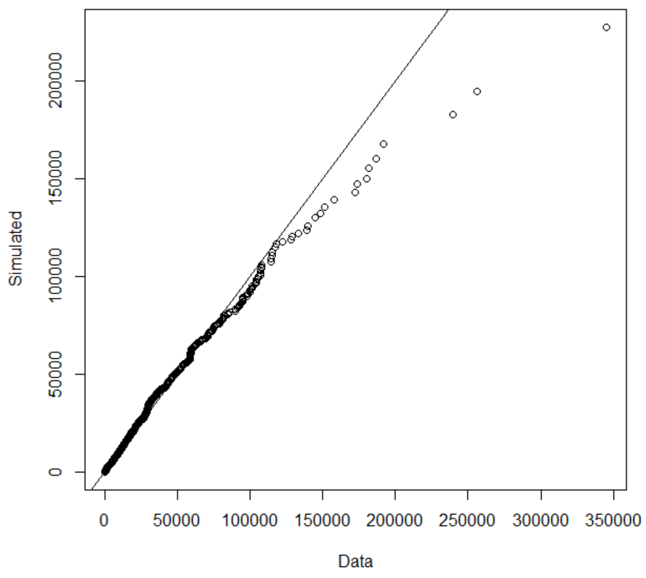


Figure C.14: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP 140-0

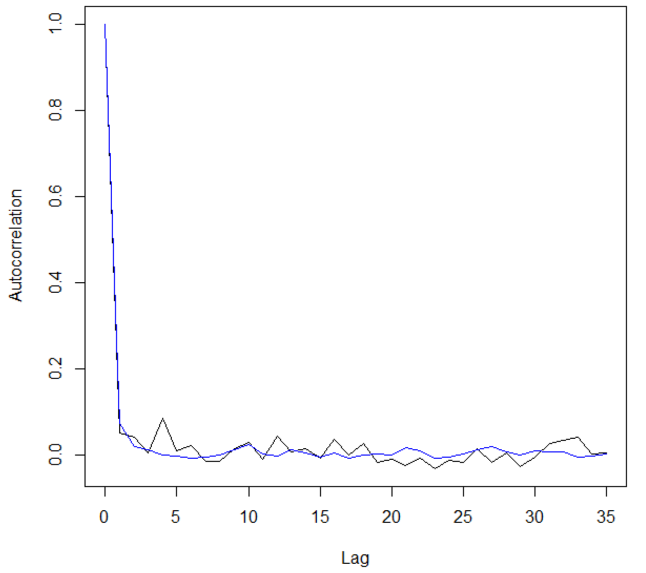


Figure C.15: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP 150-0

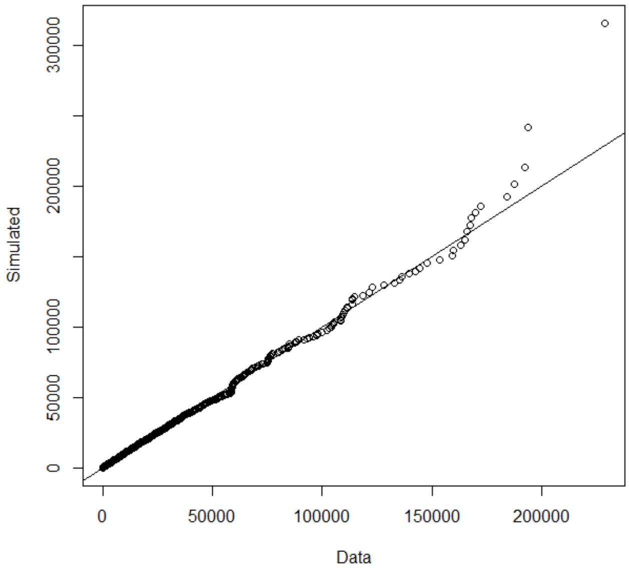


Figure C.16: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP 150-0

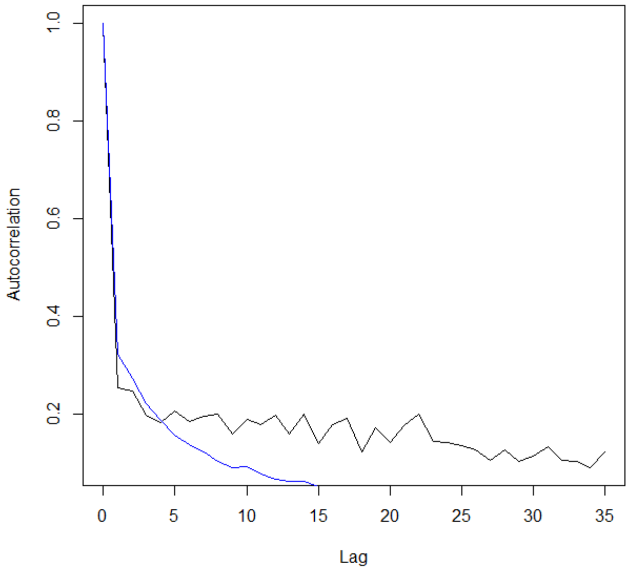


Figure C.17: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP 160-0

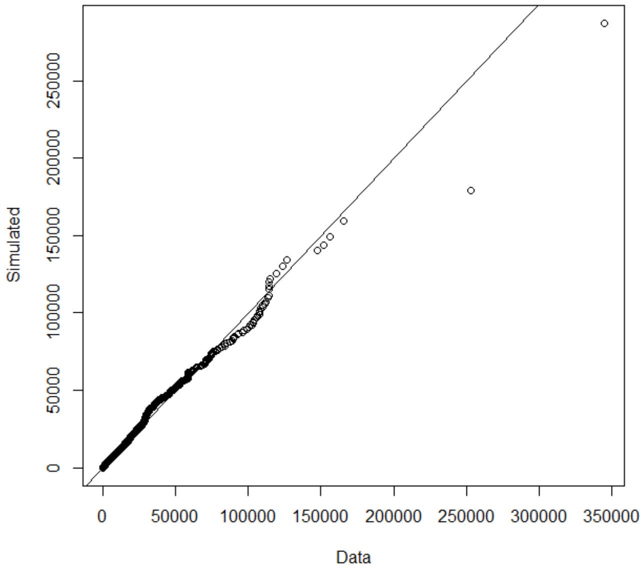


Figure C.18: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP 160-0

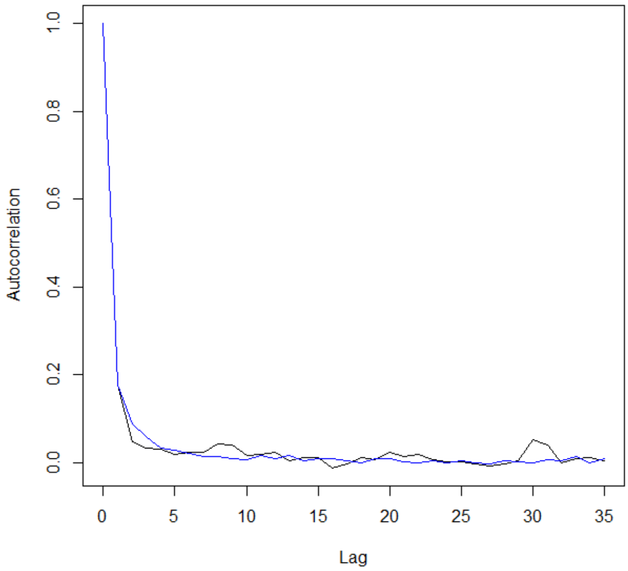


Figure C.19: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP 170_SCR-0

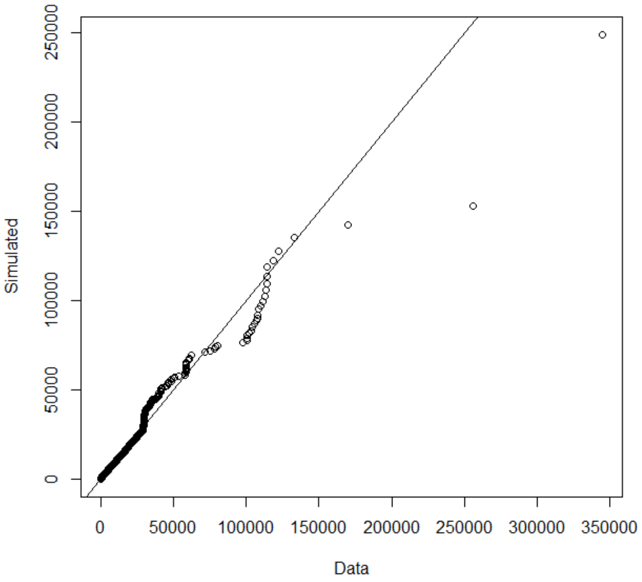


Figure C.20: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP 170_SCR-0

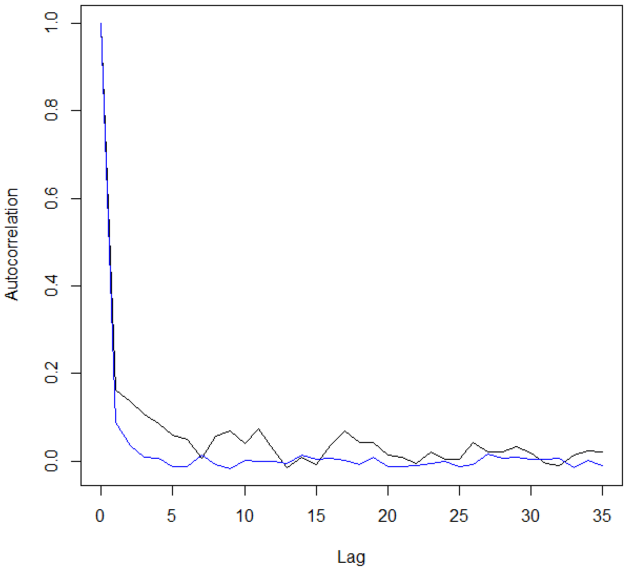


Figure C.21: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP 30A-0

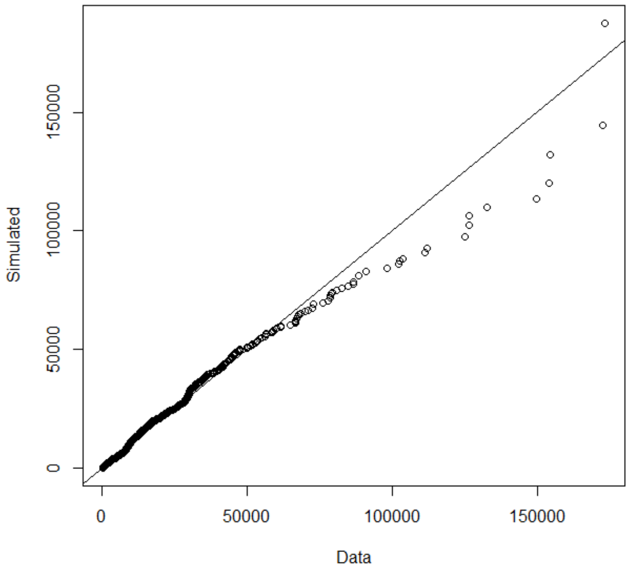


Figure C.22: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP 30A-0

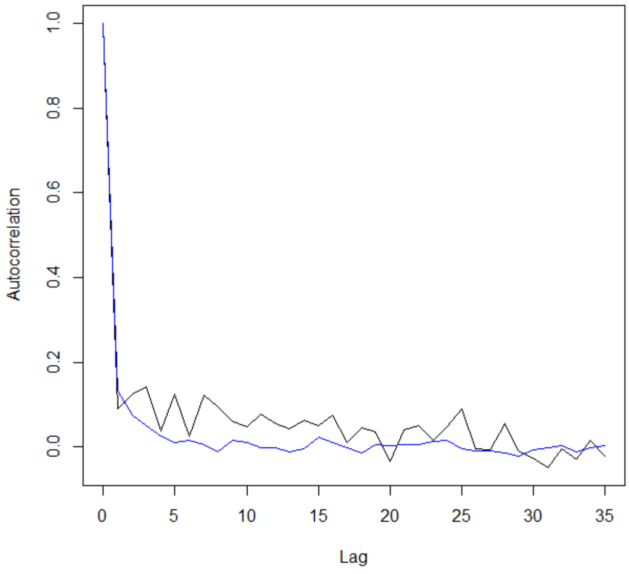


Figure C.23: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP 40-0

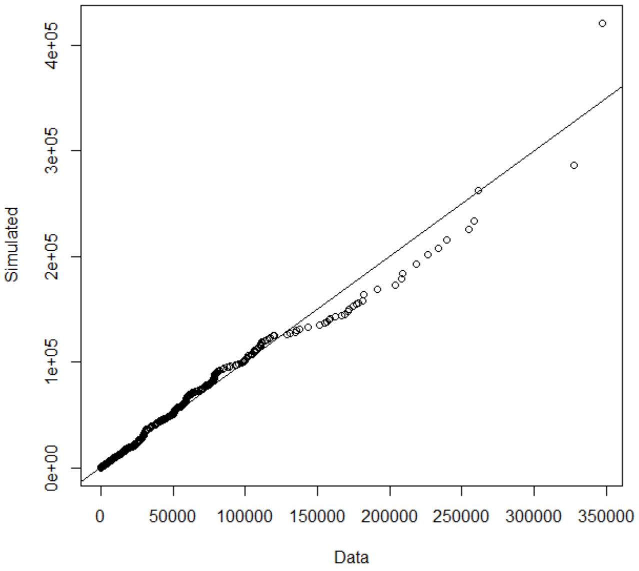


FIGURE C.24: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP 40-0

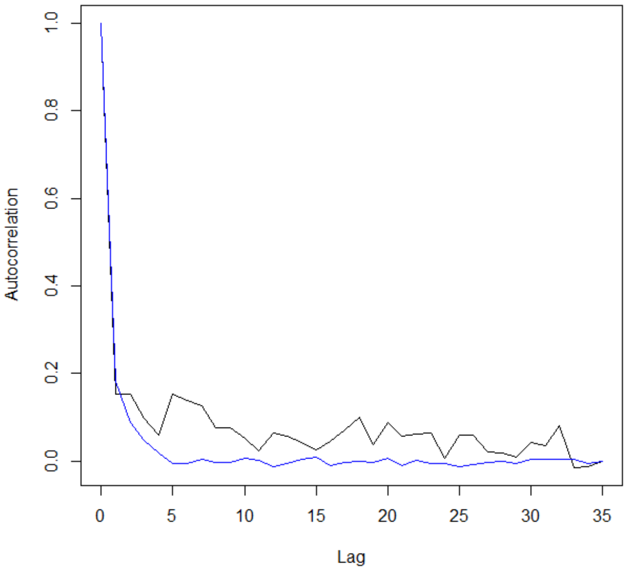


FIGURE C.25: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP 45A-0

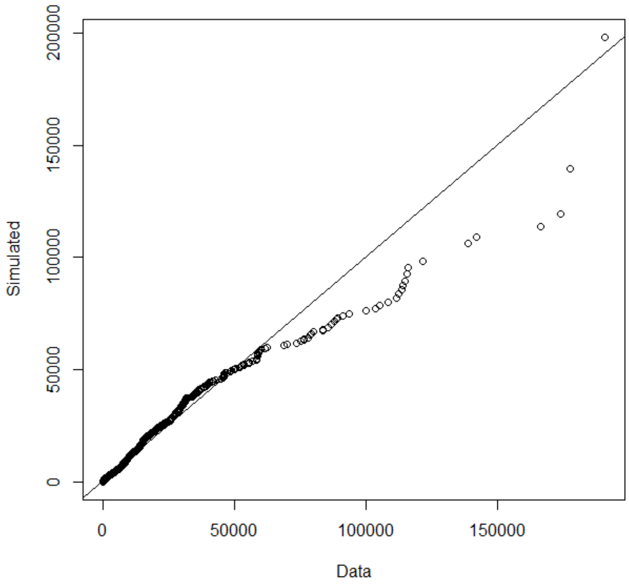


FIGURE C.26: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP 45A-0

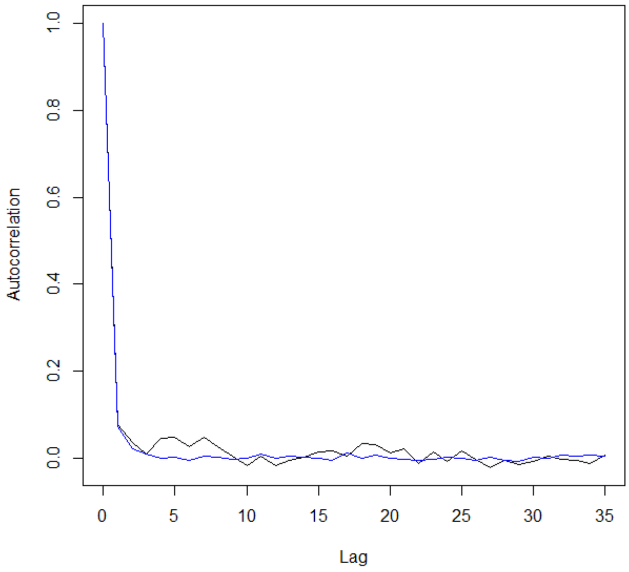


FIGURE C.27: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP 70-6

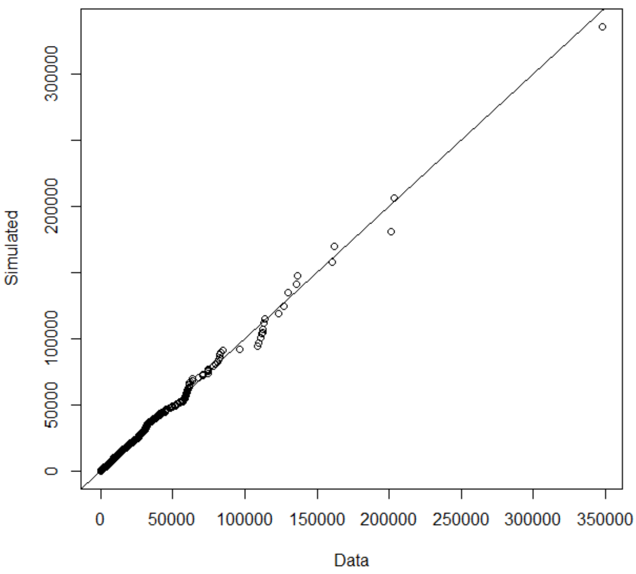


Figure C.28: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP 70-6

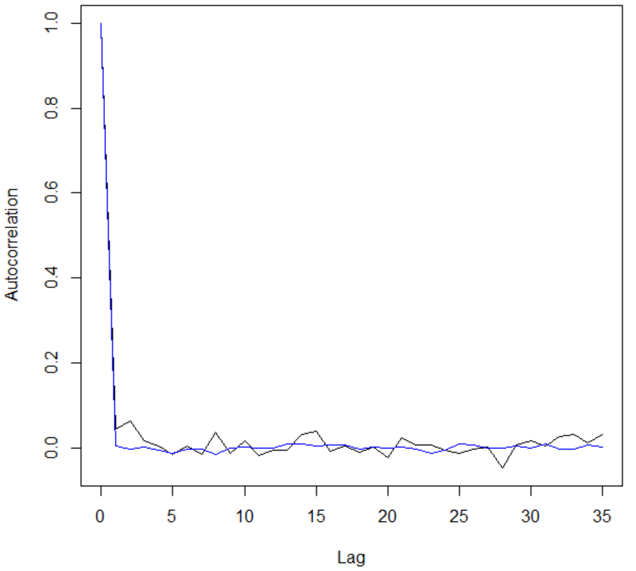


Figure C.29: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP10-1

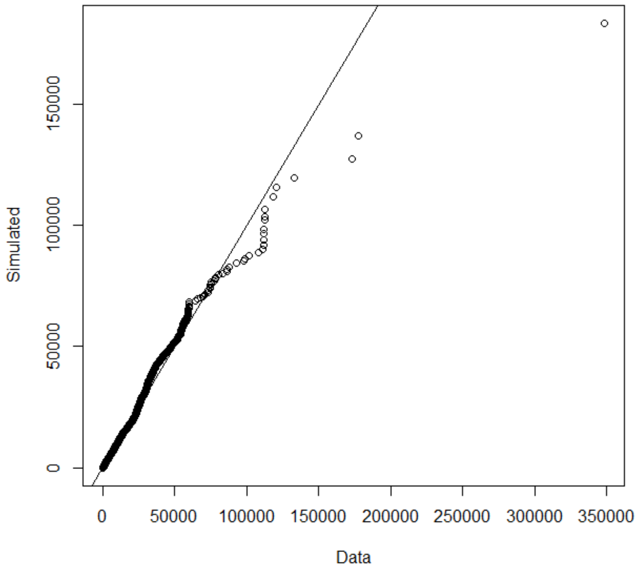


FIGURE C.30: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP10-1

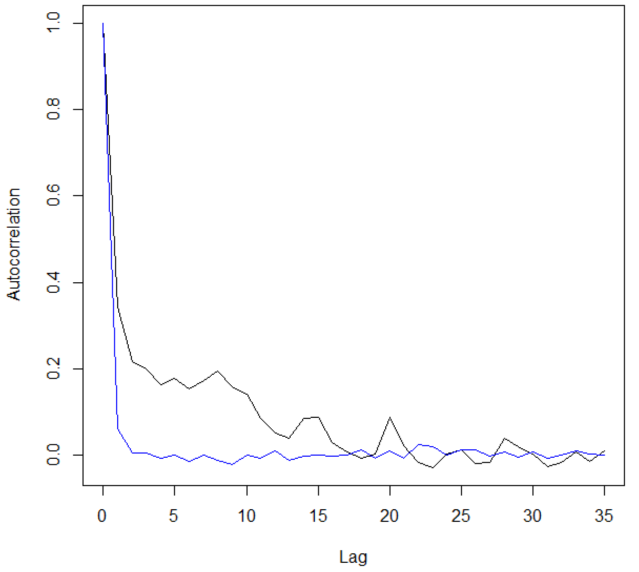


FIGURE C.31: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP10-2

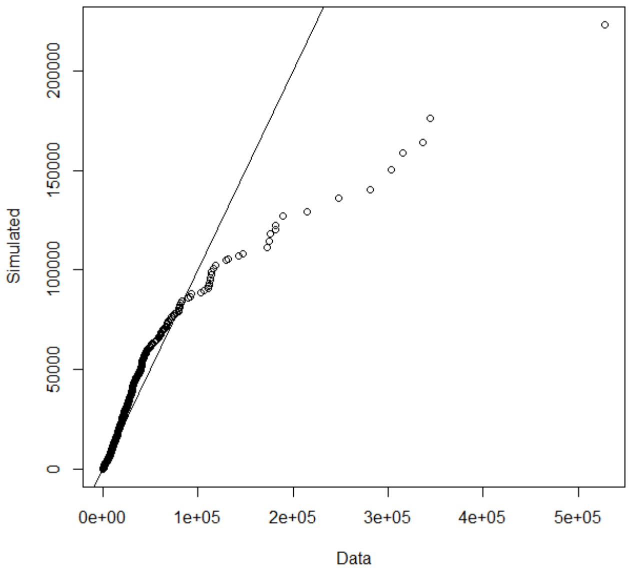


FIGURE C.32: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP10-2

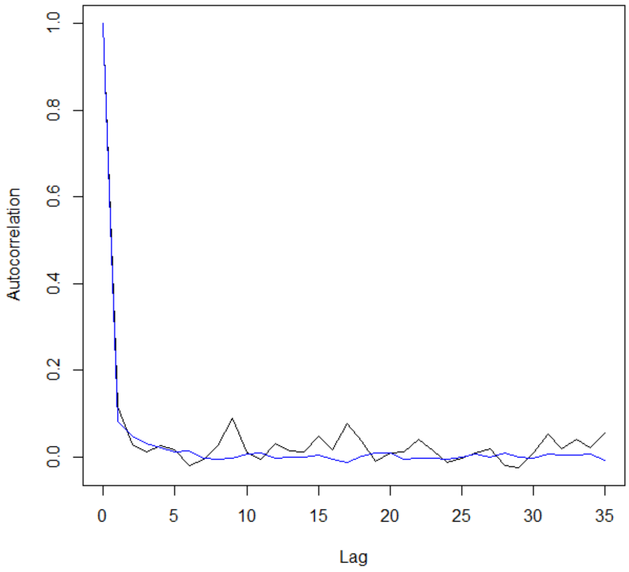


FIGURE C.33: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP10-3

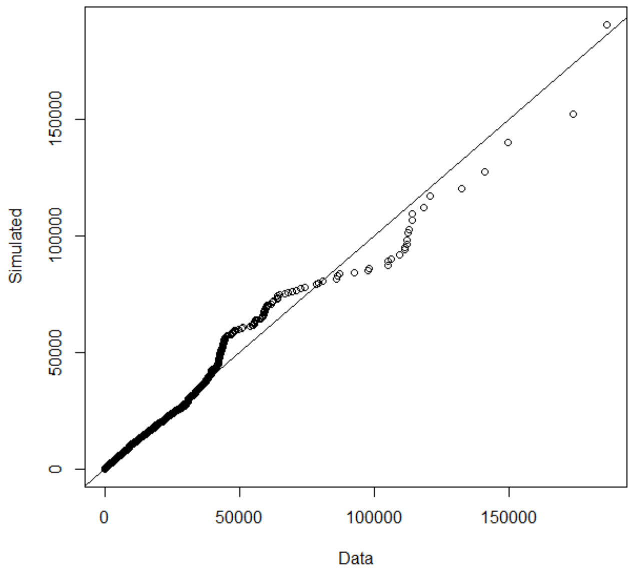


