Simulation Assignment Report

Laurens van den Brink 4022831 and Nadia Boudewijn 3700607

March 26, 2014

Contents

1 Introduction

This is report accompanies a simulation model for a production line of DVD's. This simulation model is designed for the Simulation Assignment for the simulation course 2013-2014 at Utrecht University.

1.1 Problem description

We will be performing a simulation study for a production line of DVD's, hoping to find improvements for the line. At some points in the line there is limited buffer space and part of the production is batch processing. The producing company wants to find out the best buffer- and batch sizes to increase the dvd production. They are also interested in possible improvements of their production line, especially in alleviating the impact of bottlenecks.

To gain insight on the production line we questioned the domain expert Marjan van den Akker. She provided us with information about the production process. A schematic overview of the production line can be found on the next page.

1.2 Project goal

The goal is to perform a scientific sound simulation study in order to determine if it's possible to increase the DVD production productivity.

2 Problem Analysis

In this section we will discuss the methods we have used to reach the project goal, as well as some domain-specific issues to this simulation study that need to be adressed.

2.1 Assumptions

A model is supposed to be a simplification of reality, therefore in order to create our model we made some simplifying assumptions that we will discuss in this section.

2.1.1 General assumptions

- All machines have enough resources available, when they run out, recourses are being refilled immediately (which may cause the machine to stop it's production for some time during the refill).
- The production line runs 24/7.
- When we start the production line we assume none of the machines is broken down.
- At any moment in time enough repairman are available, so that repairs can be scheduled immediately.

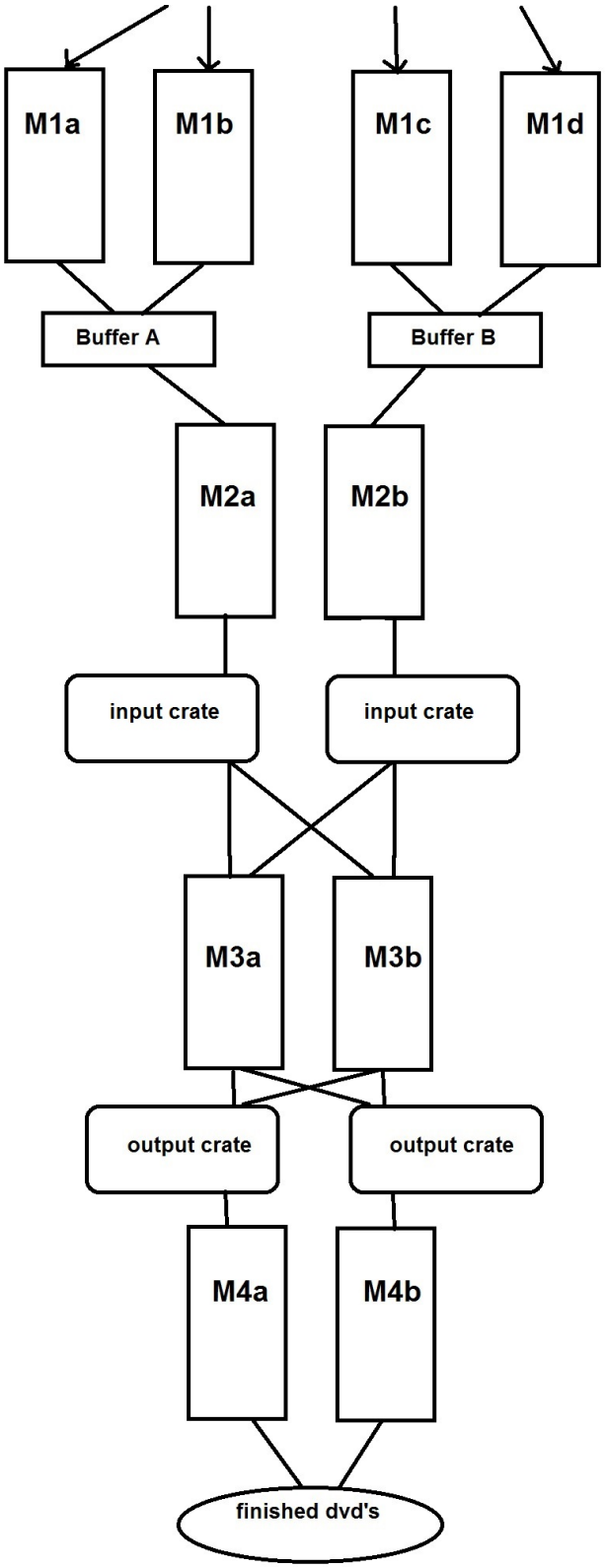


Figure 1: Figure 1: A schematic representation of the DVD production line.