FIGURE C.34: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP10-3

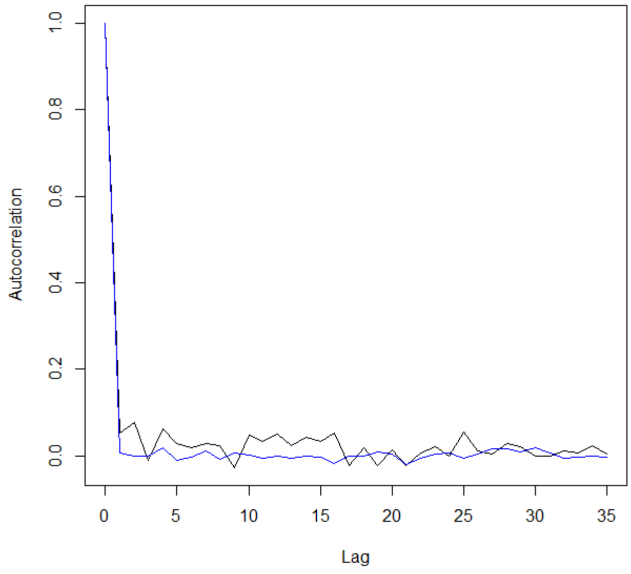


FIGURE C.35: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP10-4

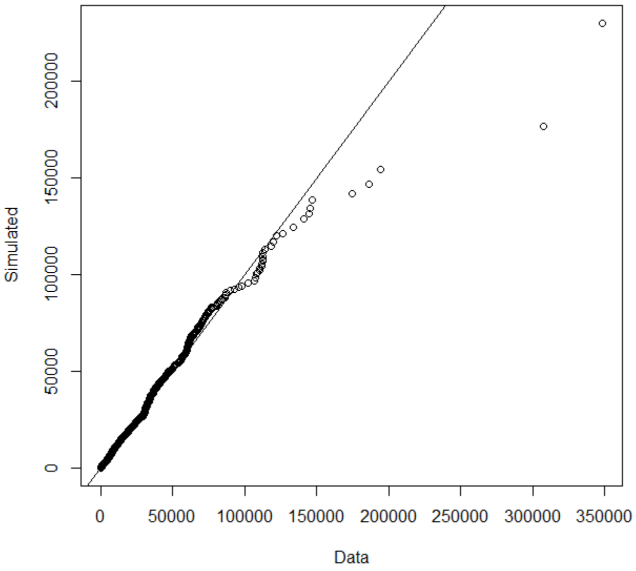


Figure C.36: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP10-4

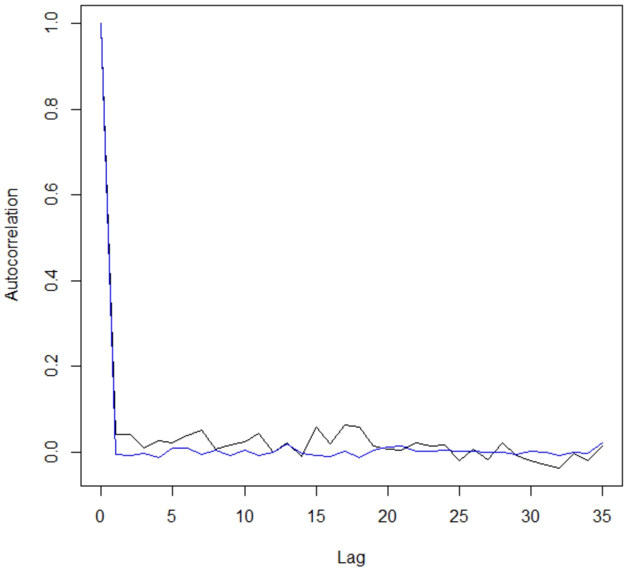


Figure C.37: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP10-5

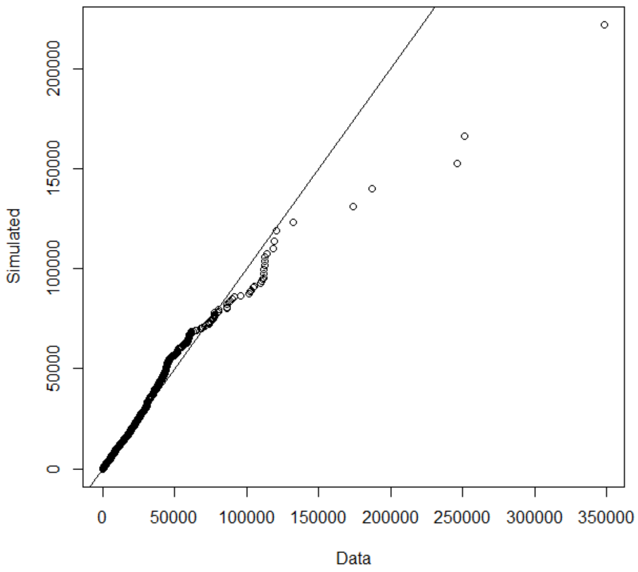


FIGURE C.38: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP10-5

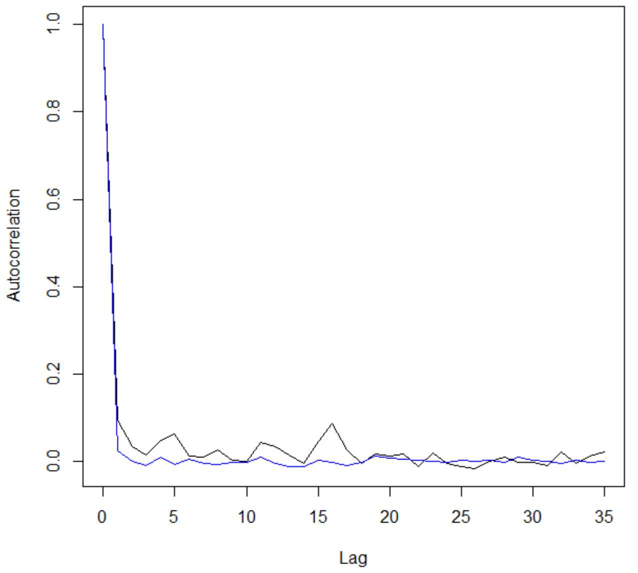


FIGURE C.39: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP10-6

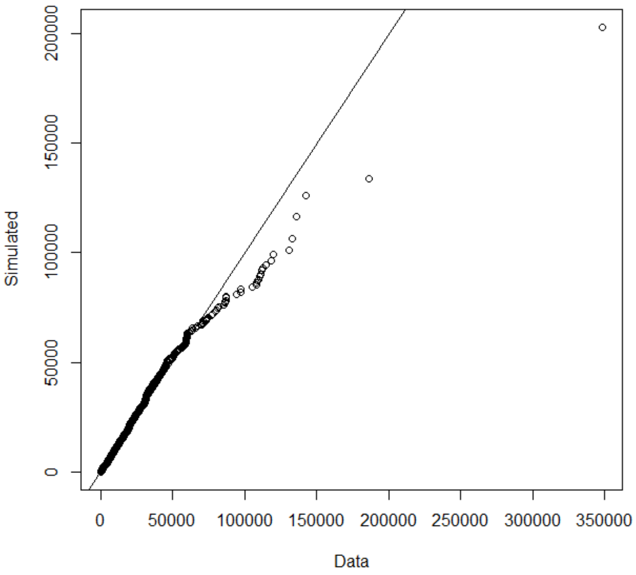


FIGURE C.40: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP10-6

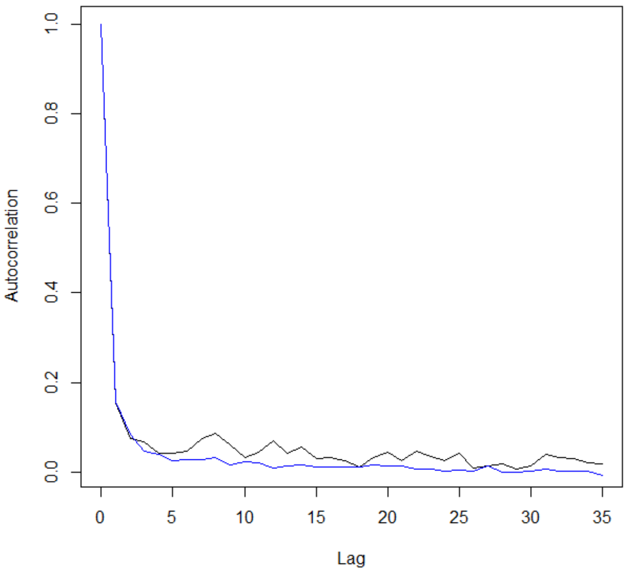


FIGURE C.41: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP100G-0

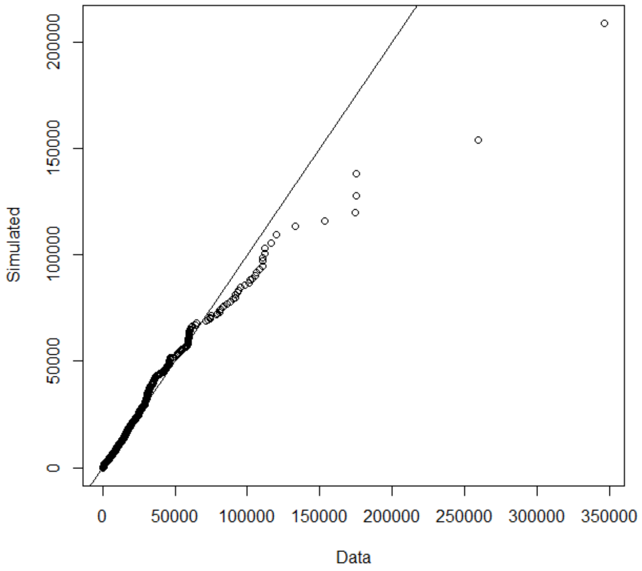


FIGURE C.42: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP100G-0

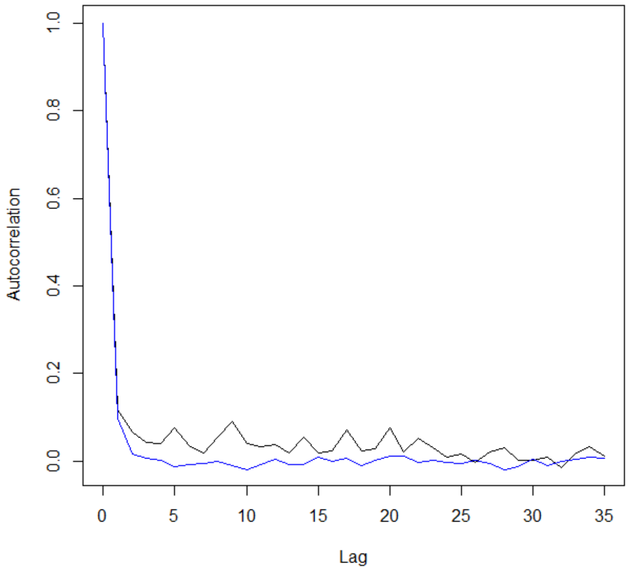


FIGURE C.43: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP10G-0

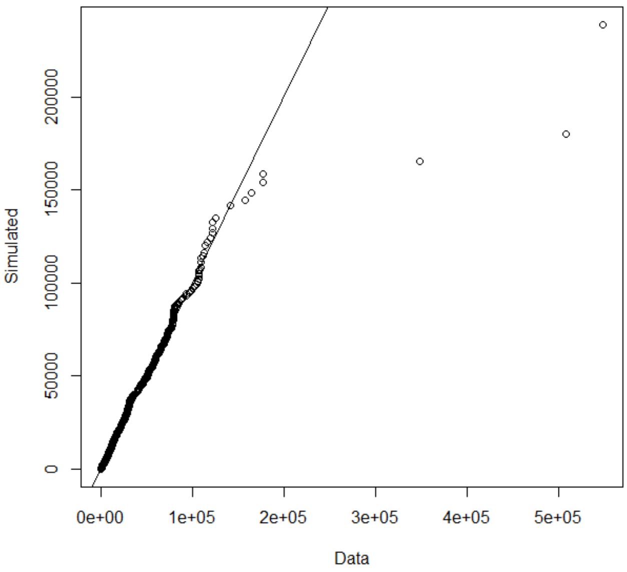


Figure C.44: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP10G-0

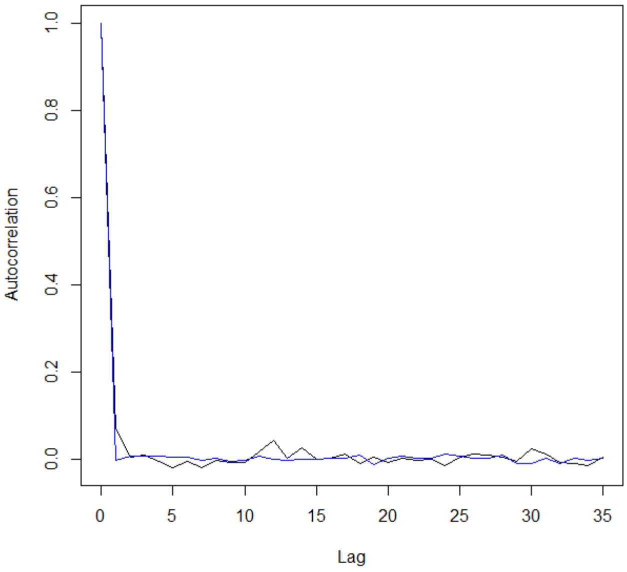


Figure C.45: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP120-1

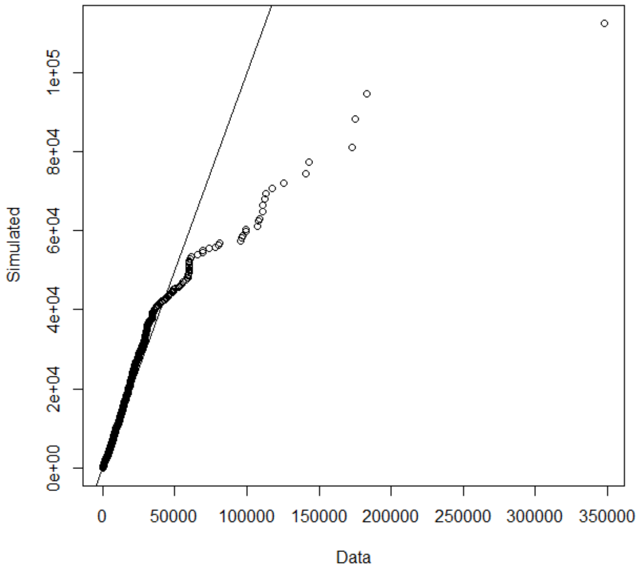


FIGURE C.46: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP120-1

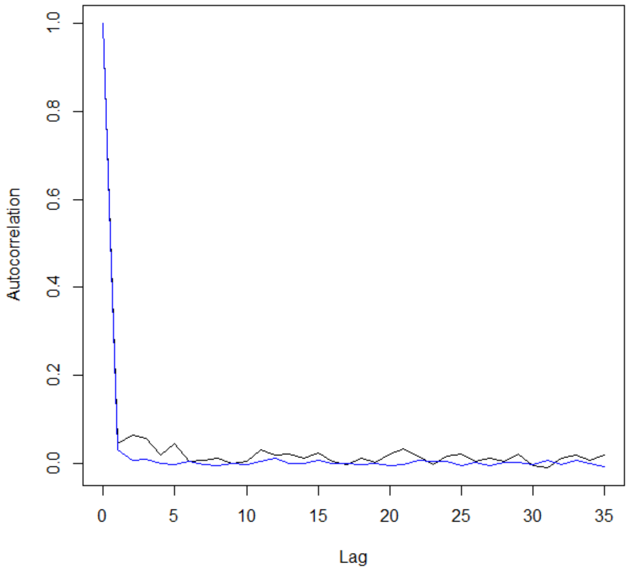


FIGURE C.47: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP120-2

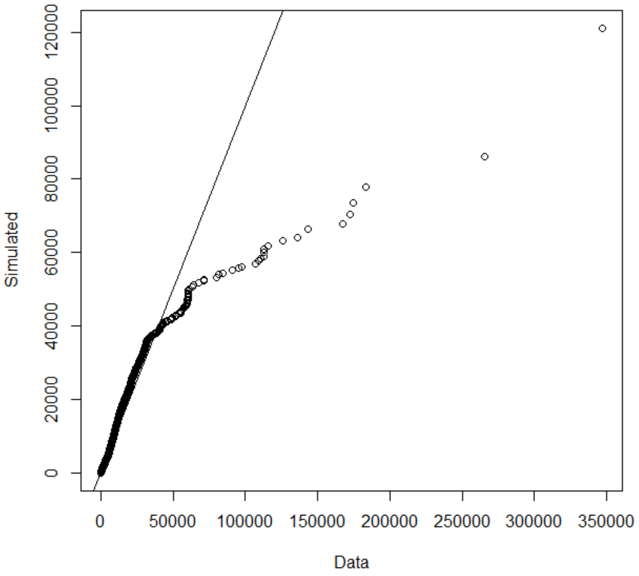


Figure C.48: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP120-2

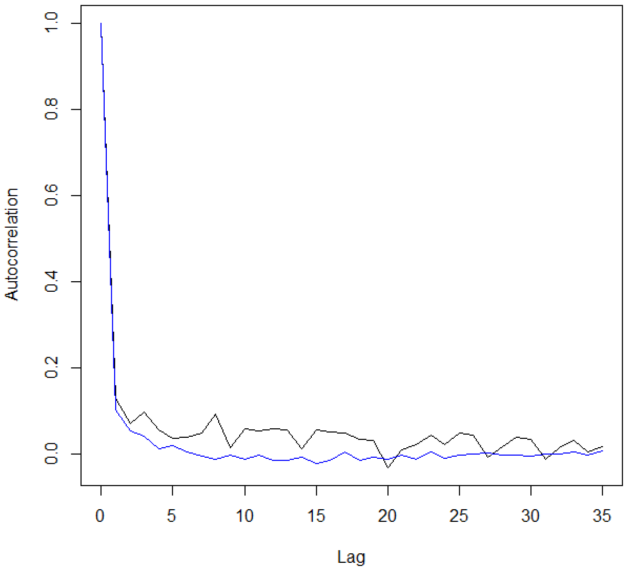


Figure C.49: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP130-1

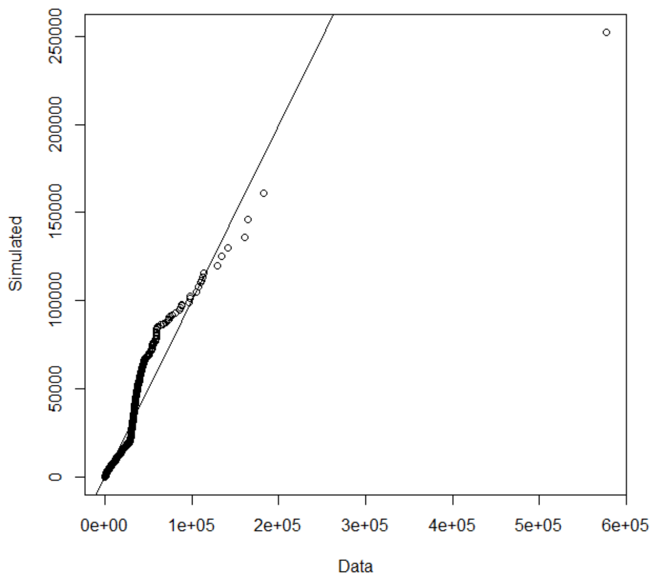


FIGURE C.50: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP130-1

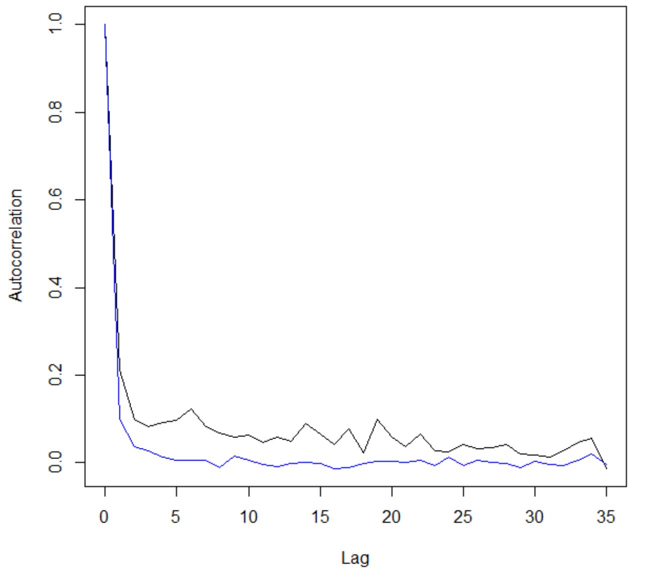


FIGURE C.51: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP130-2

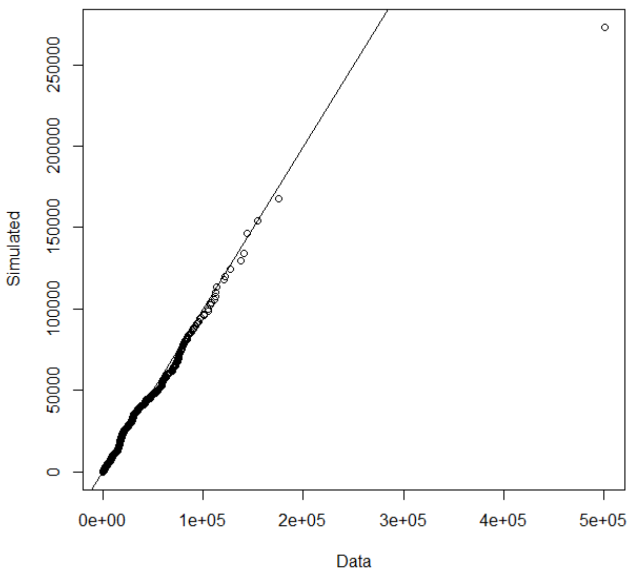


FIGURE C.52: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP130-2

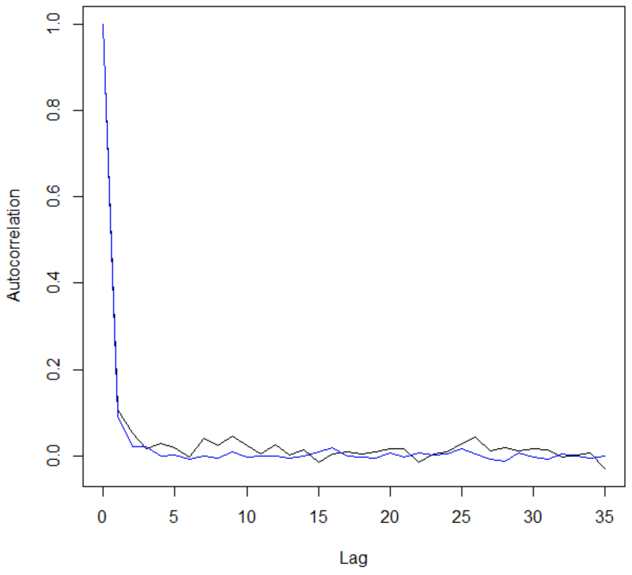


FIGURE C.53: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP130G-0

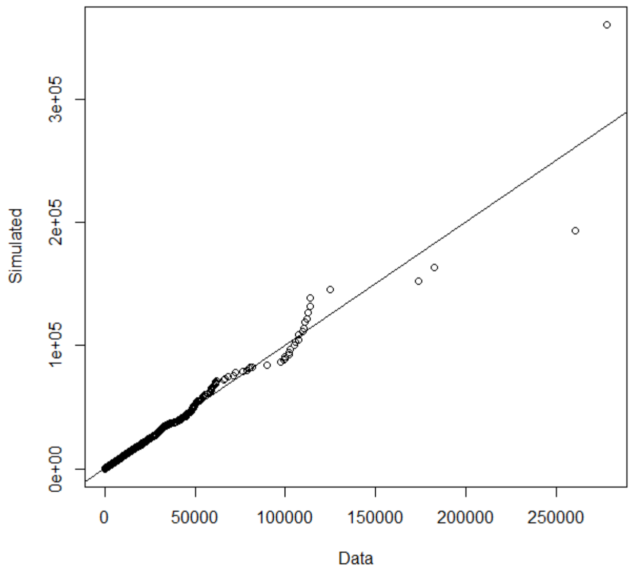


Figure C.54: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP130G-0

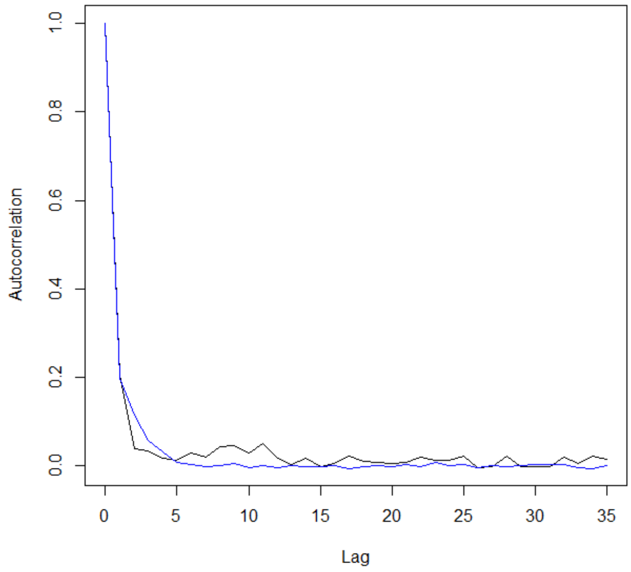


Figure C.55: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP170_LEAK-0

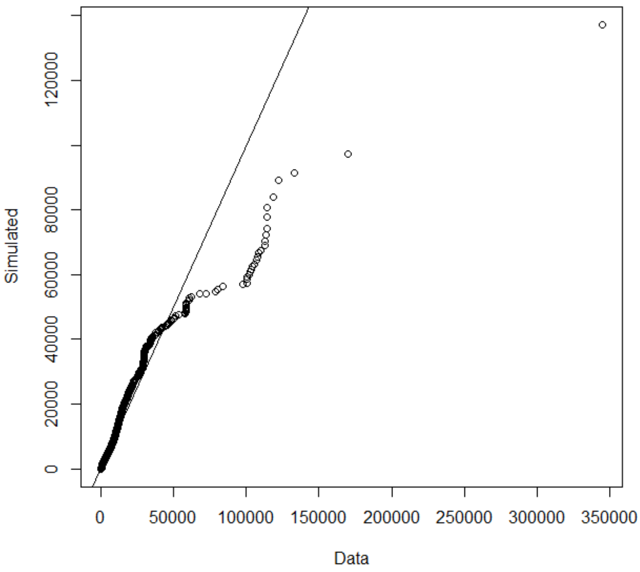


Figure C.56: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP170_LEAK-0

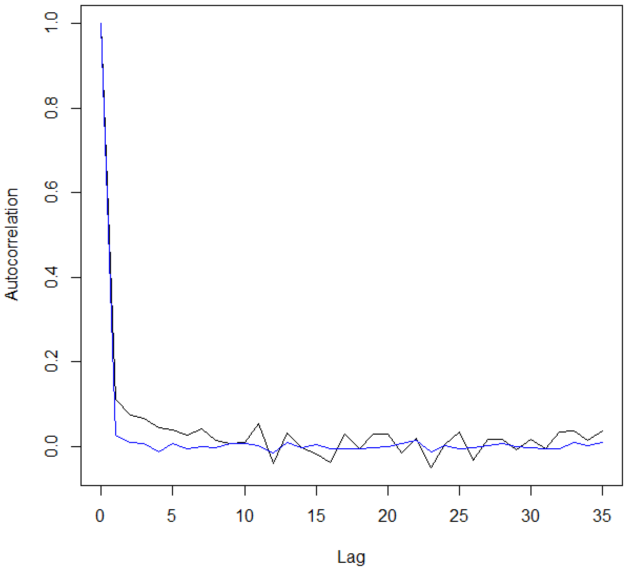


Figure C.57: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP20-1

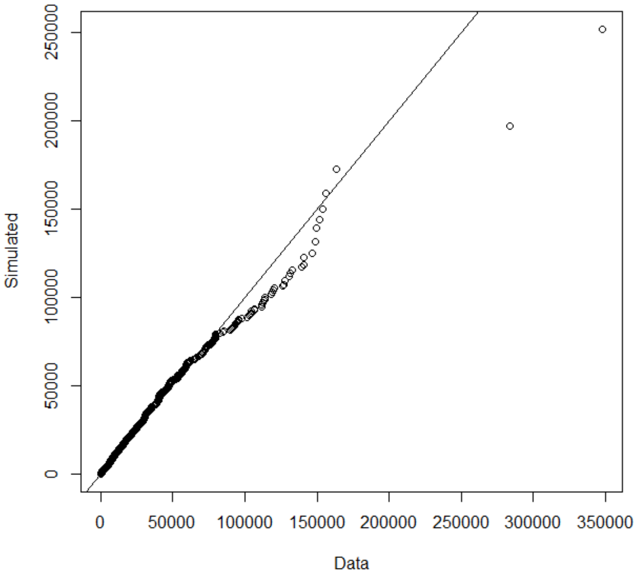


FIGURE C.58: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP20-1

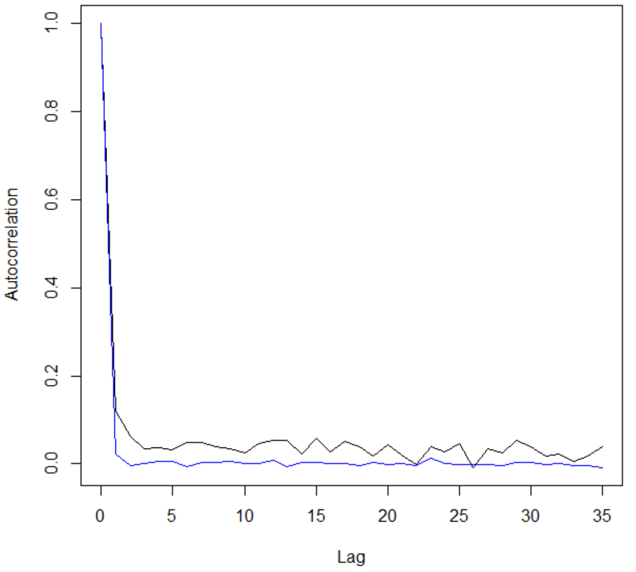


FIGURE C.59: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP20-10

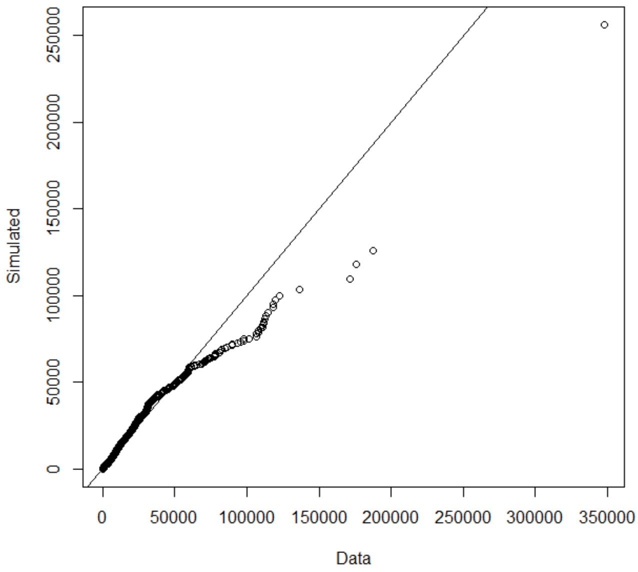


Figure C.60: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP20-10

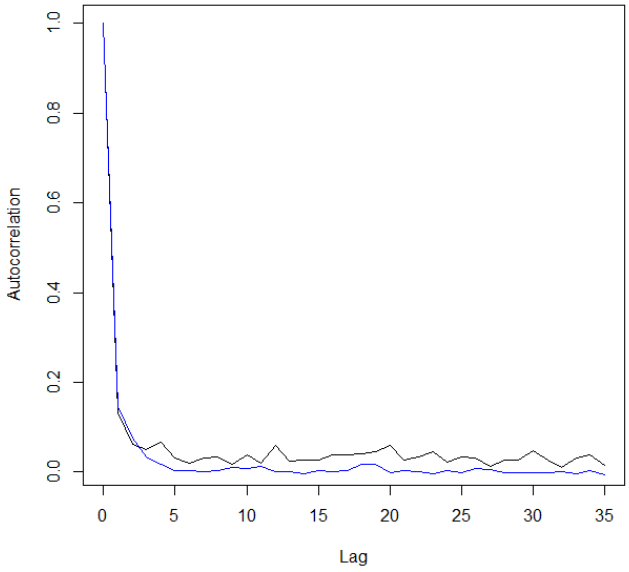


Figure C.61: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP20-11

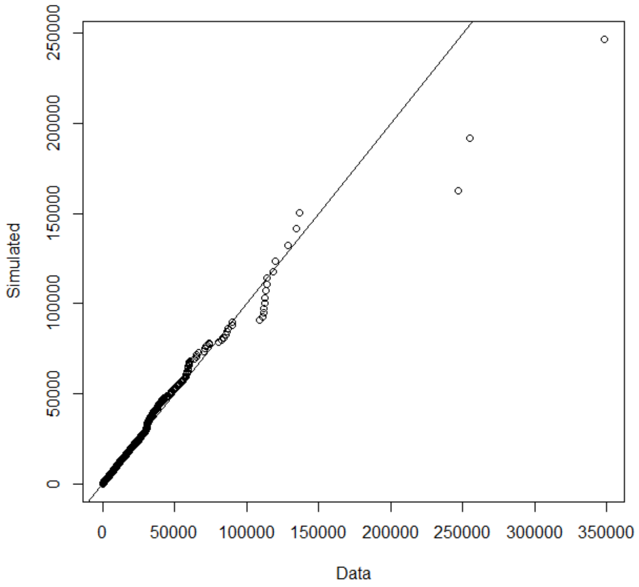


Figure C.62: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP20-11

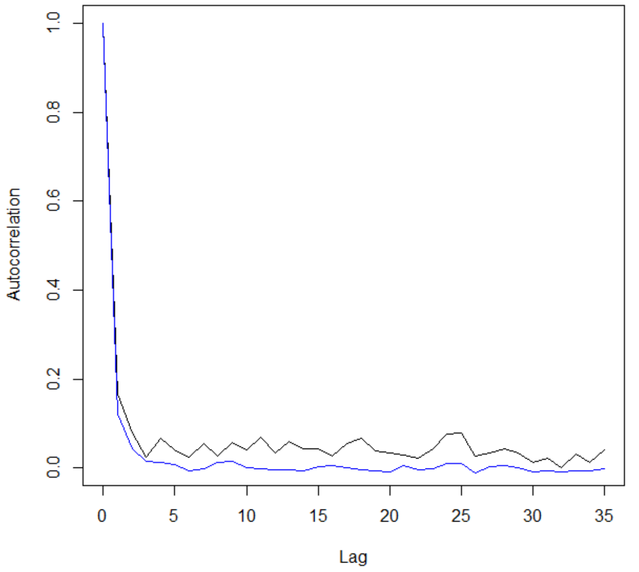


Figure C.63: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP20-12

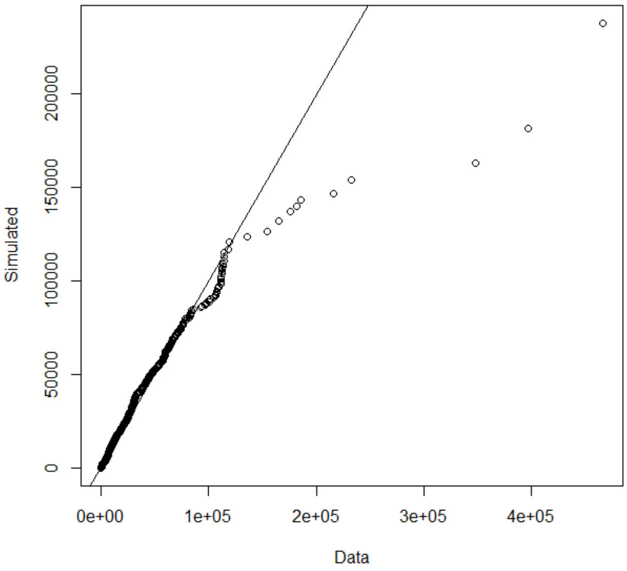


FIGURE C.64: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP20-12

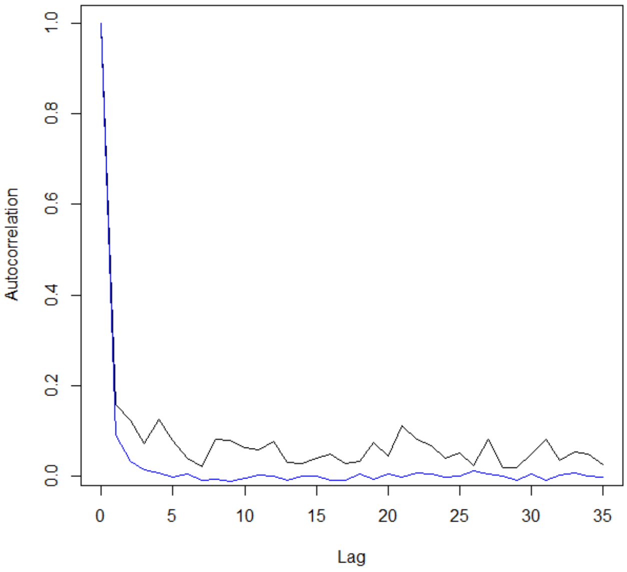


FIGURE C.65: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP20-2

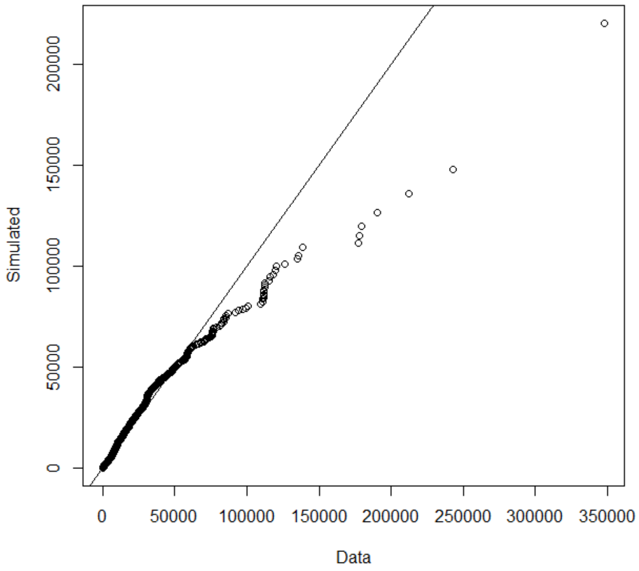


FIGURE C.66: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP20-2

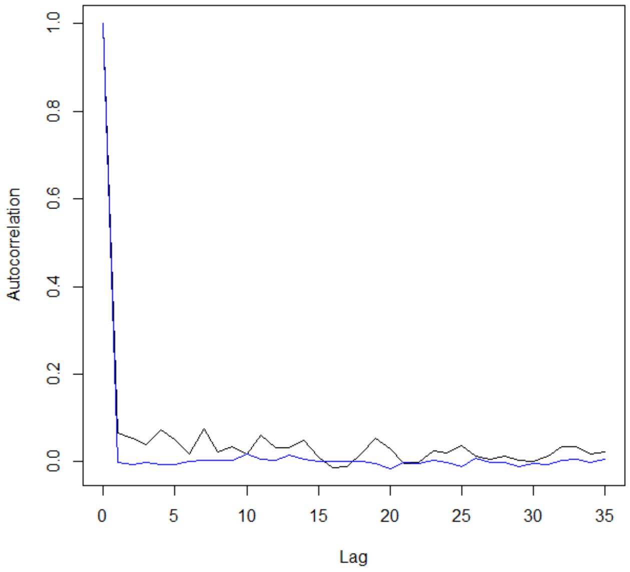


FIGURE C.67: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP20-3

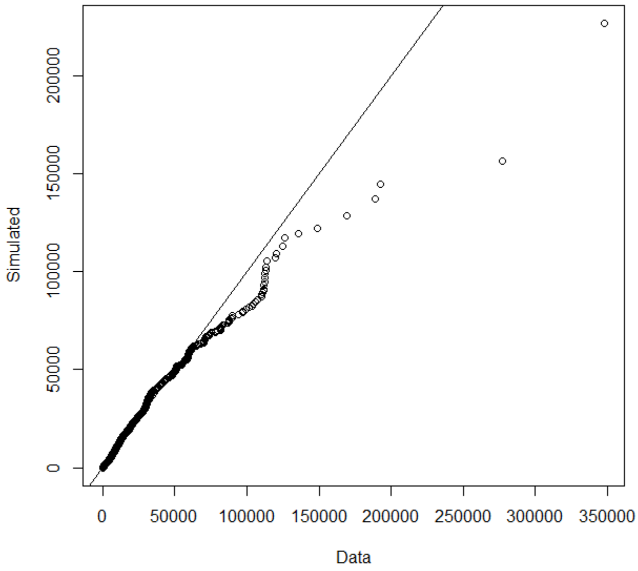


FIGURE C.68: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP20-3

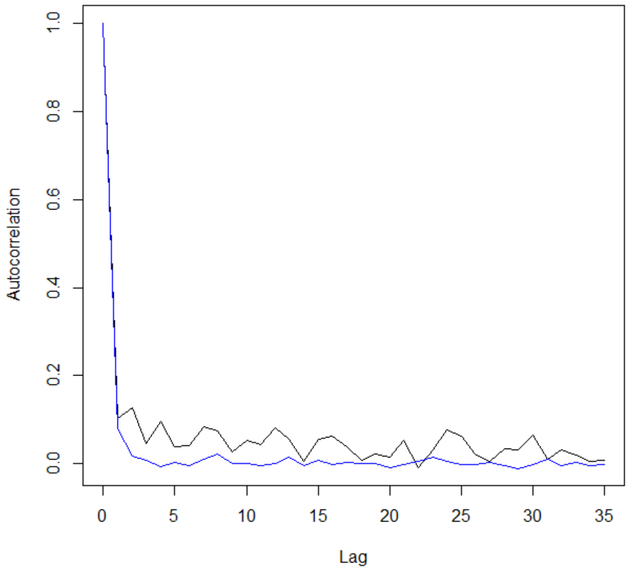


FIGURE C.69: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP20-4

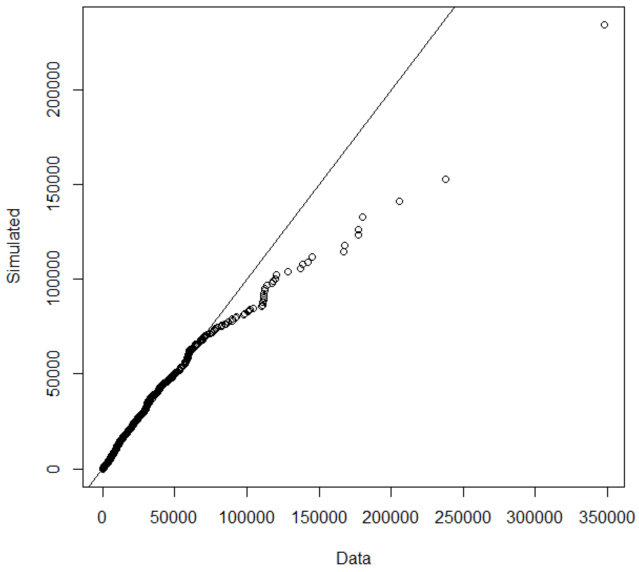


FIGURE C.70: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP20-4

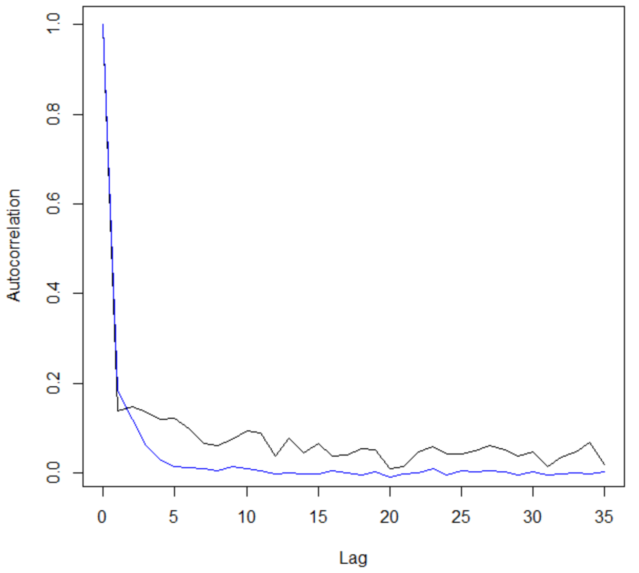


FIGURE C.71: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP20-5

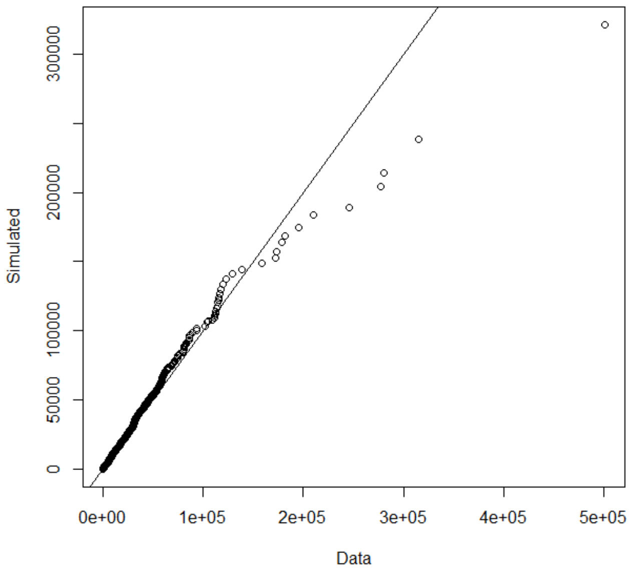


Figure C.72: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP20-5

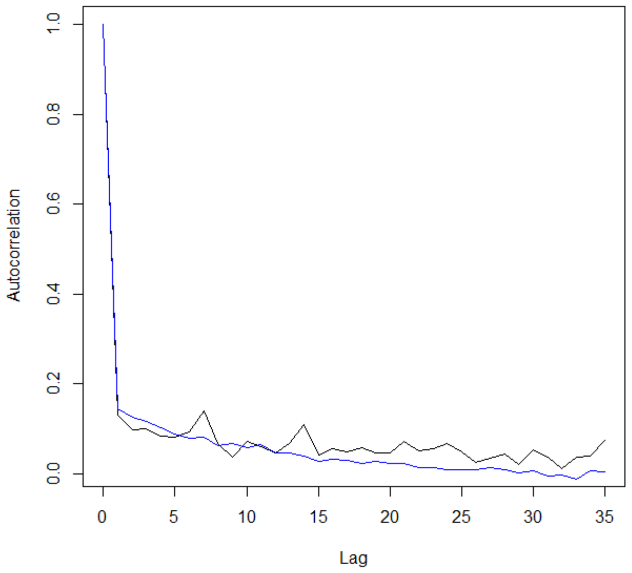


Figure C.73: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP20-6

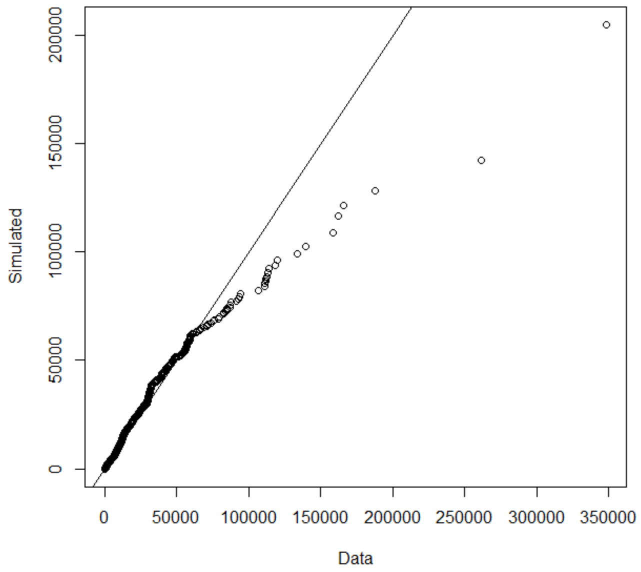


FIGURE C.74: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP20-6

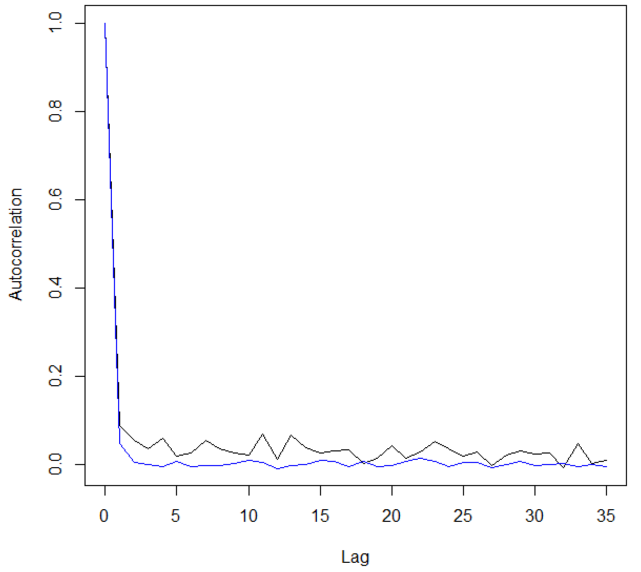


FIGURE C.75: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP20-7