2.1.2 Disitribution Assumptions

- Distributions for time to failure and time to repair are equal for machines of the same sort (machines with the same number in our case).
- The 500 observations for the processing time of M1/M2/M4 are not enough to run a reliable trace-driven simulation. Therefore we construct an emprical distribution for the processing times of M1/M2/M4 from these observations.

2.2 Model explanation

We will start with detailed descriptions of each subsystem in bullet format and how these subsystems interact. Next we will give an overview of the events for this model, and how they are handled. We will also discuss the model state and the performance measures, followed by limitations of the simulation model.

2.2.1 M1

Machine 1 - Injecton Molding

- There are 4 injection machines (M1a, M1b, M1c, M1d) that may break down individually.
- A machine will break down every 8 hours on average. We implement this with an exponential distribution on the advice of the expert.
- Repairing a machine takes 2 hours on average. We implement this with an exponential distribution on the advice of the expert.
- There is a buffer (B1) for 20 dvd's between M1a, M1b and M2a. Another buffer (B2) that can hold 20 dvd's is placed between M1c, M1d and M2b.

2.2.2 Buffer

- Buffer A and B can each store 20 DVD's
- Buffer A can be filled by M1a and M1b, and offers DVD's to M2a
- Buffer B can be filled by M1c and M1d, and offers DVD's to M2b

2.2.3 M2

Machine 2 - Dye coating and drying

- There are 2 machines, M2a and M2b.
- A machine ruins 2
- There is 5 minutes travel (drying time) between M2 and M3.

2.2.4 Crate

- There are 6 crates.
- Three of these crates can be filled by M2a and can go into the production of M3a. The other three crates can be filled by M2b and go into production of M3b.
- These crates are in te most ideal case: an input, an output and a processing crate. But it is possible for the 3 crates to be all input, or all output but only 1 crate at a time can be processed by a M3 machine.

2.2.5 M3

Machine 3 - Sputtering, lacquer coating and drying

- There are 2 machines, M3a and M3b.
- A machine only works on batches of size 20.
- A batch is stored in a crate
- 3 percent of de dvd's are in a delayed batch due to cleaning time of the sputtering part.

- On avergage it takes 5 minutes to clean the sputtering part. We implement this with an exponential distribution on the advice of the expert.
- If the sputtering part doens't have to be cleaned the sputtering takes 10 seconds on average per dvd. We implement this with an exponential distribution on the advice of the expert.
- Lacquer coating takes 6 seconds on average per dvd. We implement this with an exponentiall distribution on the advice of the expert.
- At the end the whole batch needs 3 minutes of drying before it's ready for output.

2.2.6 M4

Machine 4 - Printing and Finishing

- There are 2 machines, M4a and M4b.
- On average, after printing 200 dvd's the inkt will have to be refilled. We implement this with a normal distribution on the advice of the expert.
- Replacing the inkt takes 15 minutes on average, with a standard deviation of 1 minute.

2.2.7 Program Initialization

```
MAIN PROGRAM {
    while time < runlenght
    {
        advance simulation time;
        get next event from event list;
        update statistics + system state;
        generate future events and add them them to event list;
    }
}
```

2.2.8 Events and event handlers

We distinguish te following events (see event graph on the next page):

- M1Finished
- M2Finsihed
- AddDVDToCrate
- M3Finished
- M4Finished
- BreakdownM1
- RepairedM1
- BreakdownM3
- RepairedM3
- BreakdownM4
- RepairedM4

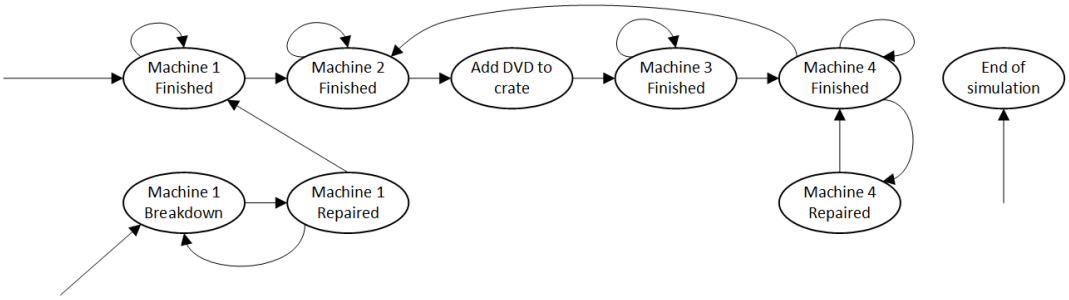


Figure 2: Event graph

We use the following **event handlers** and *subroutines* to deal with the events:

Machine 1 Finished:{
if (machine state = BROKEN) {
Store the time the machine should have finished
}
else if (machine state = WASBROKEN) {
the machine has been broken during producing the dvd
schedule a machine 1 finished event over the time the machine has been repaired
}
else if (machine = machine 1a or machine 1b) {
update bufferA
schedule the next machine 1 finish event to keep producing
if (machine 2a = IDLE) {
schedule machine 2a event
}
}**else** {
update bufferB
schedule the next machine 1 finish event
if (machine 2b = IDLE) {
schedule machine 2b event
}
}
}

Schedule Machine 1: {
the limit of the buffer is 19 when both machines are producing, otherwise 20
if (the output buffer is full) {
state of machine = BLOCKED;
}**else** {
update dvds in production
add machine 1 finished event to event list
}
}

Breakdown Machine 1:{
MachineState of the machine = BROKEN;
Time machine 1 has broken down = time
schedule repair machine 1: 2 hours exponentially distributed
}

Repair Machine 1: {
schedule breakdown machine 1
if(the machine did finish in the mean time) {
delay of machine = Time machine should have finished = time has broken down
machine state of machine = BUSY;
schedule machine 1 finished: over the time the machine has been delayed
}
else {
it is not known when the machine should have finished,
so when it happens machine 1 finished event reschedules the event
}
}

Schedule Machine 1 breakdown: {
add breakdown machine 1 event to event list in exp(8 hours)
}

Schedule Machine 1 repair: {
add repaired machine 1 event to event list in exp(2 hours)
}

Machine 2 Finished: {
schedule AddDVDtoCrate event
schedule next machine 2 finished event check if machine was waiting and if so, schedule Machine 1a and 1b or machine 1c and 1d finish event
(depending on which machine 2 was finished)
}

Schedule Machine 2: {
check if there is room left within the crates to be filled by m3
if(dvdReadyforM3 <= crateSize) {
if(buffer > 0) {
update buffer − = 1
add machine 2 finished event to the event list
}**else** {
no input for the machine
machine state = IDLE;
}
}**else** {
the machine will not be able to output the next dvd
machine state = BLOCKED;
}
}

AddDVDtoCrate: {
update dvdReadyforM3
if(a crate is full and m2 is IDLE) {
schedule machine 3 finished event
}
}

Schedule AddDVDtoCrate: {
add AddToCrate event to event list in 5 minutes
}

Machine 3 Finished: {
    update dvd ready for machine 4 with 20
    schedule machine 3 finished event
    if machine 4 is IDLE {
        schedule machine 4
    }
}

Schedule Machine 3: {
    if(a full crate is available for input machine 3) {
        start producing this crate
        crates to be filled by machine 3 = − 1
        dvd ready for machine 3 - CrateSize
        add machine 3 finished event to event list in exp(10) + exp(6) + 3 ∗ 60
    }**else** {
        output crate and go back to waiting for input
        machine state = IDLE;
    }
}

Machine 4 Finished: {
    update statistics: dvd in production, dvd produced with 98%, dvd failed with 2% chance
    schedule machine 4 finished event
    if (the machine emptied a whole crate) {
        crates to be filled by machine 3 + = 1
        if(machine 2 was blocked by no crate available) {
            lift the blockade, there is an emtpy crate available again
        }
    }
}

Schedule Machine 4: {
    if(machine should be serviced) {
        machine state = BROKEN;
        schedule machine 4 repair event
    }**else if**(dvd ready for input macchine 4 > 0) {
        DvdReadyforInputM4 = − 1
        add machine 4 finished event to event list
    }**else** {
        machine state = IDLE
    }
}

Repair Machine 4: {
    reset the dvd counter when service is needed
    machine state of machine 4 = BUSY;
    schedule machine 4 finished event
}

Schedule Machine 4 Repair: {
    add repaired machine 4 event to event list in exp(15 min)
}

2.2.9 Performance measures

We used the following performance measures:

- Production per hour: before the simulation study the production was 175 dvd's per hour. We hope to raise this to 200 dvd's per hour.
- Throughput time of products