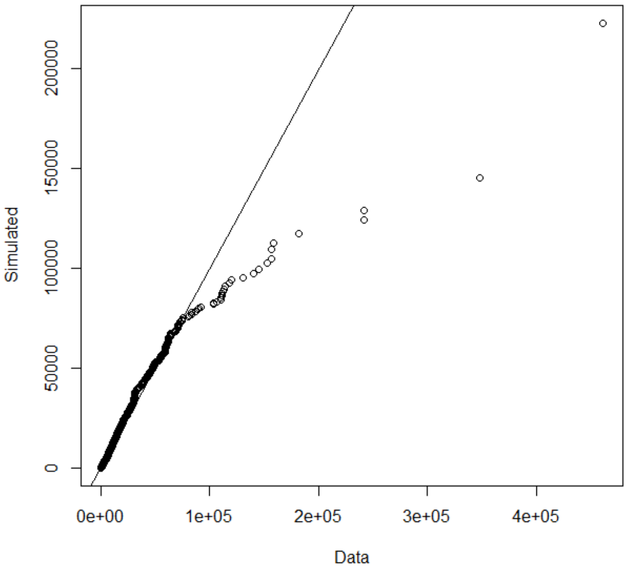


Figure C.76: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP20-7

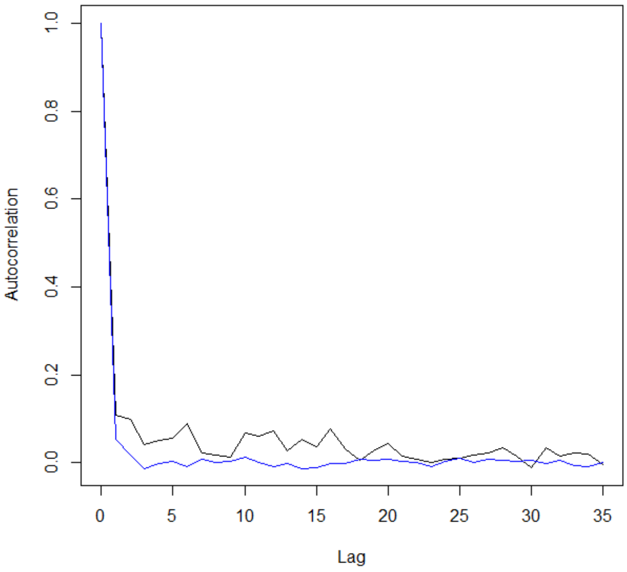


Figure C.77: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP20-8

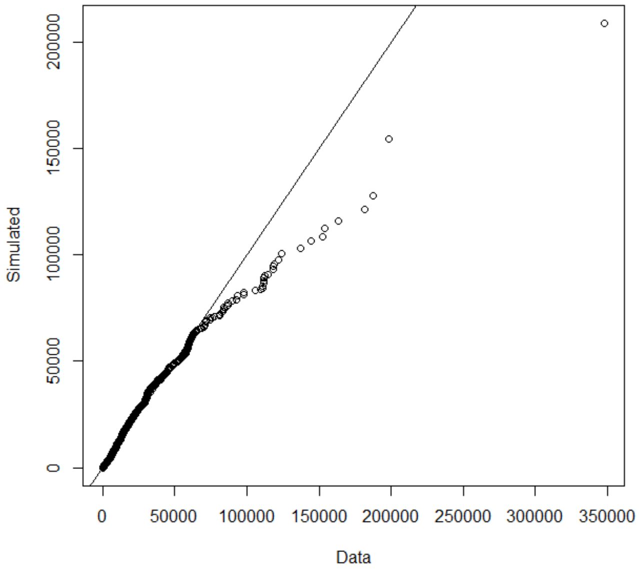


FIGURE C.78: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP20-8

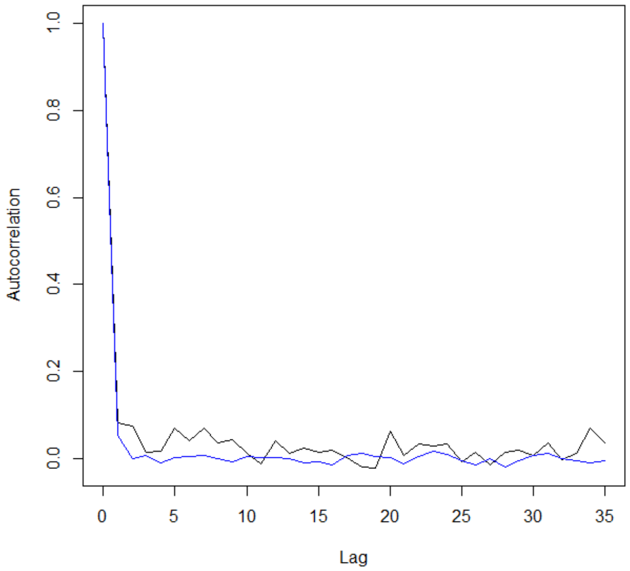


FIGURE C.79: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP20-9

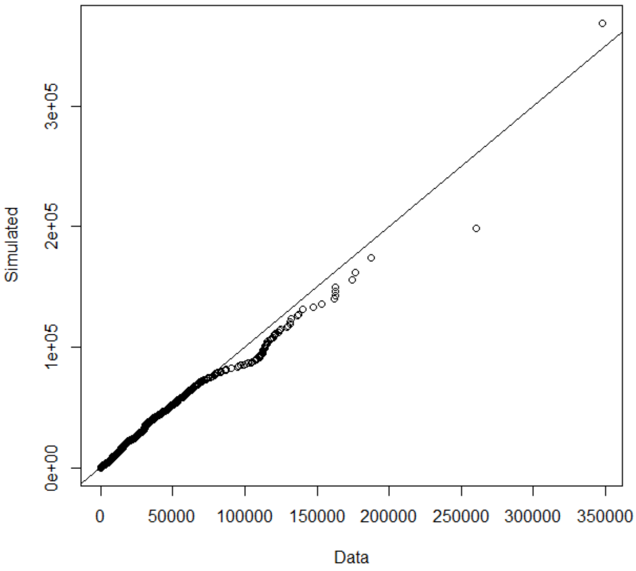


Figure C.80: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP20-9

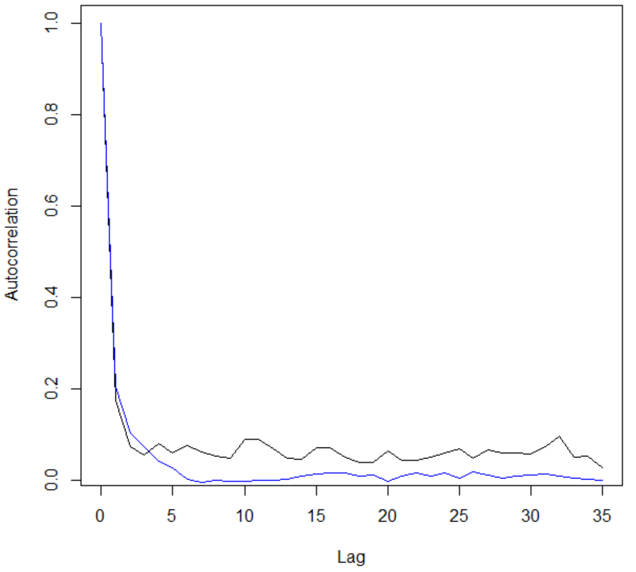


Figure C.81: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP20G-0

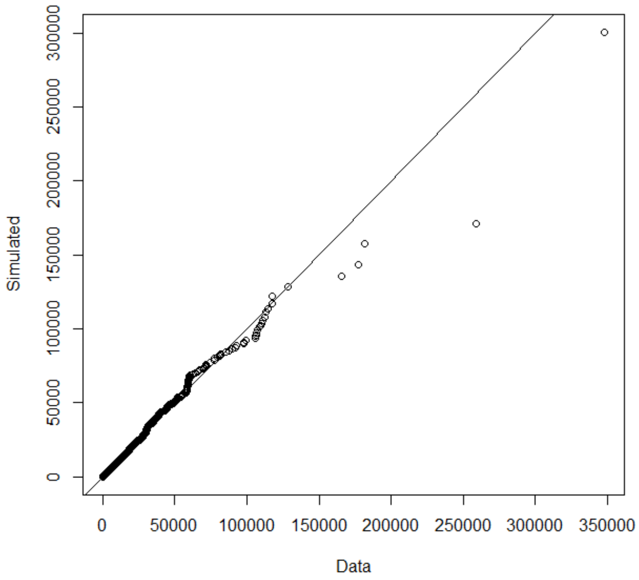


FIGURE C.82: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP20G-0

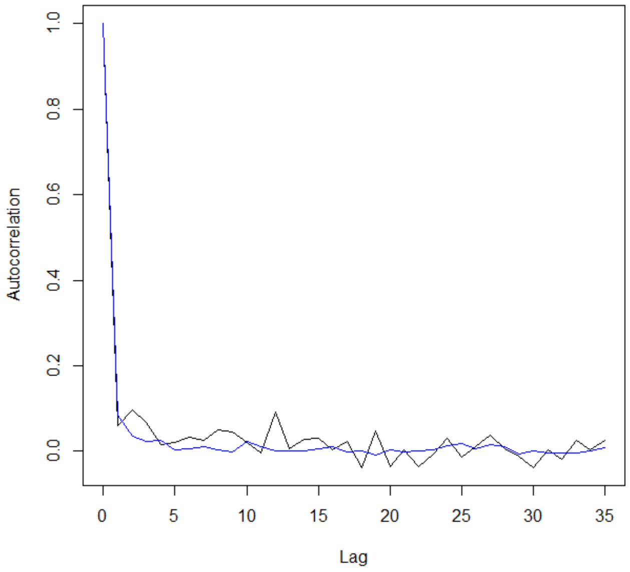


FIGURE C.83: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP30-40G-0

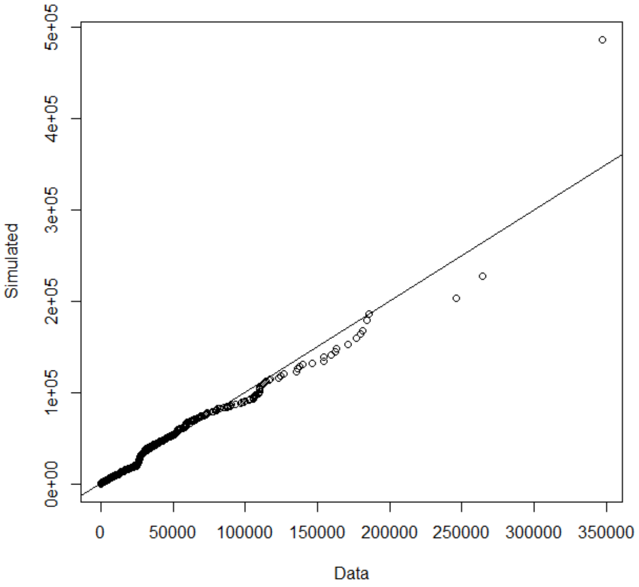


Figure C.84: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP30-40G-0

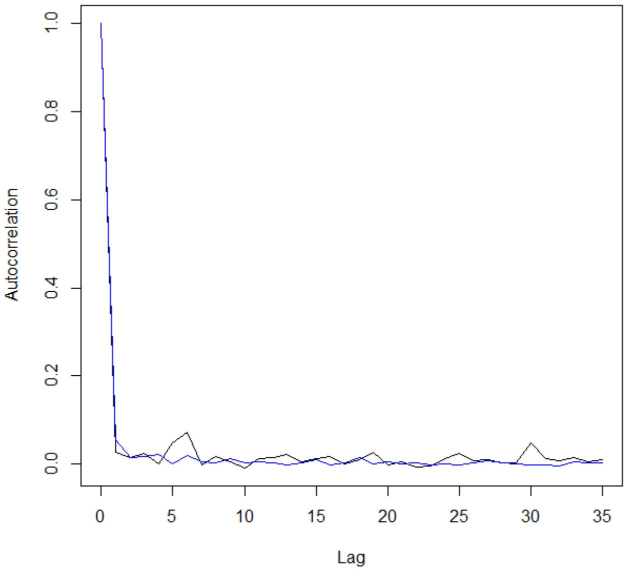


Figure C.85: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP45-50C-0

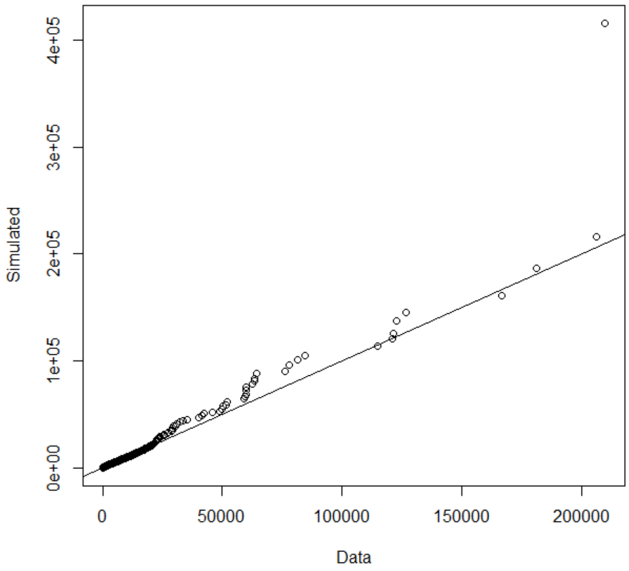


Figure C.86: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP45-50C-0

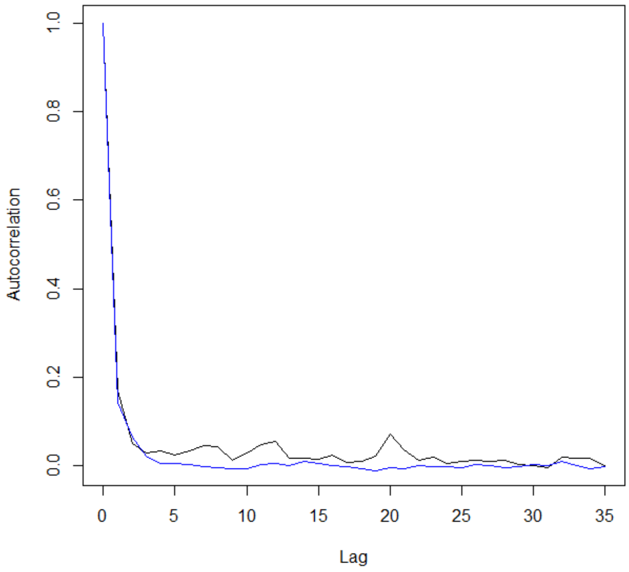


Figure C.87: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP50_BLTRD-0

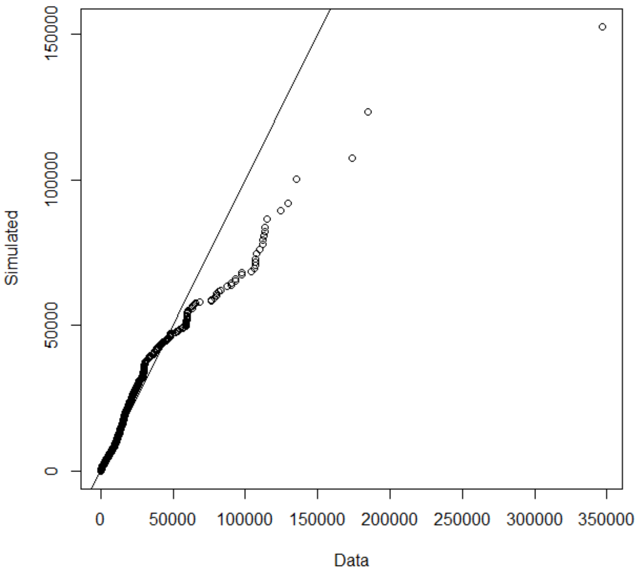


FIGURE C.88: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP50_BLTRD-0

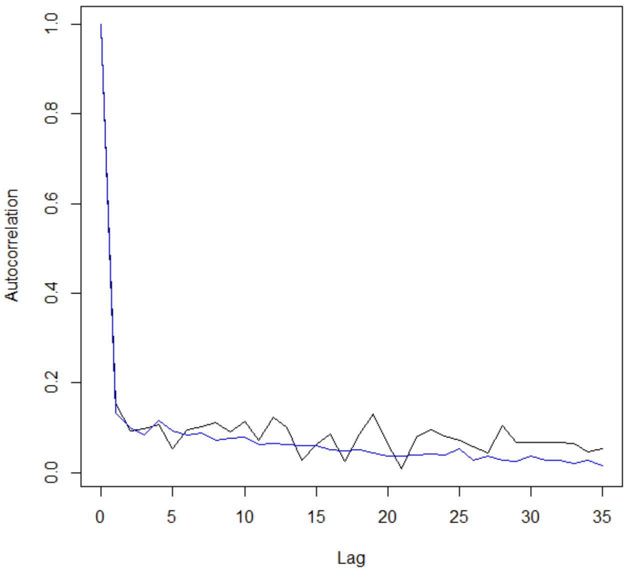


FIGURE C.89: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP55-0

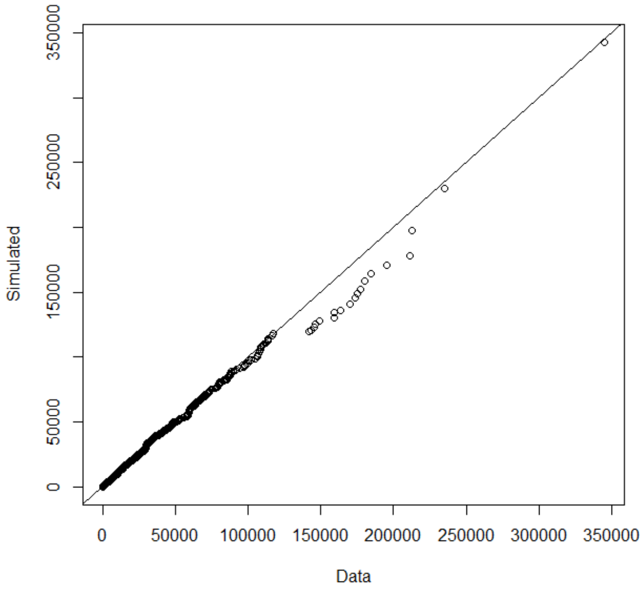


FIGURE C.90: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP55-0

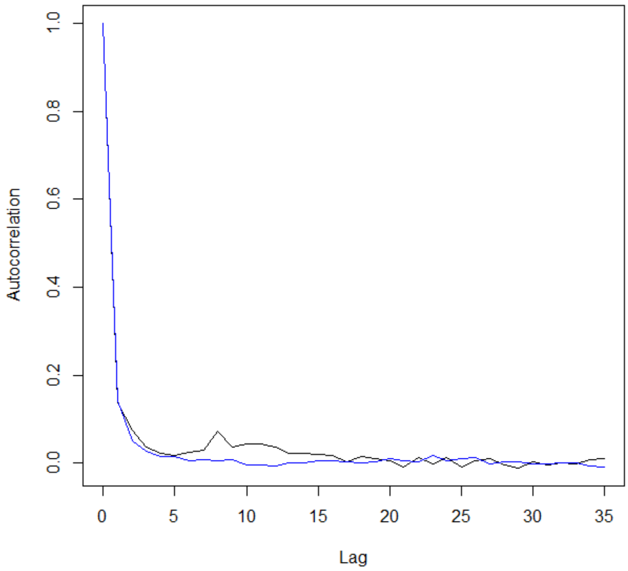


FIGURE C.91: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP70-1

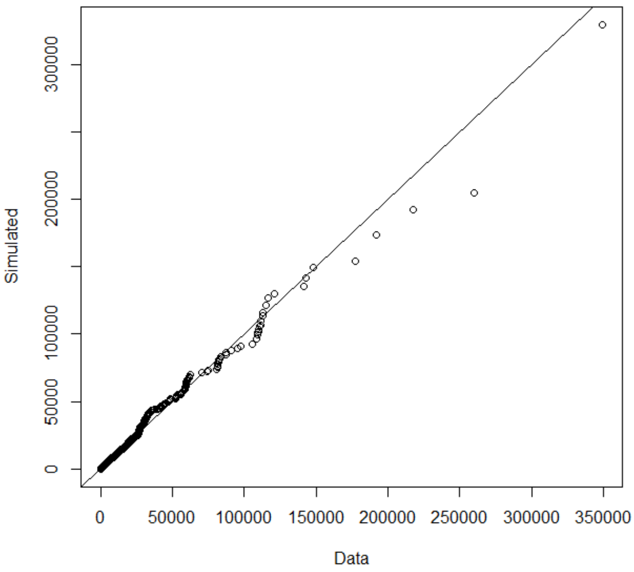


FIGURE C.92: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP70-1

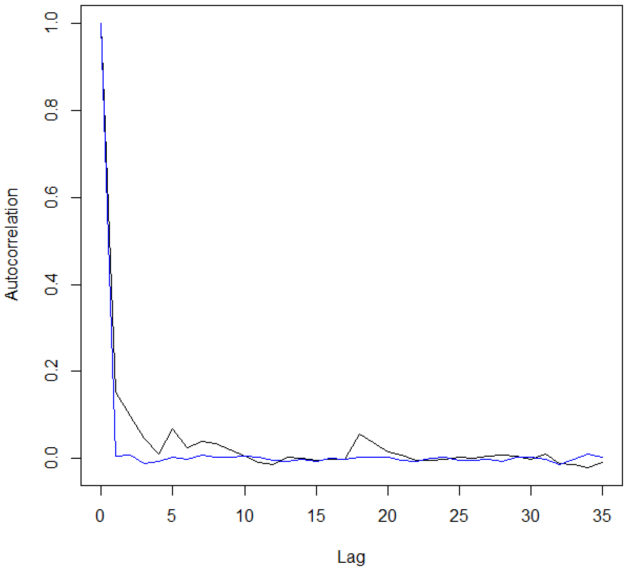


FIGURE C.93: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP70-10

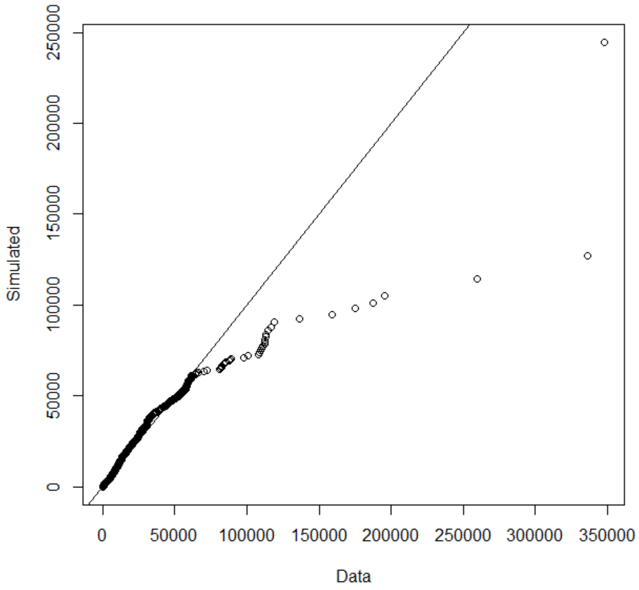


Figure C.94: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP70-10

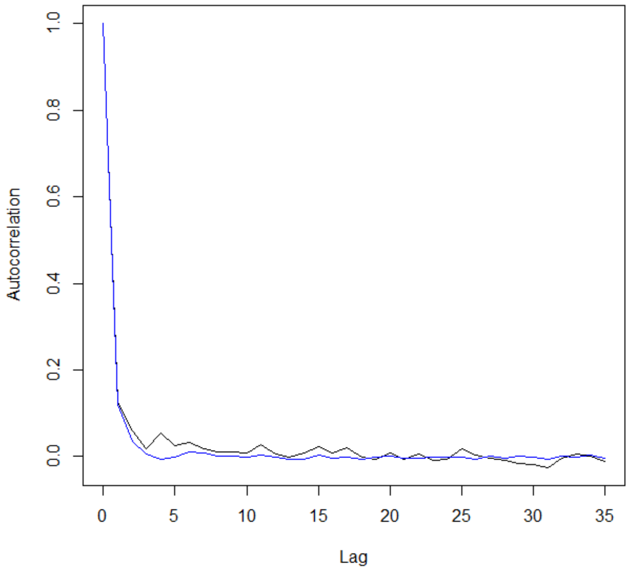


Figure C.95: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP70-12

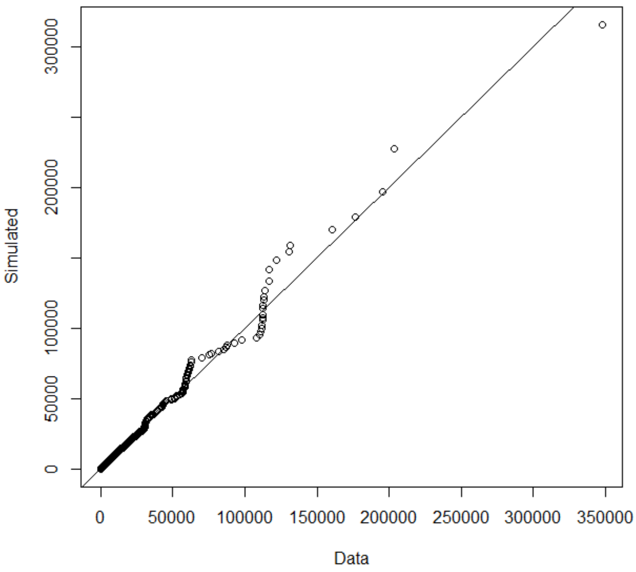


FIGURE C.96: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP70-12

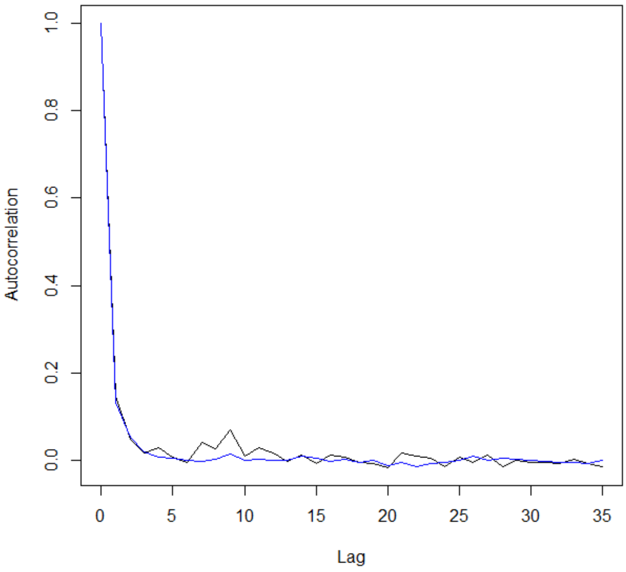


FIGURE C.97: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP70-2

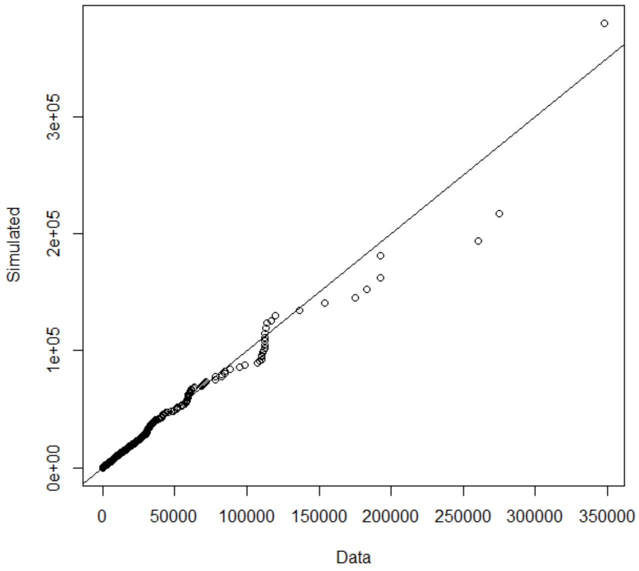


FIGURE C.98: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP70-2

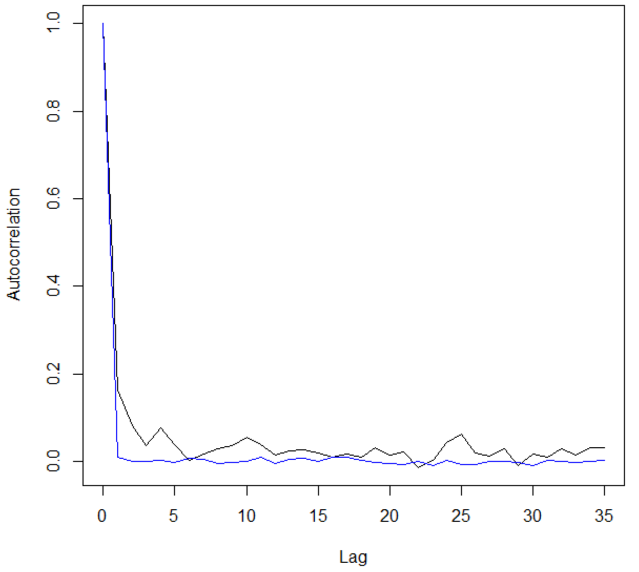


FIGURE C.99: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP70-3

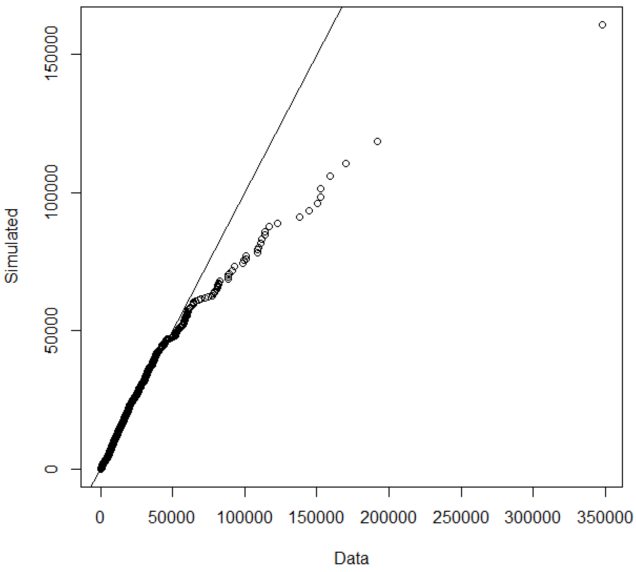


FIGURE C.100: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP70-3

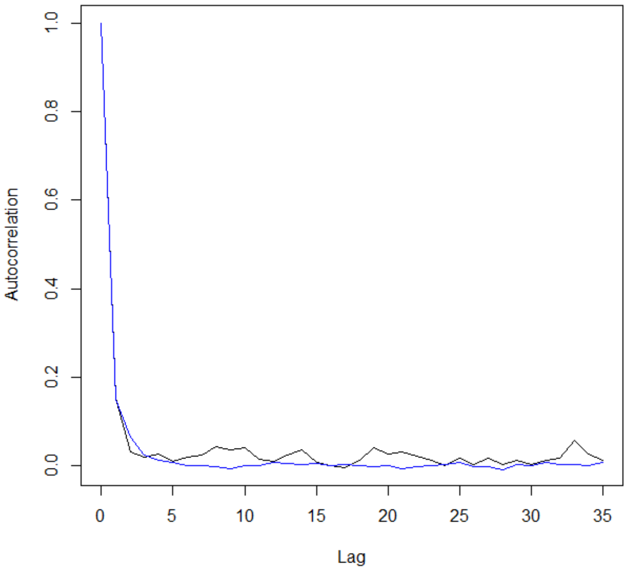


FIGURE C.101: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP70-4

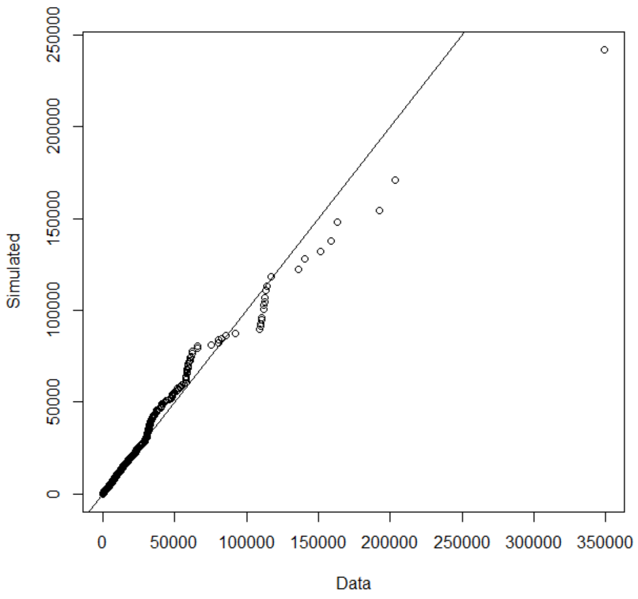


Figure C.102: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP70-4

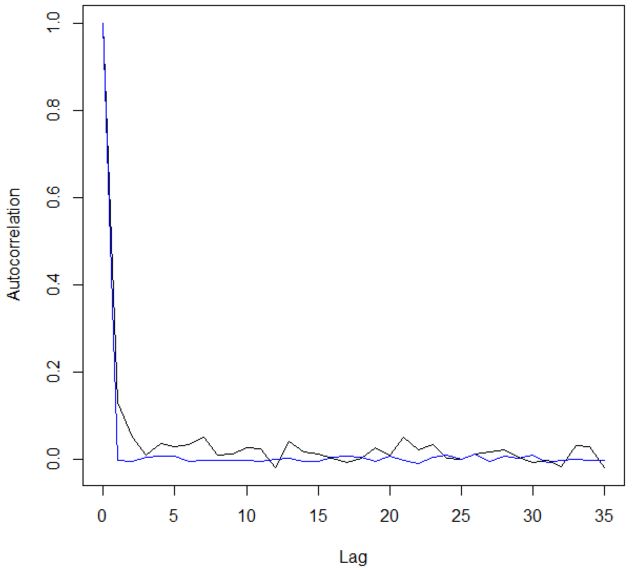


Figure C.103: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP70-5

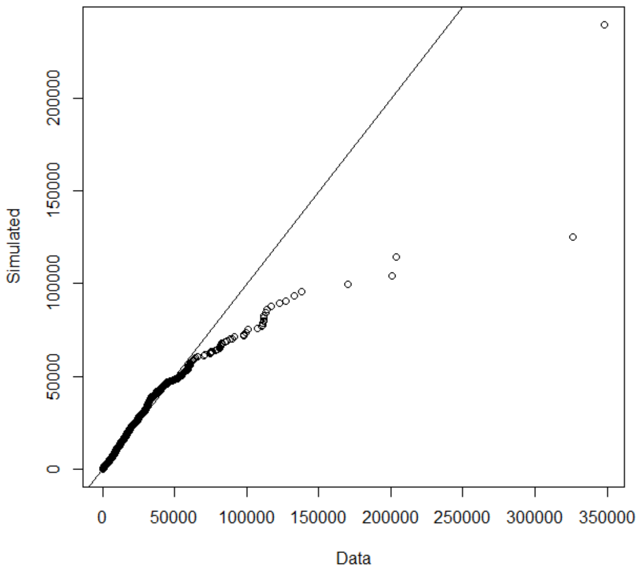


FIGURE C.104: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP70-5

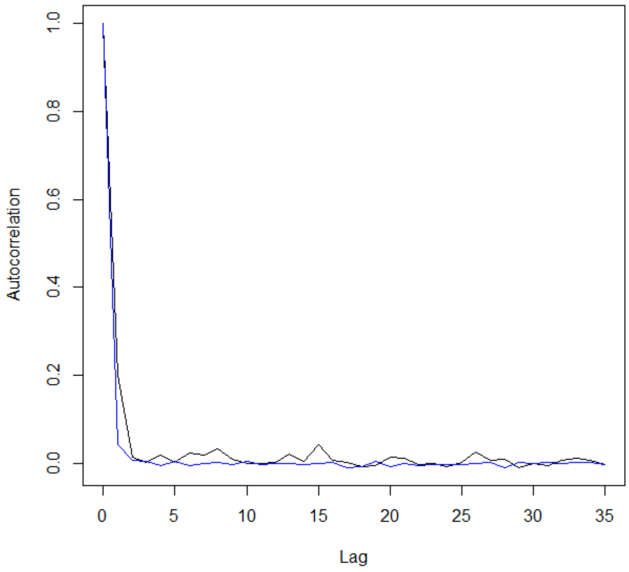


FIGURE C.105: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP70-7

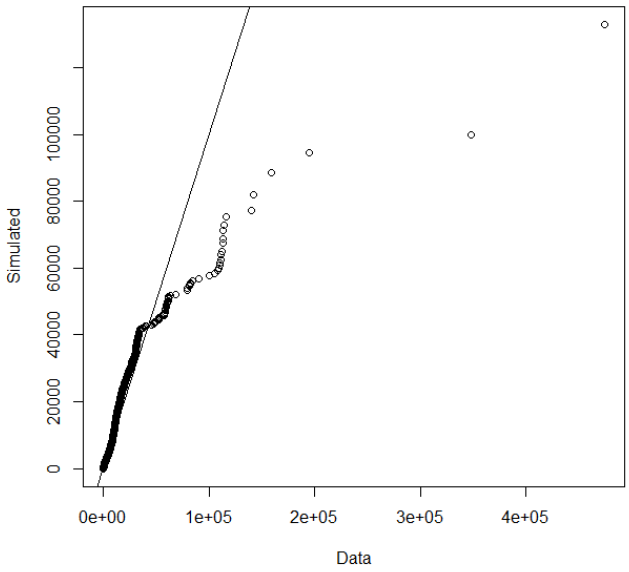


FIGURE C.106: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP70-7

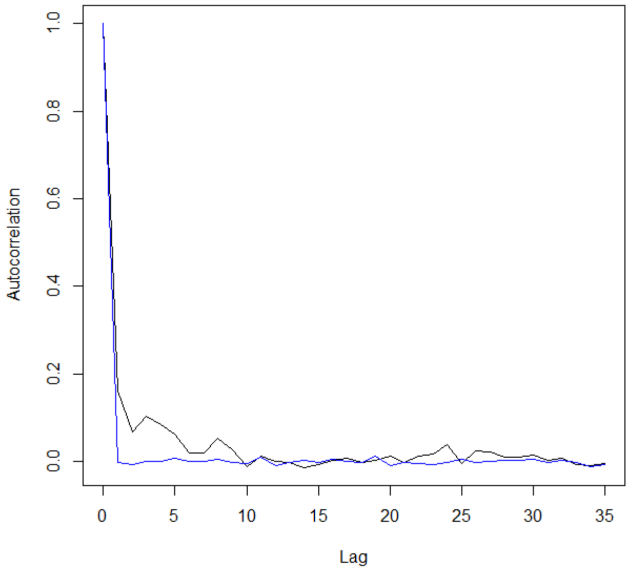


FIGURE C.107: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP70-8

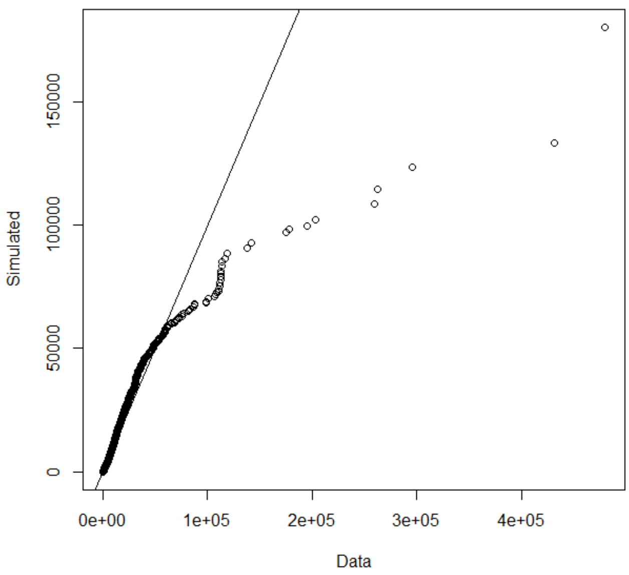


FIGURE C.108: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP70-8

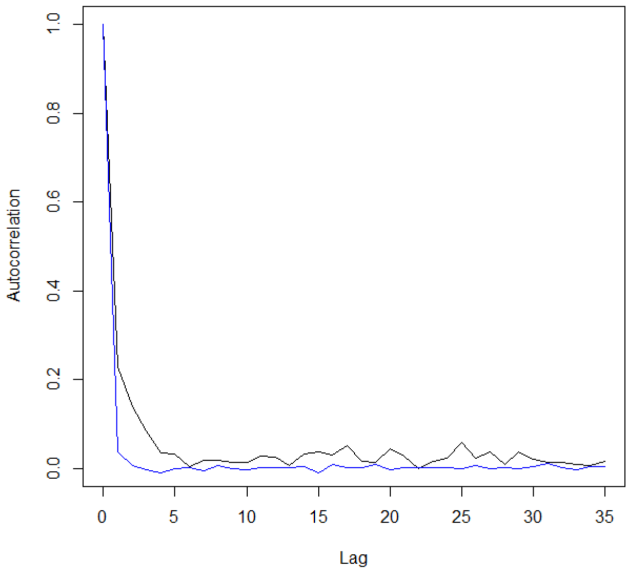


FIGURE C.109: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP70-9

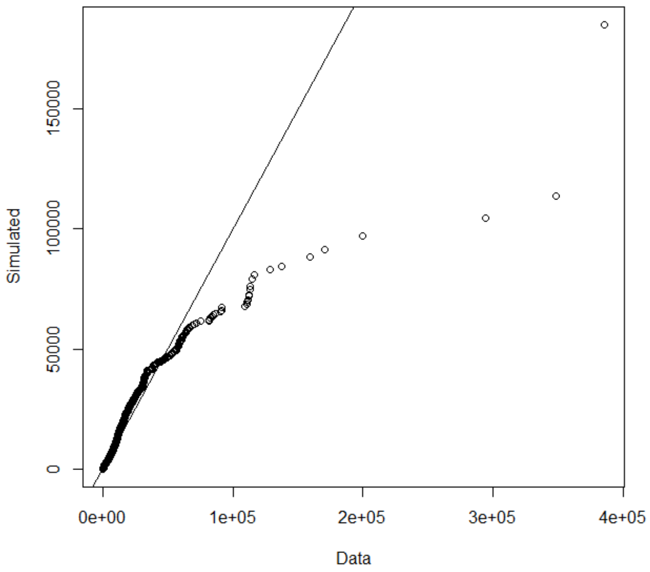


FIGURE C.110: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP70-9

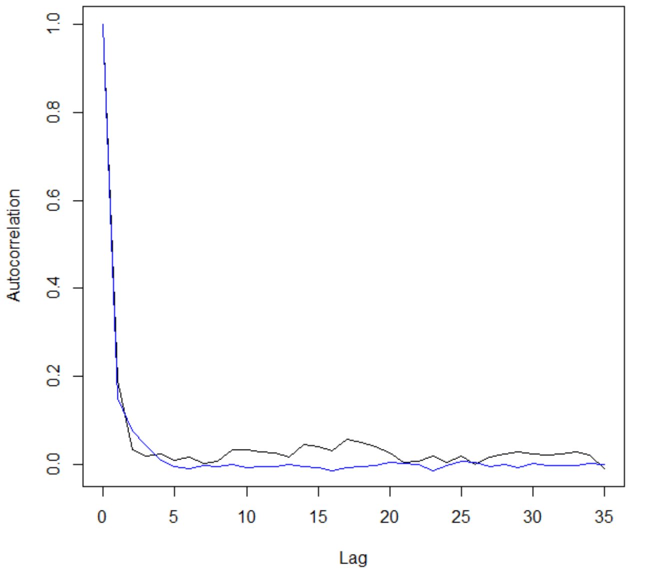


FIGURE C.111: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP70G-0

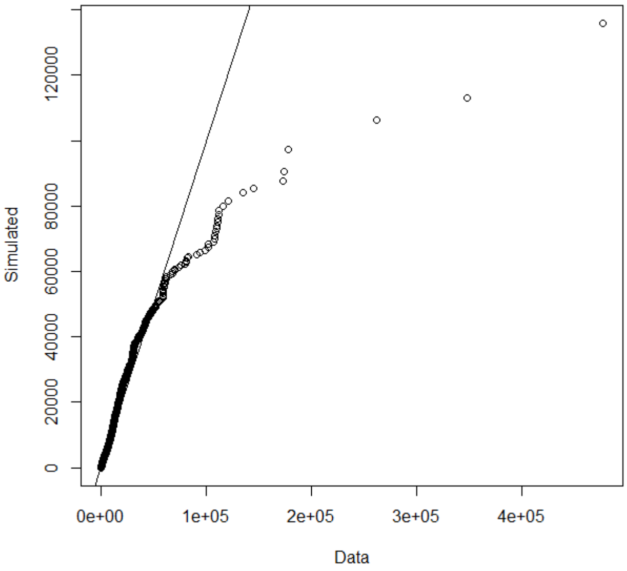


FIGURE C.112: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP70G-0

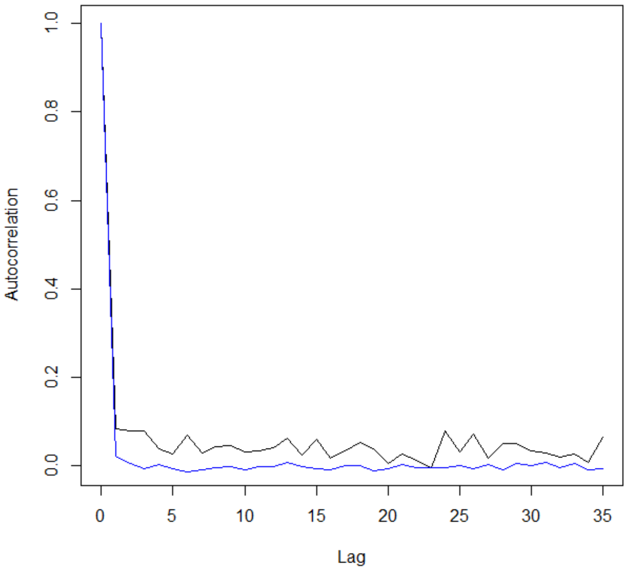


FIGURE C.113: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP80-1

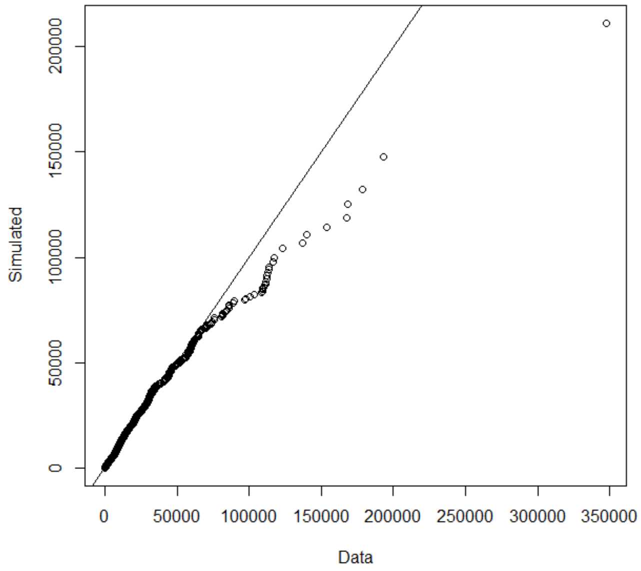


FIGURE C.114: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP80-1

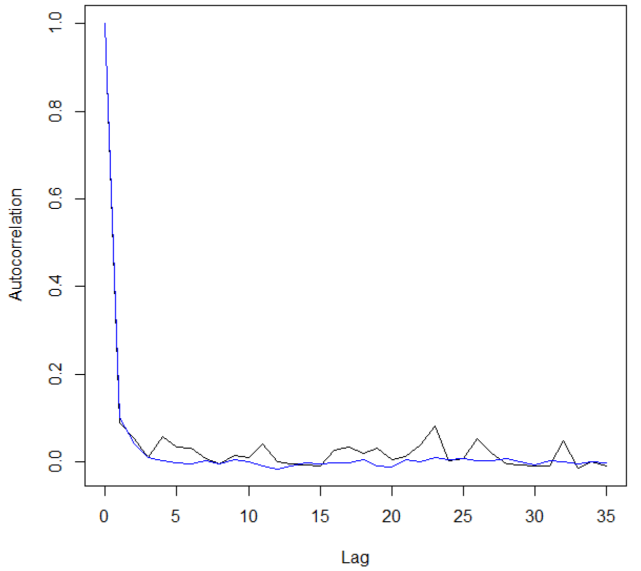


FIGURE C.115: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP80-10

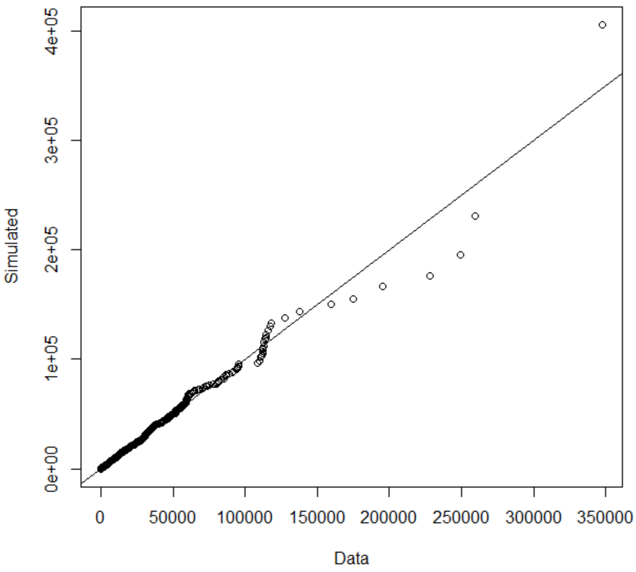


FIGURE C.116: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP80-10