2.2.10 State

To keep track of the model state we measure the following:

- Number of dvd's in production.
- State of every machine (IDLE / BUSY / BROKEN / WASBROKEN)
- Buffercontent
- cratesReadyforInputM3: the amount of crates ready for input M3
- cratesReadyforInputM4: the number of dvd's that are ready for input M4

2.2.11 Limitations

3 Experiments

Met deze dingen kunnen we rekening houden: Candidates for sensitivity analysis:

- the value of a parameter - if it's not sensitive for a specific range of values a better specification is needed
- the choice of a distribution
- the entity moving thourgh the simulated system - following 1 dvd or a whole batch may produce the same results for the perfomance measures of interest while reducing execution time
- the level of detail for a subsystem
- what data are the most crucial to collect

When one is performing a sensitivity analysis, it is important to use the method of common random numbers to control the randomness in the simulation. Otherwise, the effect of changing one factor may be cofounded with other changes (e.g. different random values from some input distribution) that inadvertently occur.

3.1 Set up

Summaries of a data set such as its sample mean and a histogram. Details are for the appendix to retain readability. If one has fitted a theoretical probability distribution to a set of observed data, then the adequacy of the representation can be assessed by using the graphical plots and goodness-of-fit tests.

3.2 Results

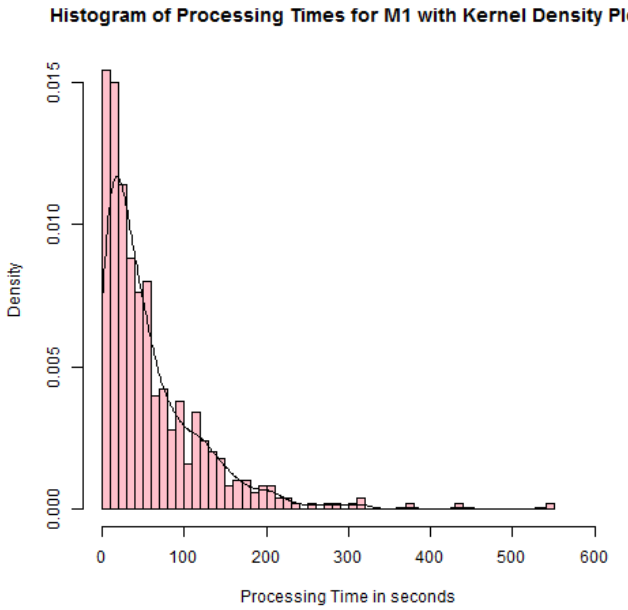
3.3 Input Analysis

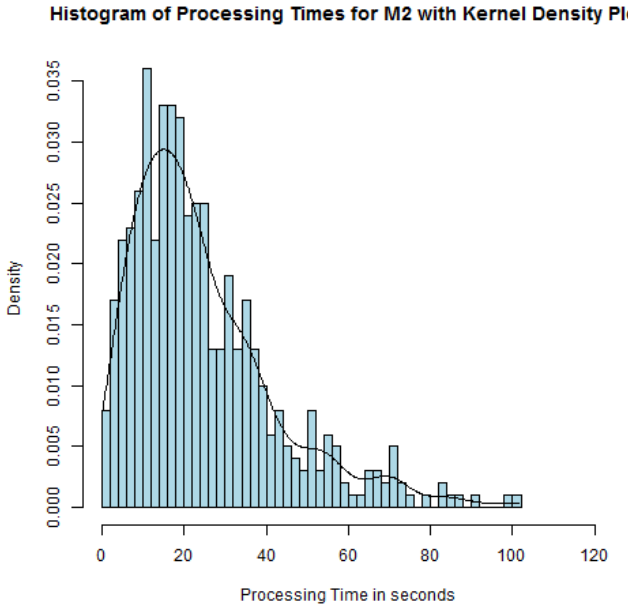
Geef een histogram van de frequentieverdeling van de processing tijden.

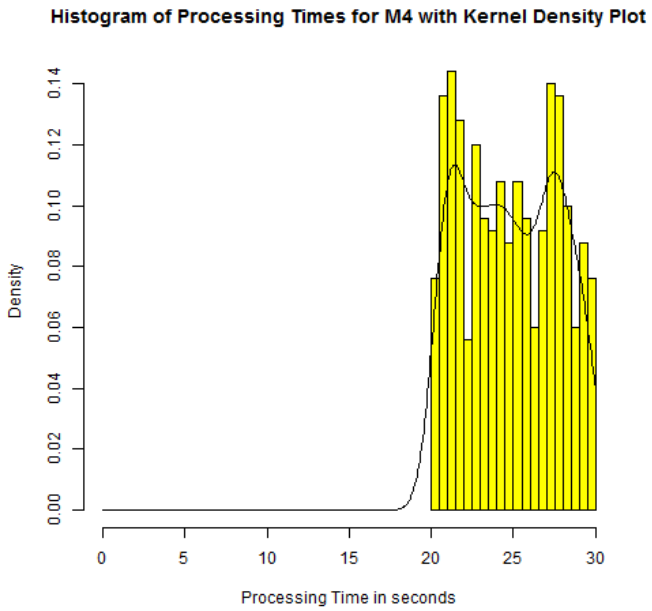
The goodness of fit of a statistical model describes how well it fits a set of observations.
Hypothesis H0: the observations X1,X2,,Xn follow distribution F
Do we accept or reject this hypothesis?

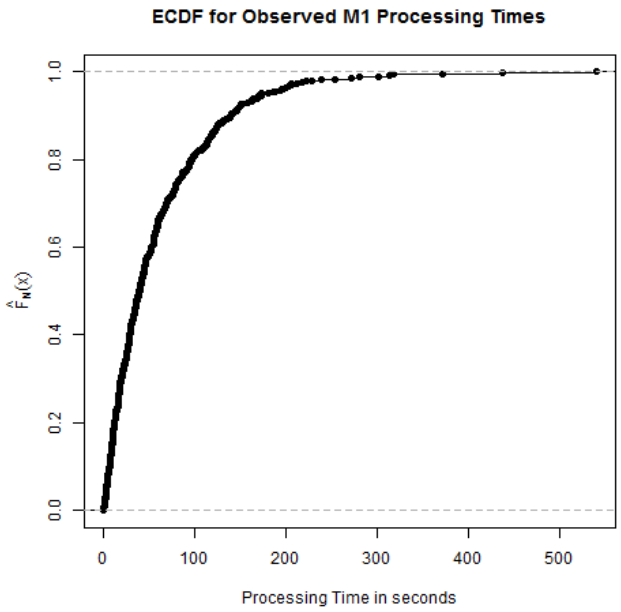
- Chi-squared test
- Kolmogorov-Smirnov test