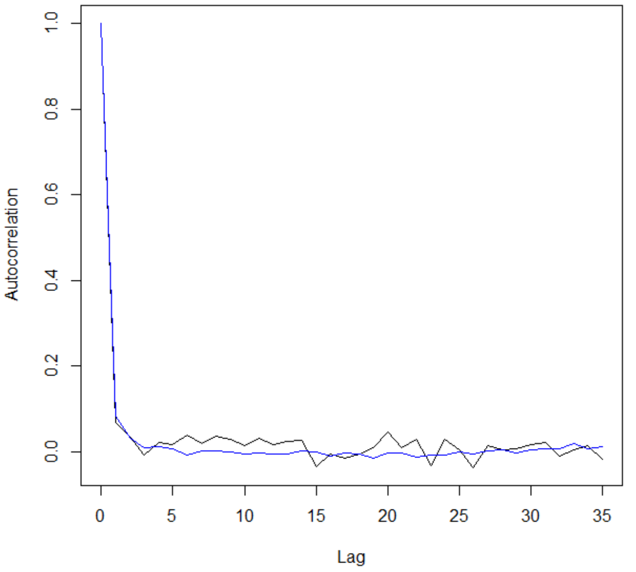


FIGURE C.117: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP80-2

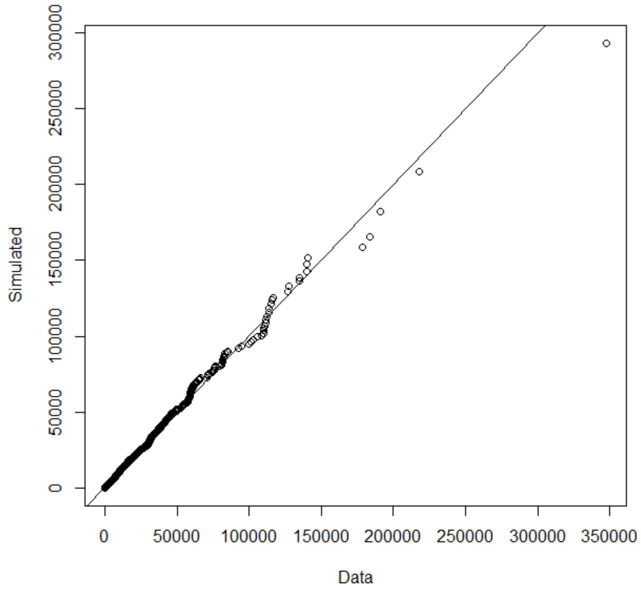


FIGURE C.118: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP80-2

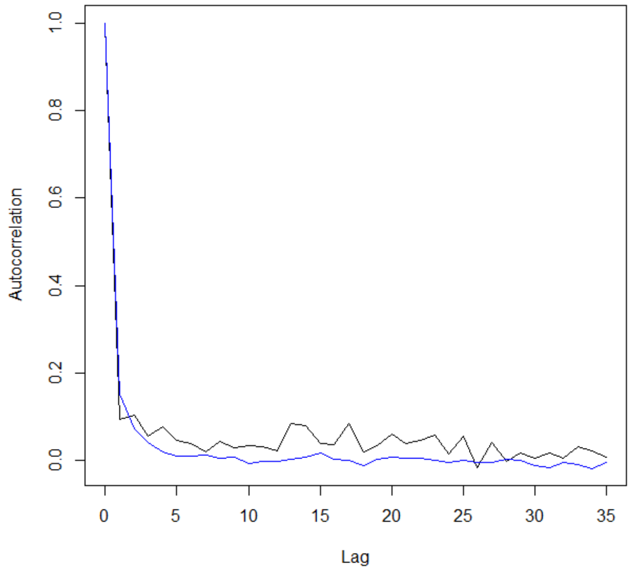


FIGURE C.119: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP80-3

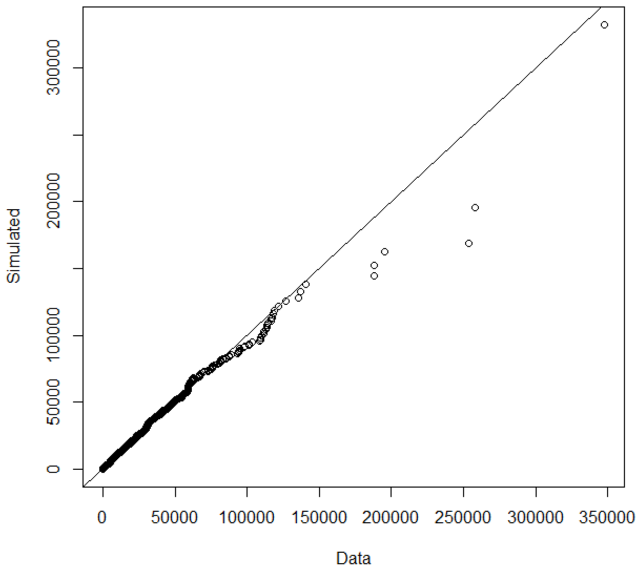


FIGURE C.120: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP80-3

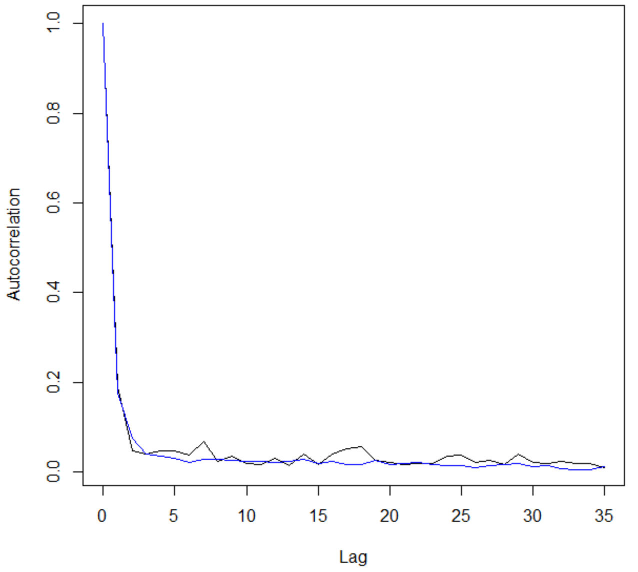


FIGURE C.121: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP80-4

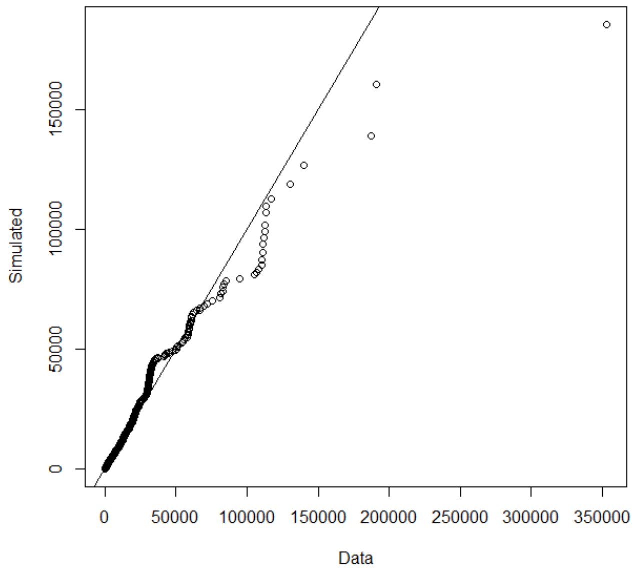


Figure C.122: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP80-4

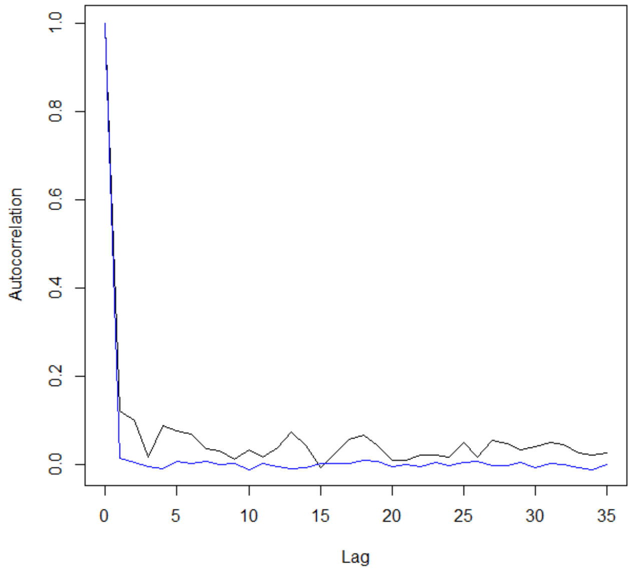


Figure C.123: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP80-5

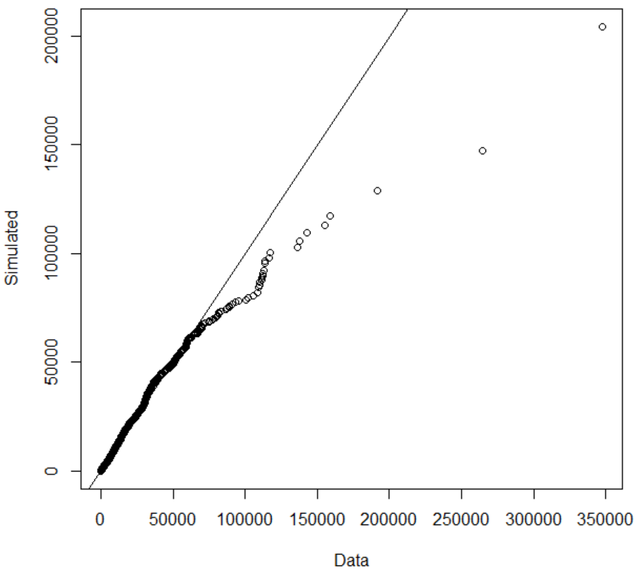


Figure C.124: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP80-5

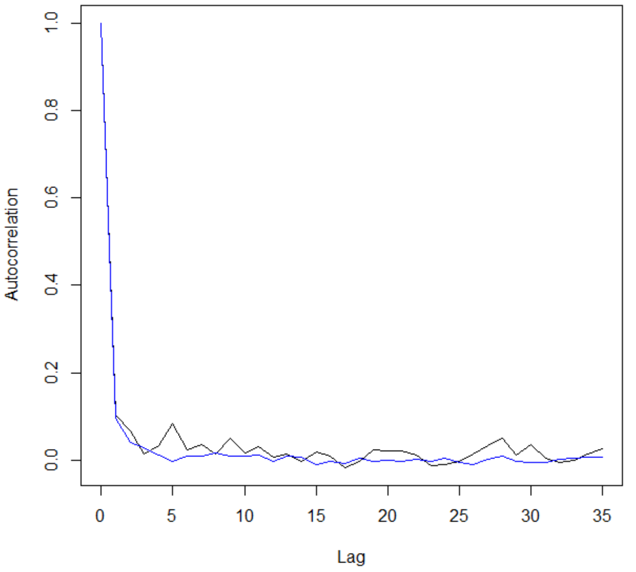


Figure C.125: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP80-6

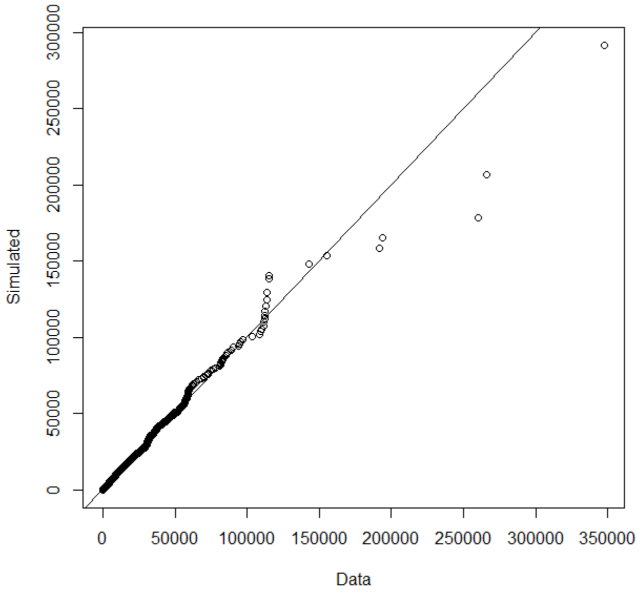


FIGURE C.126: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP80-6

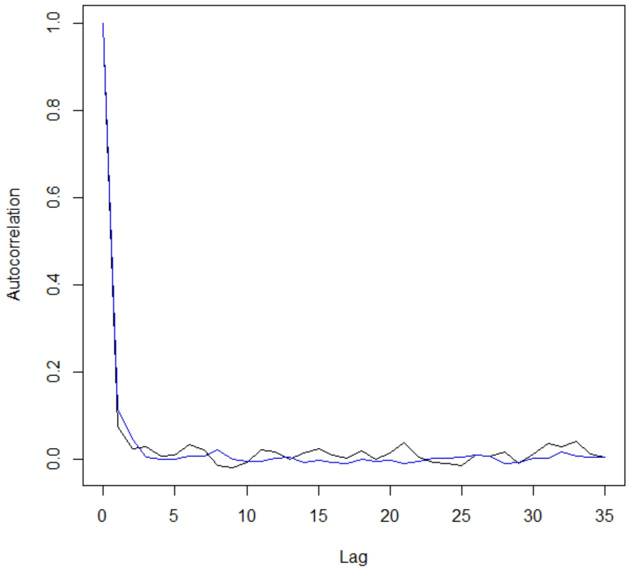


FIGURE C.127: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP80-7

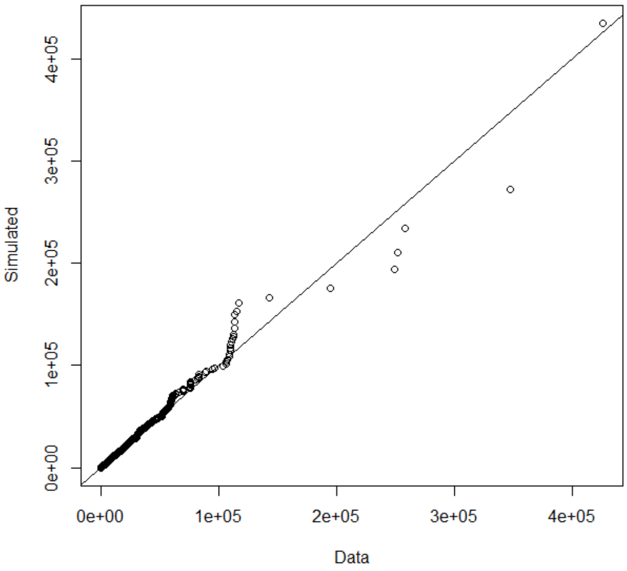


FIGURE C.128: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP80-7

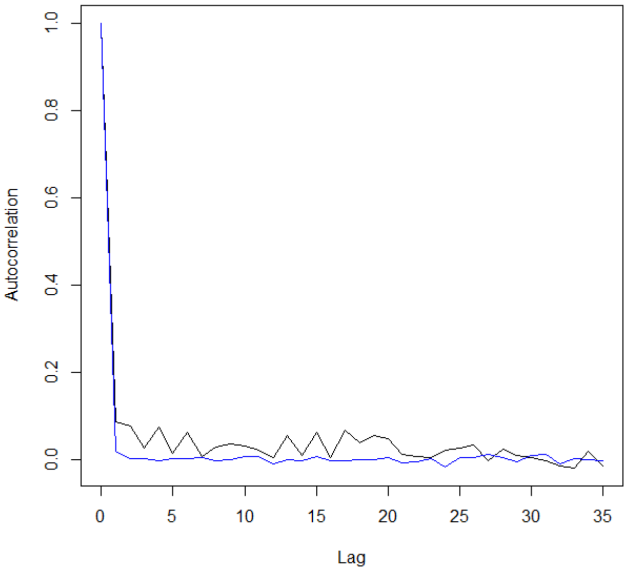


FIGURE C.129: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP80-8

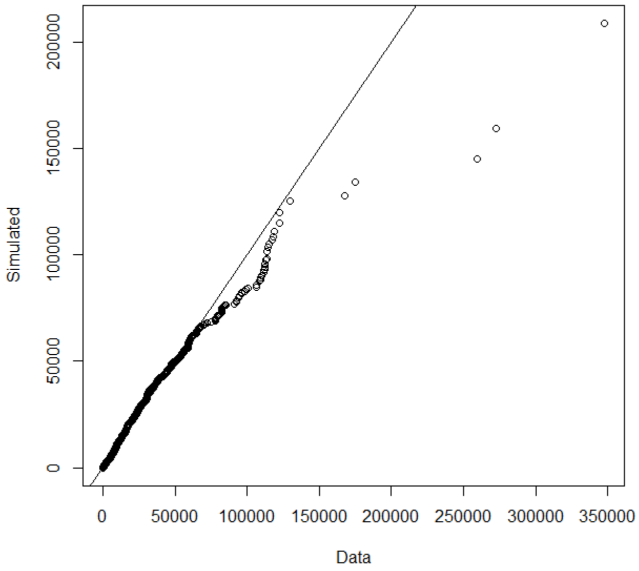


FIGURE C.130: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP80-8

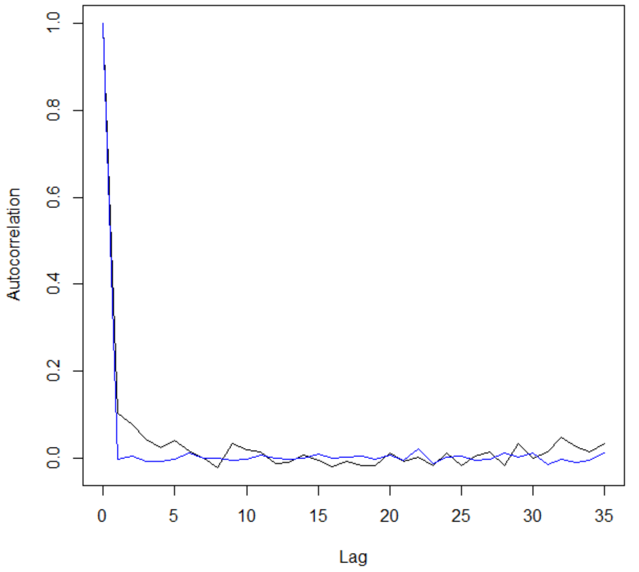


FIGURE C.131: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP80-9

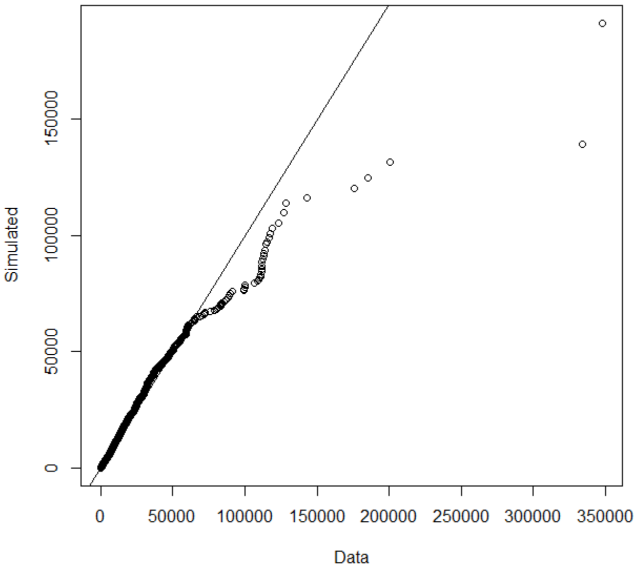


FIGURE C.132: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP80-9

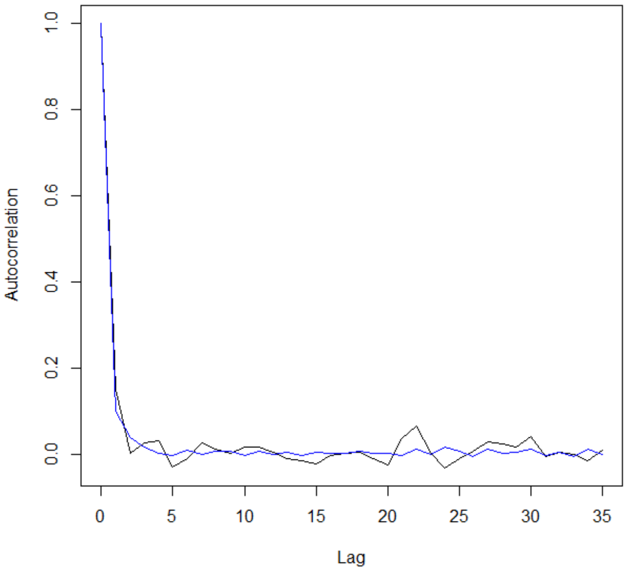


FIGURE C.133: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP80G-0

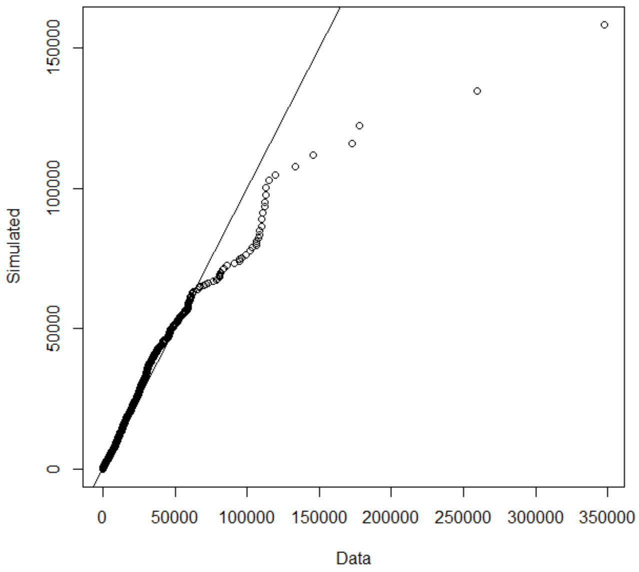


FIGURE C.134: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP80G-0

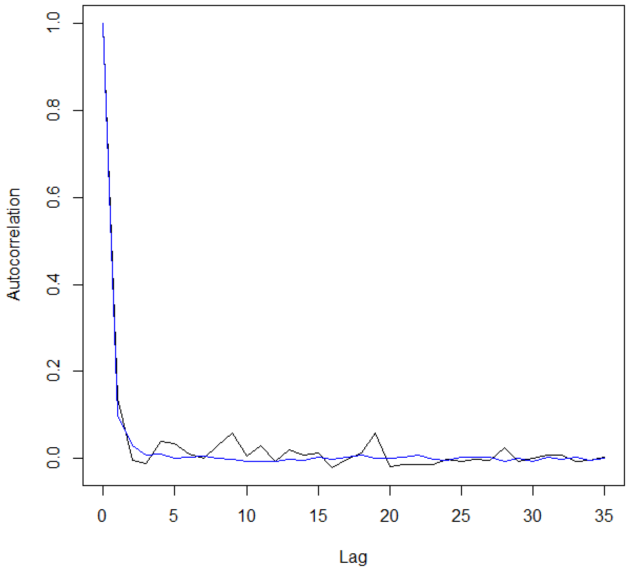


FIGURE C.135: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP90-1

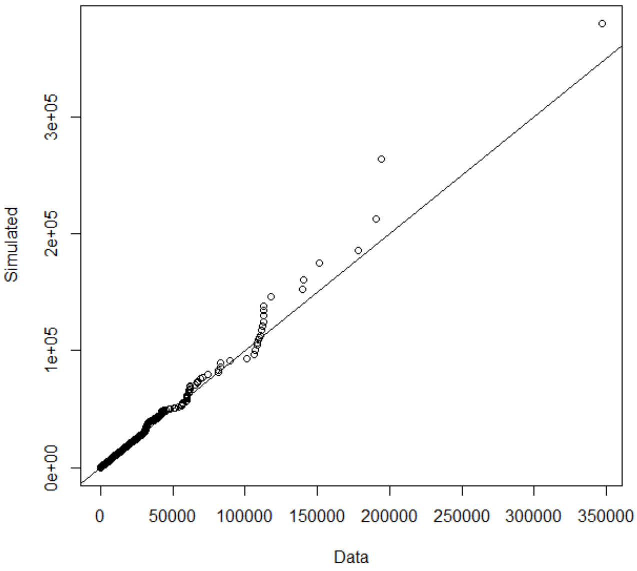


Figure C.136: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP90-1

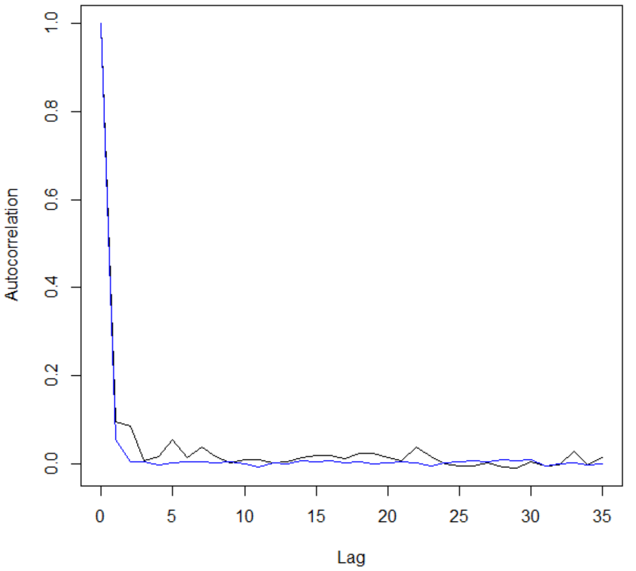