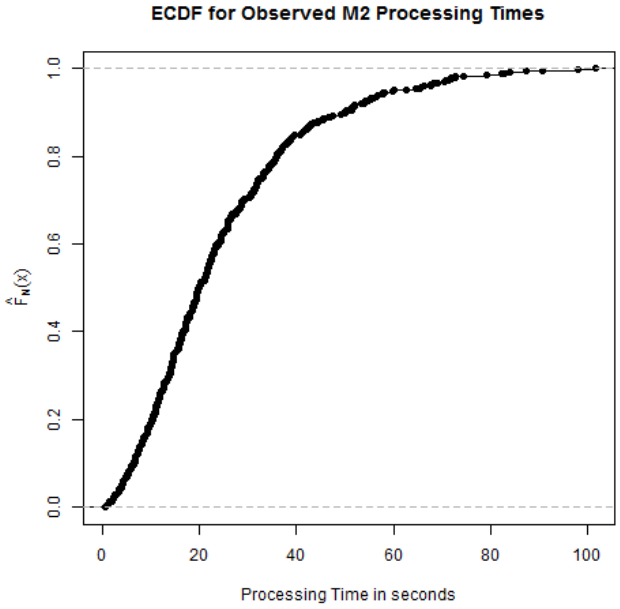
Can be performed with R

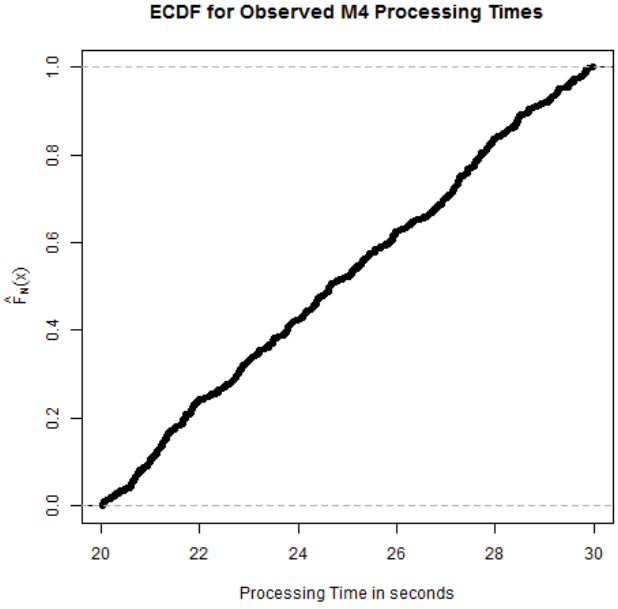

Histogram of Processing Times for M1 with Kernel Density Plot
Density
0.000
0.005
0.010
0.015
0
100
200
300
400
500
600
Processing Time in seconds


Histogram of Processing Times for M2 with Kernel Density Plot
Density
0.000
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0
20
40
60
80
100
120
Processing Time in seconds


Histogram of Processing Times for M4 with Kernel Density Plot
Density
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0
5
10
15
20
25
30
Processing Time in seconds







3.3.1 Statistical Hypothesis testing

1. Formulate hypothesis Ho
2. Choose type of test
3. Determine significance a and decision rule
4. compute test statistics and take decision

3.3.2 Q-Q Plot

3.4 Validation

Establish that the output data closely resembles the output data that would be expected from the actual system.

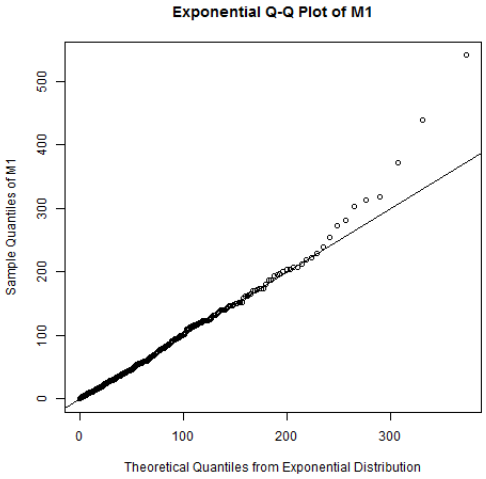
Establish the ability to predict future behavior. Use one data set for calibration and another independent set for validation.

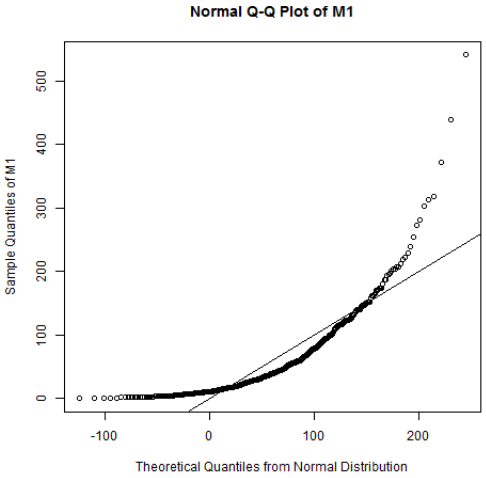
4 Conclusions

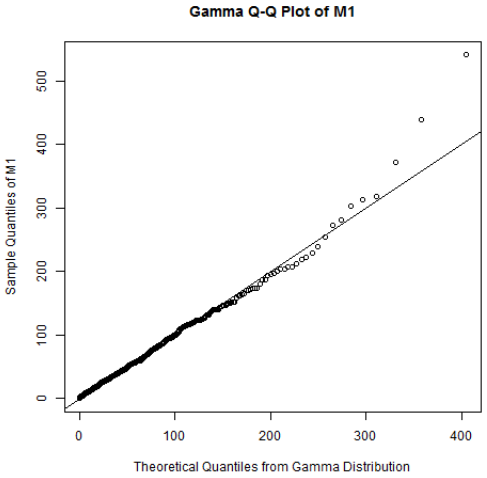
5 Appendix

5.1 Interview notes

5.2 Detailed statistical analyses


Exponential Q-Q Plot of M1
Sample Quantiles of M1
Theoretical Quantiles from Exponential Distribution
0
100
200
300
400
500


Normal Q-Q Plot of M1
Sample Quantiles of M1
Theoretical Quantiles from Normal Distribution
-100
0
100
200
0
100
200
300
400
500


Gamma Q-Q Plot of M1
Sample Quantiles of M1
Theoretical Quantiles from Gamma Distribution
0
100
200
300
400
0
100
200
300
400
500