Figure C.137: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP90-10

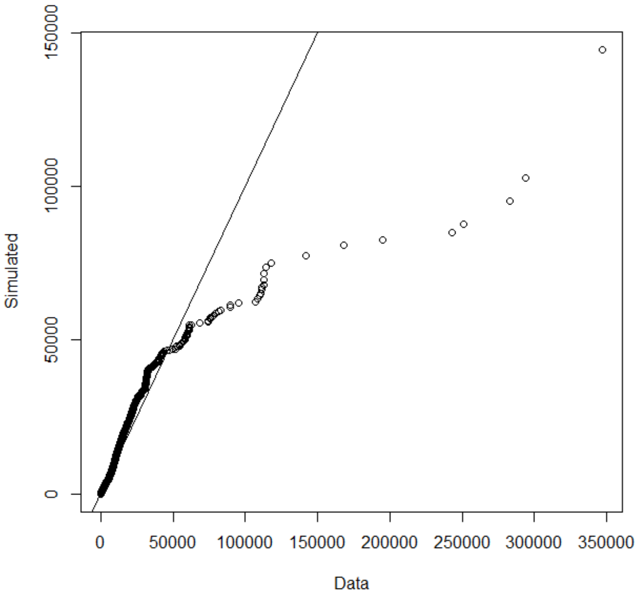


Figure C.138: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP90-10

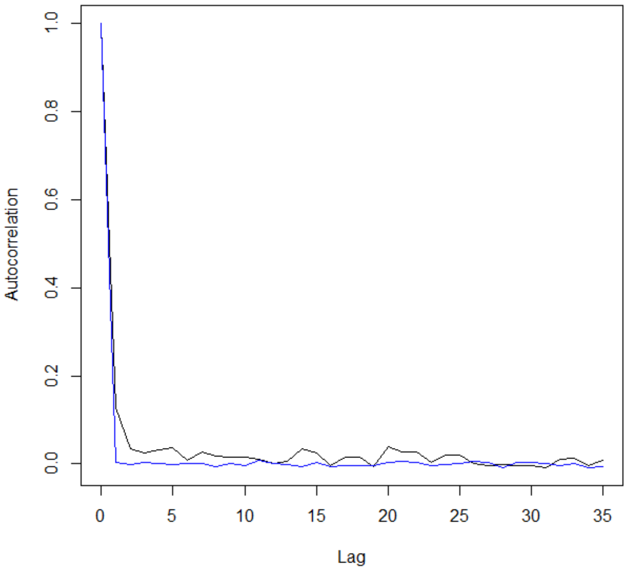


Figure C.139: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP90-12

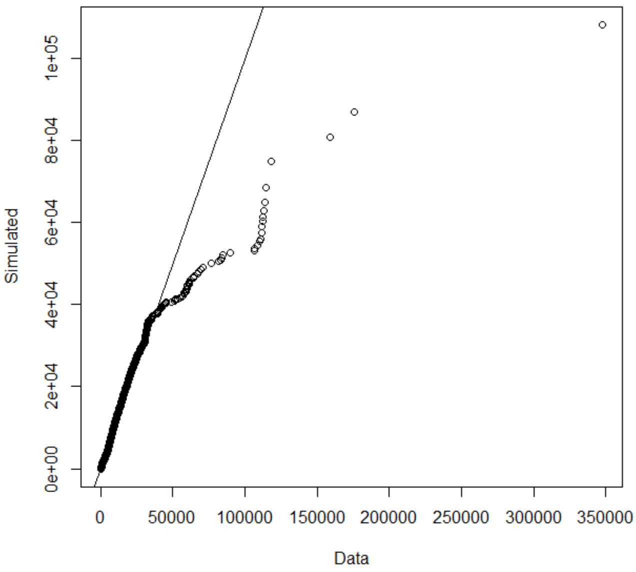


Figure C.140: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP90-12

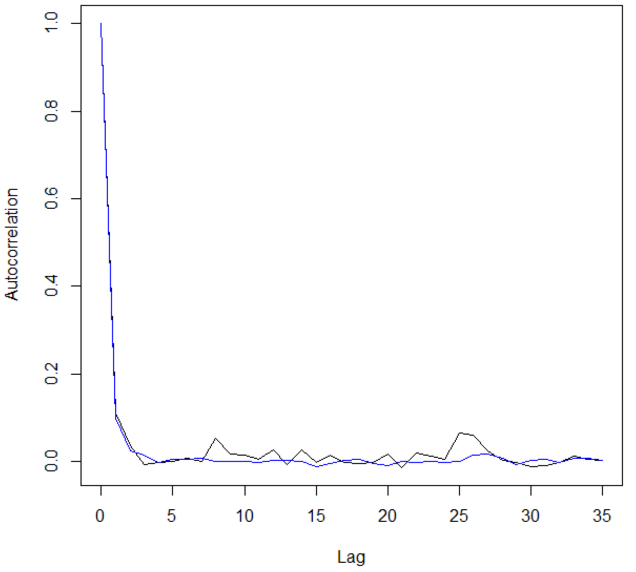


Figure C.141: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP90-2

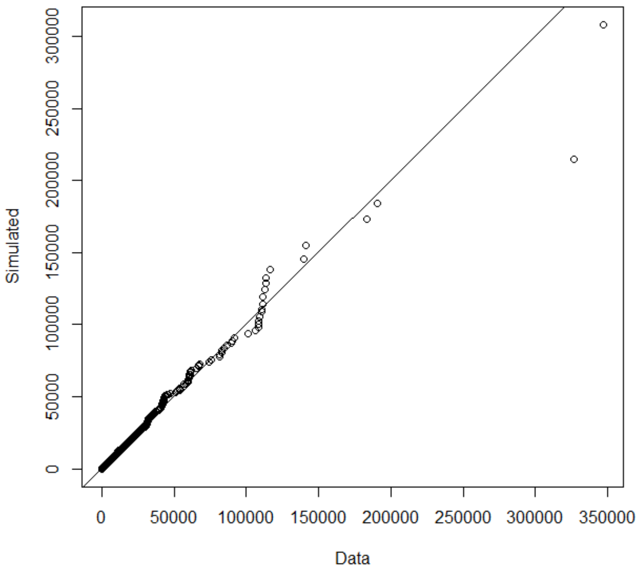


FIGURE C.142: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP90-2

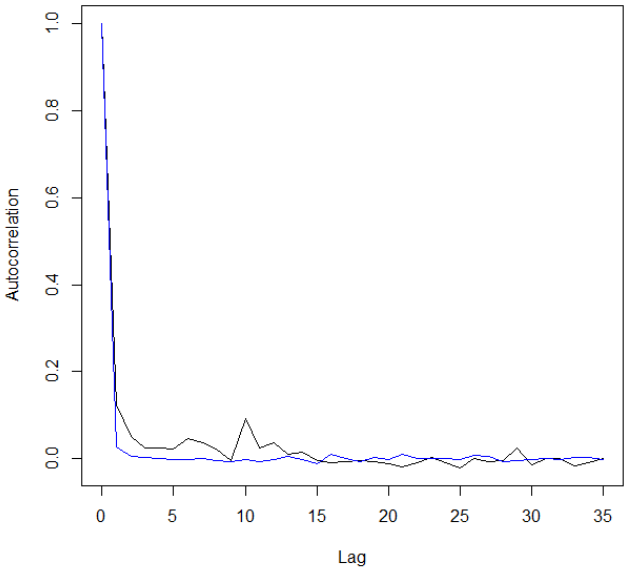


FIGURE C.143: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP90-3

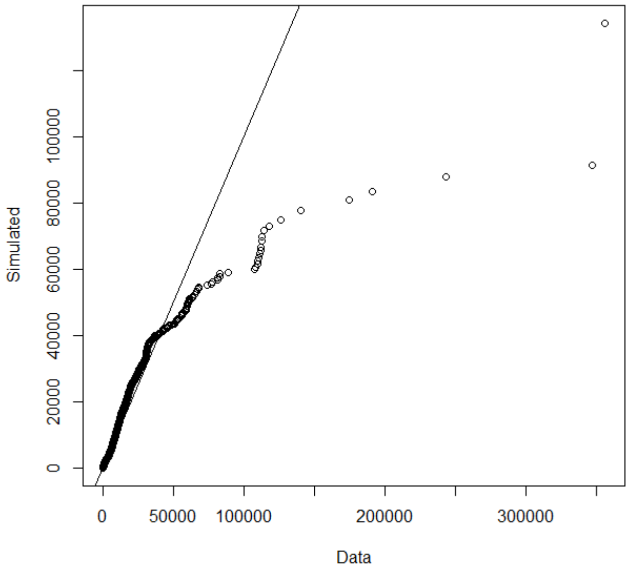


Figure C.144: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP90-3

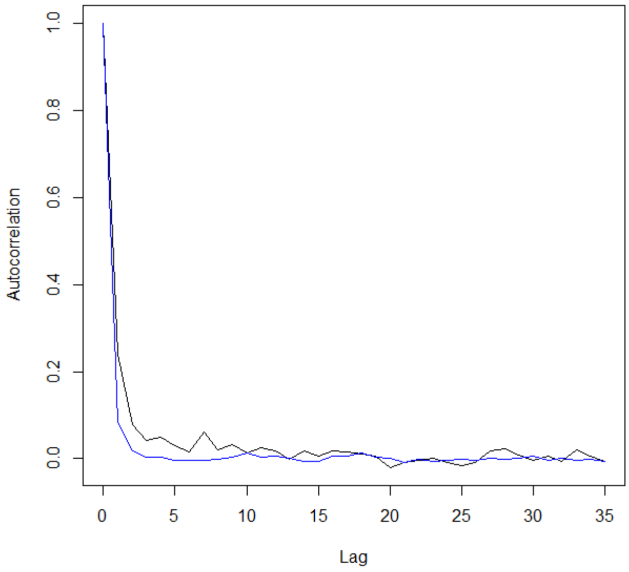


Figure C.145: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP90-4

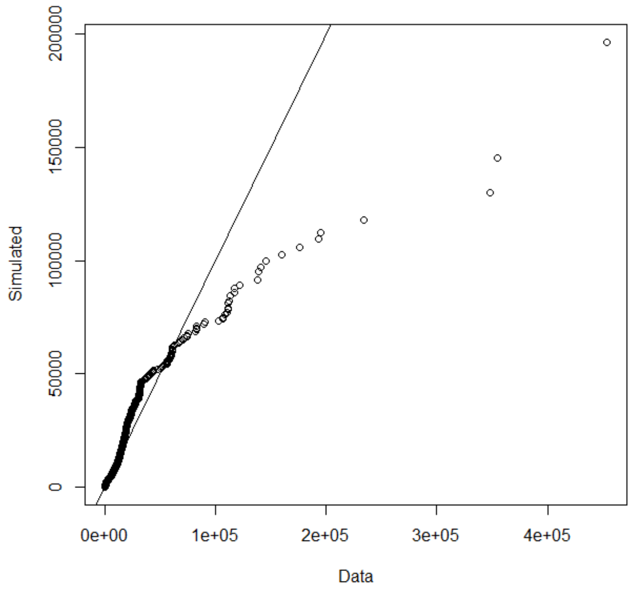


Figure C.146: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP90-4

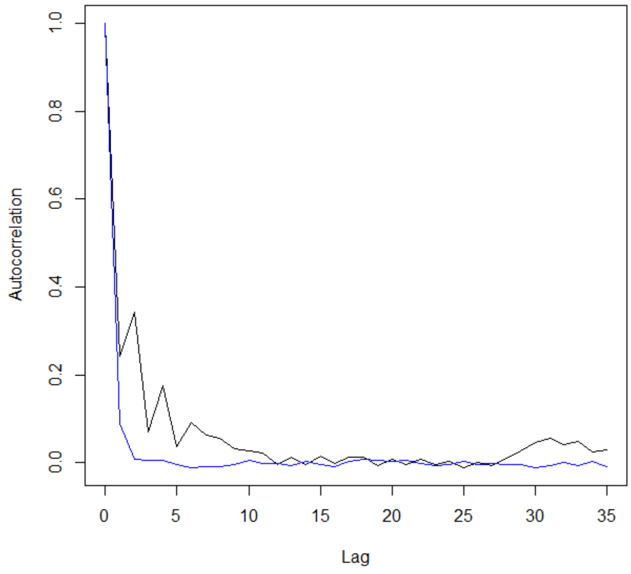


Figure C.147: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP90-5

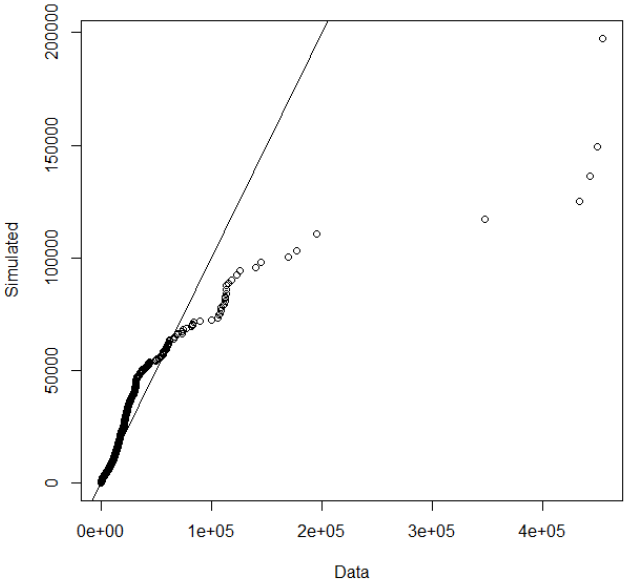


FIGURE C.148: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP90-5

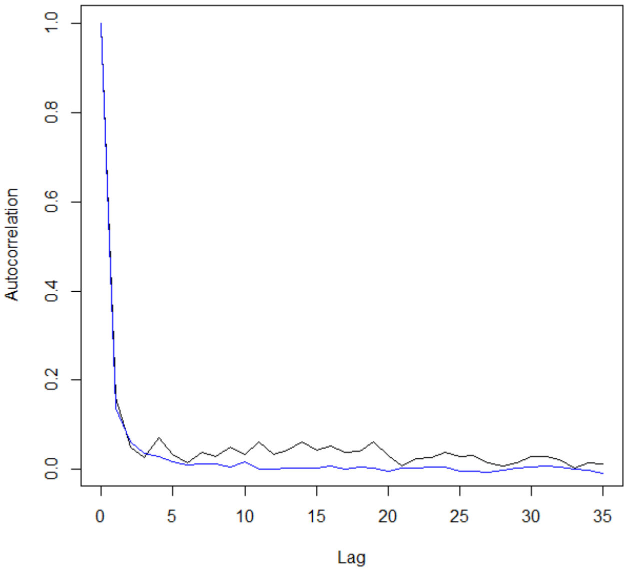


FIGURE C.149: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP90-6

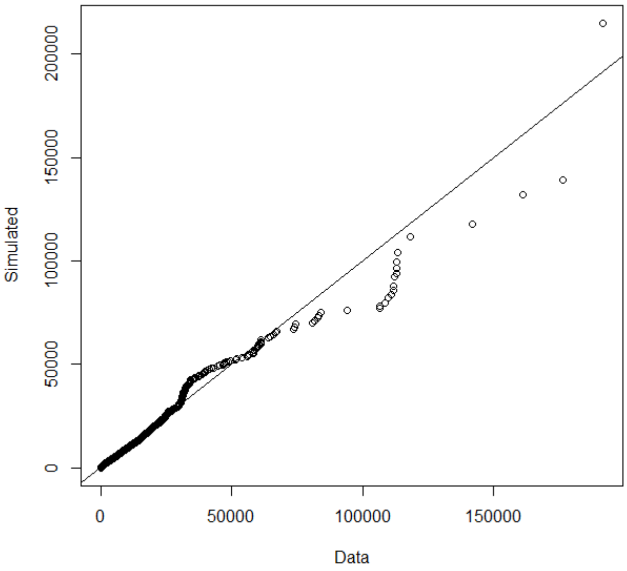


FIGURE C.150: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP90-6

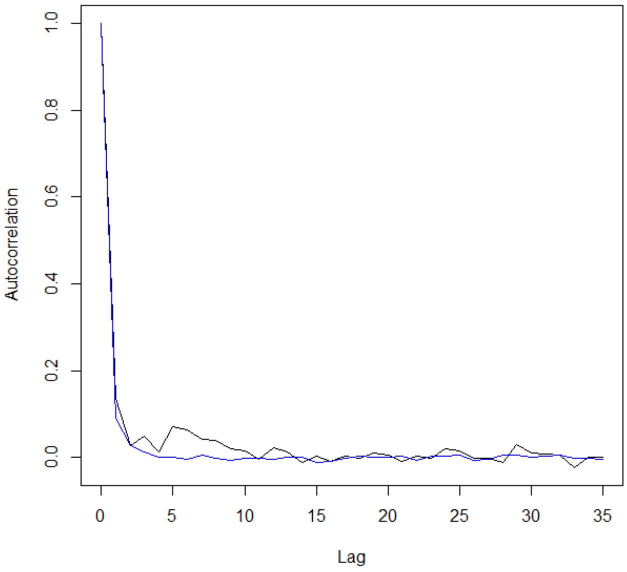


FIGURE C.151: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP90-7

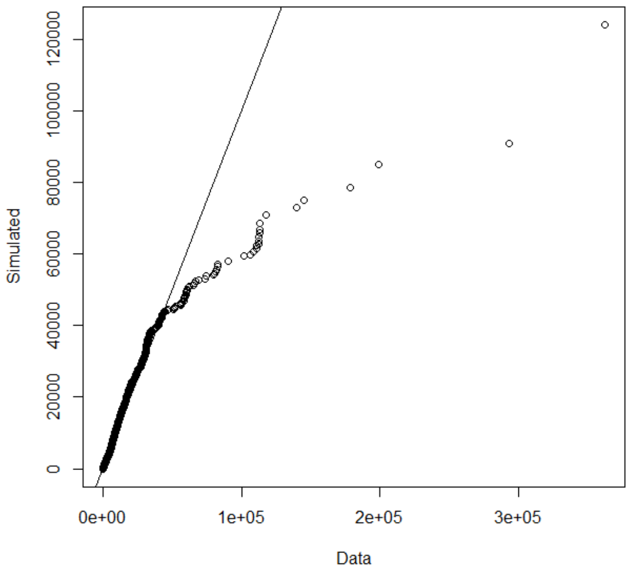


Figure C.152: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP90-7

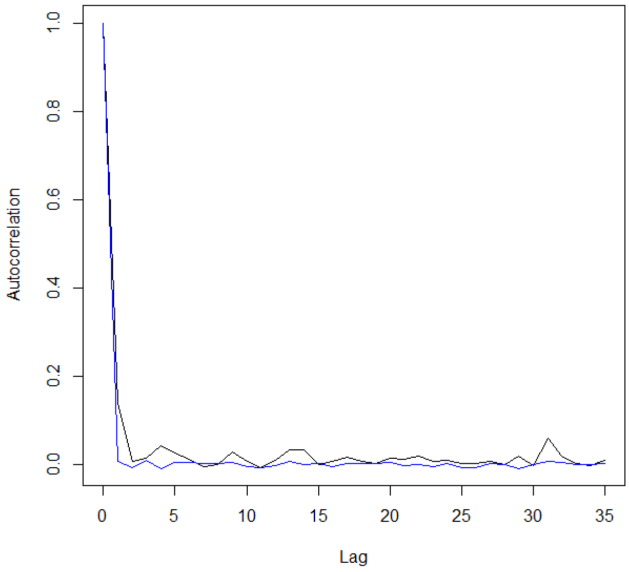


Figure C.153: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP90-8

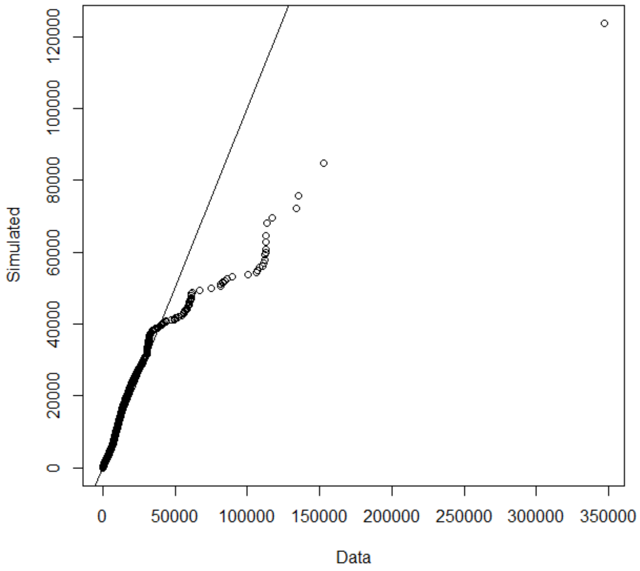


FIGURE C.154: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP90-8

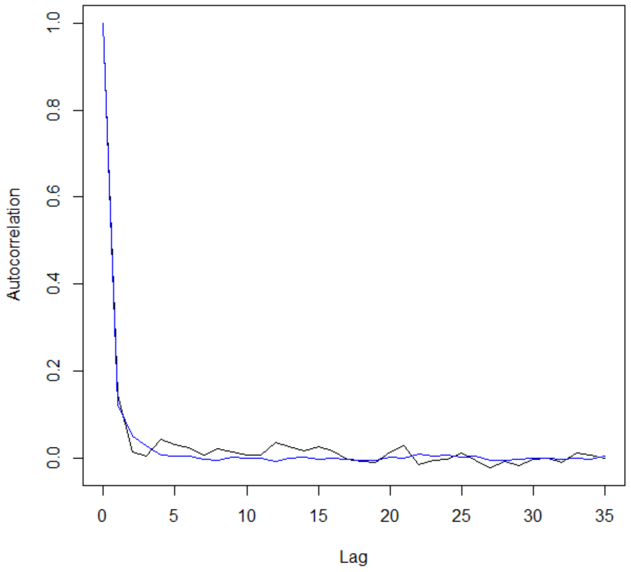


FIGURE C.155: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP90-9

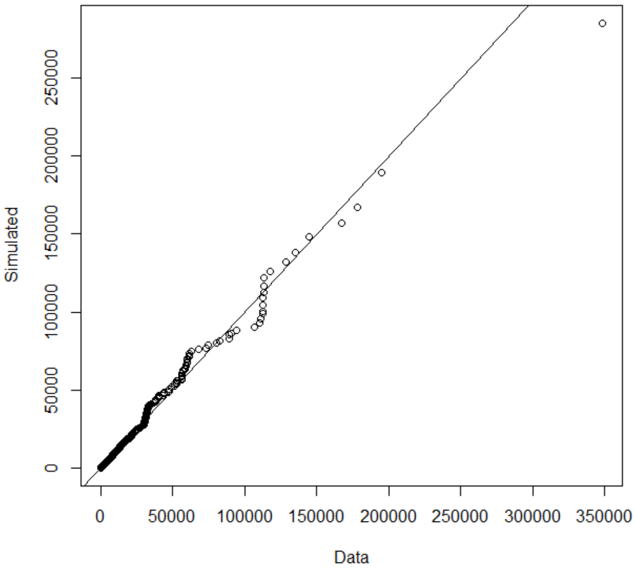


FIGURE C.156: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP90-9

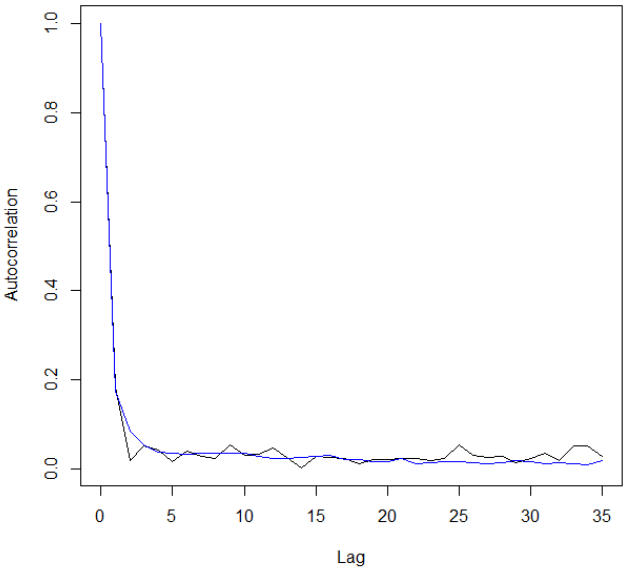


FIGURE C.157: Autocorrelation plot of simulated and actual uptimes of MMPP fit for machine OP90G-0

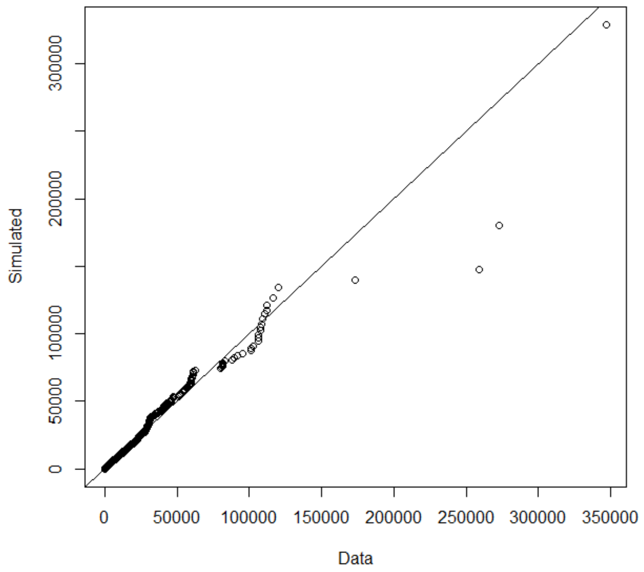


FIGURE C.158: QQ-plot of simulated and actual uptimes of MMPP fit for machine OP90G-0

# Bibliography

[1] Hervé Abdi. Holms sequential bonferroni procedure. *Encyclopedia of research design*, 1(8), 2010.

[2] Tayfur Altiok. Approximate analysis of exponential tandem queues with blocking. *European Journal of Operational Research*, 11(4):390–398, 1982.

[3] Tayfur Altiok. On the phase-type approximations of general distributions. *IIE Transactions*, 17(2):110–116, 1985.

[4] Tayfur Altiok. *Performance analysis of manufacturing systems*. Springer Science & Business Media, 2012.

[5] Tayfur Altiok and Benjamin Melamed. The case for modeling correlation in manufacturing systems. *Iie Transactions*, 33(9):779–791, 2001.

[6] Tayfur Altiok and Benjamin Melamed. The case for modeling correlation in manufacturing systems. *Iie Transactions*, 33(9):779–791, 2001.

[7] Tayfur Altiok and Raghav Ranjan. Analysis of production lines with general service times and finite buffers: a two-node decomposition approach. *Engineering Costs and Production Economics*, 17(1-4):155–165, 1989.

[8] Tayfur M Altiok and Shaler Stidham Jr. A note on transfer lines with unreliable machines, random processing times, and finite buffers. *IIE Transactions*, 14(2):125–127, 1982.

[9] GT Artamonov. Productivity of a two-instrument discrete processing line in the presence of failures. *Cybernetics*, 12(3):464–468, 1976.

[10] MS Bartlett. The spectral analysis of point processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 264–296, 1963.

[11] Forest Baskett, K Mani Chandy, Richard R Muntz, and Fernando G Palacios. Open, closed, and mixed networks of queues with different classes of customers. *Journal of the ACM (JACM)*, 22(2):248–260, 1975.

[12] Richard F Bass. *Stochastic processes*, volume 33. Cambridge University Press, 2011.

[13] Anil K Bera and Matthew L Higgins. Arch models: properties, estimation and testing. *Journal of economic surveys*, 7(4):305–366, 1993.

[14] Jan Beran, Robert Sherman, Murad S Taqqu, and Walter Willinger. Long-range dependence in variable-bit-rate video traffic. *IEEE Transactions on communications*, 43(234):1566–1579, 1995.

[15] Oded Berman. Efficiency and production rate of a transfer line with two machines and a finite storage buffer. *European Journal of Operational Research*, 9(3):295–308, 1982.

[16] JM Bernardo, MJ Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, and M West. The markov modulated poisson process and markov poisson cascade with applications to web traffic modeling. *Bayesian Statistics*, 2003.

[17] Dennis E Blumenfeld. A simple formula for estimating throughput of serial production lines with variable processing times and limited buffer capacity. *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, 28(6):1163–1182, 1990.

[18] A Bobbio and KS Trivedi. Computation of the completion time when the work requirement is a ph random variable. In *Int. Conf. on Analysis and Control of Large Scale Stochastic Systems*, 1988.

[19] Paula R Bouzas, Mariano J Valderrama, and Ana M Aguilera. On the characteristic functional of a doubly stochastic poisson process: application to a narrow-band process. *Applied mathematical modelling*, 30(9):1021–1032, 2006.

[20] George EP Box and David R Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 211–252, 1964.

[21] George EP Box and Gwilym M Jenkins. Time series analysis, control, and forecasting. *San Francisco, CA: Holden Day*, 3226(3228):10, 1976.

[22] George EP Box, Gwilym M Jenkins, and Gregory C Reinsel. *Time series analysis: forecasting and control*, volume 734. John Wiley & Sons, 2011.

[23] George EP Box, Mervin E Muller, et al. A note on the generation of random normal deviates. *The annals of mathematical statistics*, 29(2):610–611, 1958.

[24] JA Buzacott. *Markov chain analysis of automatic transfer line with buffer stock*. PhD thesis, The University of Birmingham, 1967.

[25] JA Buzacott and LE Hanifin. Models of automatic transfer lines with inventory banks a review and comparison. *AIIE Transactions*, 10(2):197–207, 1978.

[26] John A Buzacott. Automatic transfer lines with buffer stocks. *The International Journal of Production Research*, 5(3):183–200, 1967.

[27] John A Buzacott. The effect of station breakdowns and random processing times on the capacity of flow lines with in-process storage. *AIIE Transactions*, 4(4):308–312, 1972.

[28] John A Buzacott and J George Shanthikumar. *Stochastic models of manufacturing systems*, volume 4. Prentice Hall Englewood Cliffs, NJ, 1993.

[29] II Carson, David M Nicol, Barry L Nelson, Jerry Banks, et al. Discrete-event system simulation, 2005.

[30] Erhan Çinlar. *Introduction to stochastic processes*. Prentice-Hall, Englewood Cliffs, NJ, 1975.

[31] Ismail Civelek, Bahar Biller, and Alan Scheller-Wolf. The impact of dependence on queueing systems. 2009.

[32] David R Cox. Some statistical methods connected with series of events. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 129–164, 1955.

[33] Mark E Crovella and Azer Bestavros. Self-similarity in world wide web traffic: evidence and possible causes. *IEEE/ACM Transactions on networking*, 5(6):835–846, 1997.

[34] Yves Dallery, Rene David, and Xiao-Lan Xie. An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers. *IIE transactions*, 20(3):280–283, 1988.

[35] Yves Dallery and Yannick Frein. On decomposition methods for tandem queueing networks with blocking. *Operations research*, 41(2):386–399, 1993.

[36] Yves Dallery and Stanley B Gershwin. Manufacturing flow line systems: a review of models and analytical results. *Queueing systems*, 12(1-2):3–94, 1992.

[37] Alan De Genaro Dario and Adilson Simonis. Properties of doubly stochastic poisson process with affine intensity. *arXiv preprint arXiv:1109.2884*, 2011.

[38] AC Davison and NI Ramesh. Some models for discretized series of events. *Journal of the American Statistical Association*, 91(434):601–609, 1996.

[39] Li Deng and Jon W Mark. Parameter estimation for markov modulated poisson processes via the em algorithm with time discretization. *Telecommunication Systems*, 1(1):321–338, 1993.

[40] Esra Dogan-Sahiner and Tayfur Altiok. Planning repair effort in transfer lines. *IIE transactions*, 30(10):867–881, 1998.

[41] Qing Du. A monotonicity result for a single-server queue subject to a markov-modulated poisson process. *Journal of Applied Probability*, 32(4):1103–1111, 1995.

[42] T.J. Durbin and S.J. Koopman. *Time Series Analysis by State Space Methods: Second Edition*. Oxford Statistical Science Series. Oxford University Press, 2012.

[43] Wolfgang Fischer and Kathleen Meier-Hellstern. The markov-modulated poisson process (mmpp) cookbook. *Performance evaluation*, 18(2):149–171, 1993.

[44] Victor S Frost and Benjamin Melamed. Traffic modeling for telecommunications networks. *IEEE Communications Magazine*, 32(3):70–81, 1994.

[45] DP Gaver Jr. A waiting line with interrupted service, including priorities. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 73–90, 1962.

[46] Stanley B Gershwin. An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking. *Operations research*, 35(2):291–305, 1987.

[47] Stanley B Gershwin. *Manufacturing systems engineering*. Wiley, 1994.

[48] Stanley B Gershwin and Oded Berman. Analysis of transfer lines consisting of two unreliable machines with random processing times and finite storage buffers. *AIIE transactions*, 13(1):2–11, 1981.

[49] Stanley B Gershwin and Irvin C Schick. Modeling and analysis of three-stage transfer lines with unreliable machines and finite buffers. *Operations Research*, 31(2):354–380, 1983.

[50] Stanley B Gershwin and James E Schor. Efficient algorithms for buffer space allocation. *Annals of Operations research*, 93(1):117–144, 2000.

[51] Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. *International journal of pattern recognition and artificial intelligence*, 15(01):9–42, 2001.

[52] Jan Grandell. *Doubly stochastic Poisson processes*, volume 529. Springer, 2006.

[53] Levent Guen. Tandem queueing systems subject to blocking with phase type servers: Analytical solutions and approximations. Technical report, MARYLAND UNIV COLLEGE PARK SYSTEMS RESEARCH CENTER, 1986.

[54] Levent Guen and Armand M Makowski. An approximation method for general tandem queueing systems subject to blocking. Technical report, MARYLAND UNIV COLLEGE PARK SYSTEMS RESEARCH CENTER, 1987.

[55] Leo Eugene Hanifin. *Increased transfer line productivity utilizing systems simulation*. PhD thesis, University of Detroit, 1975.

[56] Brian J Haydon. *The behaviour of systems of finite queues.* PhD thesis, University of NSW, 1973.

[57] Armin Heindl. Decomposition of general tandem queueing networks with mmpp input. *Performance Evaluation*, 44(1):5–23, 2001.

[58] Kevin B Hendricks and John O McClain. The output process of serial production lines of general machines with finite buffers. *Management Science*, 39(10):1194–1201, 1993.

[59] Frederick S Hillier and Ronald W Boling. Finite queues in series with exponential or Erlang service times - a numerical approach. *Operations Research*, 15(2):286–303, 1967.

[60] Yosef Hochberg. A sharper bonferroni procedure for multiple tests of significance. *Biometrika*, 75(4):800–802, 1988.

[61] Yosef Hochberg and Ajit C Tamhane. Multiple comparison procedures. 2009.

[62] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[63] Gerhard Hommel. A stagewise rejective multiple test procedure based on a modified bonferroni test. *Biometrika*, 75(2):383–386, 1988.

[64] Rob Hyndman and Yeasmin Khandakar. Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software, Articles*, 27(3):1–22, 2008.

[65] Mohsen A Jafari and J George Shanthikumar. An approximate model of multistage automatic transfer lines with possible scrapping of workpieces. *IIE transactions*, 19(3):252–265, 1987.

[66] Raj Jain. Congestion control and traffic management in atm networks: Recent advances and a survey. *Computer Networks and ISDN systems*, 28(13):1723–1738, 1996.

[67] Frank P Kelly. *Reversibility and stochastic networks.* Cambridge University Press, 2011.

[68] Leonard Kleinrock. Queuing systems, vol. 1: Theory. *J. Wiley and Sons*, 1975.

[69] Steven M Klivansky, Amarnath Mukherjee, and Cheng Song. On long-range dependence in nsfnet traffic. Technical report, Georgia Institute of Technology, 1994.

[70] Allen Dixon Knott. *The efficiency of series production lines.* PhD thesis, University of New South Wales, 1967.

[71] Allen Dixon Knott. The inefficiency of a series of work stationsa simple formula. *The International Journal of Production Research*, 8(2):109–119, 1970.

[72] J Ladbrook. Breakdowns modellingan inquest. *Master in Philosophy, University of Birmingham*, 1998.

[73] Guy Latouche and Vaidyanathan Ramaswami. *Introduction to matrix analytic methods in stochastic modeling.* SIAM, 1999.

[74] Averill M Law, W David Kelton, and W David Kelton. *Simulation modeling and analysis*, volume 2. McGraw-Hill New York, 1991.

[75] Averill M Law and Michael G McComas. Simulation of manufacturing systems. In *Proceedings of the 19th conference on Winter simulation*, pages 631–643. ACM, 1987.

[76] Averill M Law and Michael G McComas. Secrets of successful simulation studies. In *Proceedings of the 23rd conference on Winter simulation*, pages 21–27. IEEE Computer Society, 1991.

[77] Averill M Law and Michael G McComas. Simulation of manufacturing systems. In *Proceedings of the 30th conference on Winter simulation*, pages 49–52. IEEE Computer Society Press, 1998.

[78] Will E Leland, Murad S Taqqu, Walter Willinger, and Daniel V Wilson. On the self-similar nature of ethernet traffic. In *ACM SIGCOMM Computer Communication Review*, volume 23, pages 183–193. ACM, 1993.

[79] Will E Leland, Murad S Taqqu, Walter Willinger, and Daniel V Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on networking*, 2(1):1–15, 1994.

[80] XG Liu and JA Buzacott. A zero-buffer equivalence technique for decomposing queueing networks with blocking. *Queueing Networks with Blocking*, pages 87–104, 1989.

[81] Miron Livny, Benjamin Melamed, and Athanassios K Tsiolis. The impact of autocorrelation on queuing systems. *Management science*, 39(3):322–339, 1993.

[82] Greta M Ljung and George EP Box. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, 1978.

[83] Lanting Lu. *Modelling breakdown durations in simulation models of engine assembly lines.* PhD thesis, University of Southampton, 2009.

[84] David M Lucantoni. New results on the single server queue with a batch markovian arrival process. *Communications in Statistics. Stochastic Models*, 7(1):1–46, 1991.

[85] David M Lucantoni, Kathleen S Meier-Hellstern, and Marcel F Neuts. A single-server queue with server vacations and a class of non-renewal arrival processes. *Advances in Applied Probability*, 22(3):676–705, 1990.

[86] James T Luxhoj and Huan-Jyh Shyur. Reliability curve fitting for aging helicopter components. *Reliability Engineering & System Safety*, 48(3):229–234, 1995.

[87] Benjamin Melamed and Jon R Hill. A survey of tes modeling applications. *Simulation*, 64(6):353–370, 1995.

[88] Kristof Mertens, Inge Vaesen, Jenny Löffel, Bart Kemps, Bram Kamers, Johan Zoons, Paul Darius, Eddy Decuypere, Josse De Baerdemaeker, and Bart De Ketelaere. An intelligent control chart for monitoring of autocorrelated egg production process data based on a synergistic control strategy. *Computers and Electronics in Agriculture*, 69(1):100–111, 2009.

[89] Eginhard J Muth. Numerical methods applicable to a production line with stochastic servers. *Algorithmic methods in probability*, 7:143–159, 1977.

[90] Eginhard J Muth. Stochastic processes and their network representations associated with a production line queuing model. *European Journal of Operational Research*, 15(1):63–83, 1984.

[91] Eginhard J Muth and Abdullah Alkaff. The throughput rate of three-station production lines: A unifying solution. *International Journal of Production Research*, 25(10):1405–1413, 1987.

[92] Marcel F Neuts. *Matrix-geometric solutions in stochastic models: an algorithmic approach.* Courier Corporation, 1981.

[93] Erland Hejn Nielsen. Autocorrelation in queuing network-type production systemsrevisited. *International Journal of Production Economics*, 110(1):138–146, 2007.

[94] Erland Hejn Nielsen. Autocorrelation in queuing network-type production systemsrevisited. *International Journal of Production Economics*, 110(1):138–146, 2007.

[95] Kentaro OKAZAKI. *Approximate Analysis of Tandem Blocking Queueing Networks with Correlated Arrivals and Services.* PhD thesis, Kyoto University, 2005.

[96] Raif O Onvural. *A survey of closed queueing networks with finite buffers.* Univ., 1987.

[97] Raif O Onvural and HG Perros. On equivalencies of blocking mechanisms in queueing networks with blocking. *Operations Research Letters*, 5(6):293–297, 1986.

[98] Pramod Pancha and Magda El Zarki. Mpeg coding for variable bit rate video transmission. *IEEE communications magazine*, 32(5):54–66, 1994.

[99] Vern Paxson and Sally Floyd. Wide area traffic: the failure of poisson modeling. *IEEE/ACM Transactions on Networking (ToN)*, 3(3):226–244, 1995.

[100] Diego Crespo Pereira, David del Rio Vilas, Nadia Rego Monteil, Rosa Rios Prado, and Alejandro Garcia del Valle. Autocorrelation effects in manufacturing systems performance: a simulation analysis. In *Proceedings of the Winter Simulation Conference*, page 123. Winter Simulation Conference, 2012.

[101] Diego Crespo Pereira, David del Rio Vilas, Nadia Rego Monteil, Rosa Rios Prado, and Alejandro Garcia del Valle. Autocorrelation effects in manufacturing systems performance: a simulation analysis. In *Proceedings of the Winter Simulation Conference*, page 123. Winter Simulation Conference, 2012.

[102] HG Perras. Open queueing networks with blocking.

[103] Harry G Perros. *Queueing networks with blocking*. Oxford University Press, Inc., 1994.

[104] Harry G. Perros and Tayfur Altiok. Approximate analysis of open networks of queues with blocking: Tandem configurations. *IEEE transactions on software engineering*, (3):450–461, 1986.

[105] HG Perros. A survey of two-node queueing networks with blocking. *Computer Science Report, North Carolina State University, Raleigh, NC*, 1988.

[106] Bernard Philippe, Youcef Saad, and William J Stewart. Numerical methods in markov chain modeling. *Operations research*, 40(6):1156–1179, 1992.

[107] Stephen M Pollock, John R Birge, and Jeffrey Morgan Alden. Approximation analysis for open tandem queues with blocking: Exponential and general service distribution. 1985.

[108] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.

[109] Nori Prakasa Rao. On the mean production rate of a two-stage production system of the tandem type. *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, 13(2):207–217, 1975.

[110] Nori Prakasa Rao. A generalization of the bowl phenomenonin series production systems. *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, 14(4):437–443, 1976.

[111] Nori Prakasa Rao. A viable alternative to the method of stages solution of series production systems with erlang service times. *The International Journal Of Production Research*, 14(6):699–702, 1976.

[112] Dror M Rom. A sequentially rejective test procedure based on a modified bonferroni inequality. *Biometrika*, 77(3):663–665, 1990.

[113] Tobias Rydén. Parameter estimation for markov modulated poisson processes. *Stochastic Models*, 10(4):795–829, 1994.

[114] Tobias Rydén. An em algorithm for estimation in markov-modulated poisson processes. *Computational Statistics & Data Analysis*, 21(4):431–447, 1996.

[115] Paulo Salvador, António Pacheco, and Rui Valadas. Modeling ip traffic: joint characterization of packet arrivals and packet sizes using bmaps. *Computer Networks*, 44(3):335–352, 2004.

[116] BLN Sastry. *Analysis of two-machine Markovian production lines and decomposition of longer lines.* PhD thesis, Ph. D. Thesis, Department of Mechanical Engineering, Indian Institute of Technology, Bombay, India, 1985.

[117] BLN Sastry and PG Awate. Analysis of a two-station flow line with machine processing subject to inspection and rework. *Opsearch*, 25:89–97, 1988.

[118] Alexander K Schömig and Manfred Mittler. Autocorrelation of cycle times in semiconductor manufacturing systems. In *Proceedings of the 27th conference on Winter simulation*, pages 865–872. IEEE Computer Society, 1995.

[119] Steven Lee Scott. *Bayesian methods and extensions for the two state Markov modulated Poisson process.* PhD thesis, Harvard University, 1998.

[120] Boris Alexandrovich Sevast'yanov. Influence of storage bin capacity on the average standstill time of a production line. *Theory of Probability & Its Applications*, 7(4):429–438, 1962.

[121] Juliet Popper Shaffer. Multiple hypothesis testing. *Annual review of psychology*, 46(1):561–584, 1995.

[122] J George Shanthikumar and Mohsen A Jafari. Bounding the performance of tandem queues with finite buffer spaces. *Annals of Operations Research*, 48(2):185–195, 1994.

[123] R John Simes. An improved bonferroni procedure for multiple tests of significance. *Biometrika*, 73(3):751–754, 1986.

[124] Timothy L Smunt and William C Perkins. Stochastic unpaced line design: review and further experimental results. *Journal of Operations Management*, 5(3):351–373, 1985.

[125] William J Stewart. A comparison of numerical techniques in markov modeling. *Communications of the ACM*, 21(2):144–152, 1978.

[126] William J Stewart. *Numerical solution of Markov chains*, volume 8. CRC press, 1991.

[127] K Takahashi and N Nakamura. The effect of autocorrelated demand in jit production systems. *International Journal of Production Research*, 36(5):1159–1176, 1998.

[128] Yutaka Takahashi, Hideo Miyahara, and Toshiharu Hasegawa. An approximation method for open restricted queueing networks. *Operations Research*, 28(3-part-i):594–602, 1980.

[129] Tobias Uhlig, Oliver Rose, and Sebastian Rank. Evaluation of modeling tools for autocorrelated input processes. In *Winter Simulation Conference (WSC), 2016*, pages 1048–1059. IEEE, 2016.

[130] Nico M Van Dijk and Bernard F Lamond. Simple bounds for finite single-server exponential tandem queues. *Operations research*, 36(3):470–477, 1988.

[131] Richard Paul Wiley. *Analysis of a tandem queue model of a transfer line.* PhD thesis, Massachusetts Institute of Technology, 1981.

[132] S Paul Wright. Adjusted p-values for simultaneous inference. *Biometrics*, pages 1005–1013, 1992.

[133] Diethelm Wuertz. Package fgarch. Technical report, 2013.

[134] Sencer Yeralan and Eginhard J Muth. A general model of a production line with intermediate buffer and station breakdown. *IIE transactions*, 19(2):130–139, 1987.

[135] Timothy M Young and Paul M Winistorfer. The effects of autocorrelation on real-time statistical process control with solutions for forest products manufacturers. *Forest Products Journal*, 51(11/12):70, 2001.

[136] Eric Zivot. *Practical Issues in the Analysis of Univariate GARCH Models*, pages 113–155. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[137] Walter Zucchini. An introduction to model selection. *Journal of Mathematical Psychology*, pages 41–61, 2000.