Project 2

Financial Time Series

Patrik.Dennis[a]

[a] *Gothenburg University & Chalmers University of Technology*

April, 2022

Contents

1. Excercise 1.1

1.1. Problem Description

The task for the current exercises is to fit an ARMA(6,5) -model, with mean 0, to the time series data *denmark_covid_logrets.csv*. Furthermore, the residuals of the model should be stored for further analysis and fitting in upcoming sections.

1.2. Solution

To fit an ARMA(6,5) model with mean 0 in R the provided in Appendix A. Furthermore, the residuals are provided by the model fitting, which is thereafter stored for later use.

2. Exercise 1.2

2.1. Problem Description

The task is to use the residuals stored and obtained in Exercise 1.1, to plot the sample ACF and PACF of the residuals, for lags $h = 0, ..., 20$. Furthermore, the ACF of the squared residuals are to be plotted. After plotting the stated functions, the task is to draw the conclusion on whether the log returns can be modeled as white noise. Additionally, further investigation whether there is any indication for a Garch model to be provided.

2.2. Solution

The obtained plots for the ACF and PACF regarding the residuals can be seen in Figure.[1]. From (a), one can see that the sample acf values lie within the confidence interval. This implies that the log returns can be modeled as white noise.

Furthermore, from (c), the plot is showing serial correlation for the variance of the log-returns. Serial correlation regarding the variance of the returns, implies that the volatility can be foretasted by historical values. Therefore, a Garch model is deemed suitable to model the volatility of the time series. Additionaly, from (b), there appears to be significance for all lags up to 3. This could provide an argument for providing the Arima(3,0) model.

3. Exercise 1.3

3.1. Problem Description

First the task is to fit a GARCH(p,q) model for all $p, q \in (1, ..., 10)$ to the data, to the residuals obtained in Excercise 1. After fitting the GARCH(p,q) models, the objective is to find the best fitted model, by observing and minimizing the BIC information criteria

BIC Information Criteria 3-1

$$BIC := -2 \cdot LogL + M \cdot \log(n). \qquad (1)$$

The BIC information criteria utilizes the parameters stated below,

M : Number of parameters estimated.

n : sample size.

LogL : Log-likelihood of the fit.

3.2. Solution

To see how the achievement of fitting and minimizing the BIC was performed, the reader is directed to Appendix A for Excercise 3. The resulting model that minimizes the BIC, was found to be GARCH(2,3).

3.3. Further Details

To obtain the minimized model fit, an inner- and outer for-loop was made where to traverse the p- and q values. Whilst traversing, the parameters p and q are fitted to the data. Thereafter, using the fact that $p + q + 2 = M$, the BIC score can be computed and stored in a matrix. The matrix storing the BIC-scores has row index for the p values and columnn indices for the q values, i.e., the GARCH(1,1) BIC-score is stored in m[1,1].

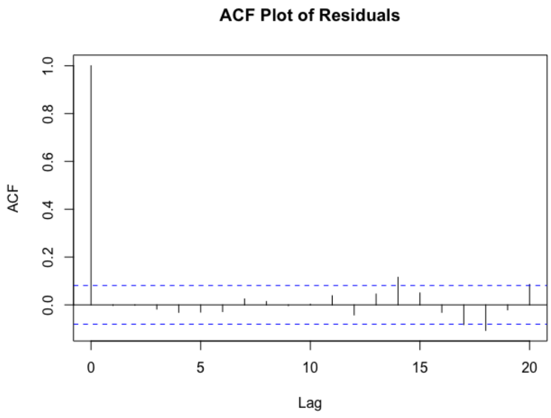
4. Exercise 4

4.1. Problem Description

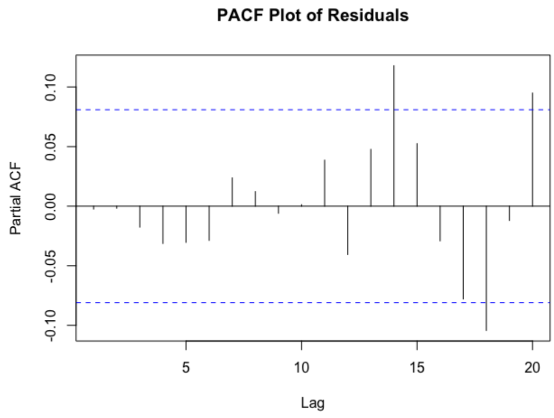
Now with the obtained model with the minimized BIC-score, one can perform a diagnostic check. The diagnostic check that will be performed in the report, is to compute and analyze the residuals. To test the validity of the model, the residuals should approach the normal distribution. Furthermore, the temporal covariance structure is investigated to test the validity of the obtained model. Additionally, the Autocorrelation plot of the residuals and squared residuals are provided to affirm the conclusions made.

4.2. Solution

Provided with the obtained GARCH(2,3)-model, one can perform the Shapiro Wilk test to test whether the residuals conform to a normal distribution.



(a) ACF Residuals



(b) PACF Residuals

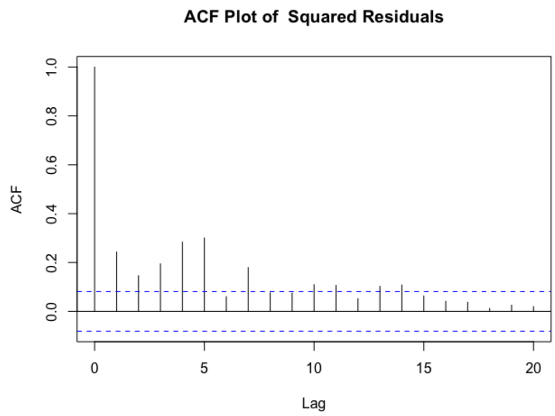


Figure 1: (c) PACF Residuals Squared

Shapiro Wilk Test **4-1**

$$W = \frac{(\sum_i^n a_i x_{(i)})^2}{\sum_i^n (x_i - \bar{x})^2} \quad (2)$$

The null hypothesis for the above is that the data is normally distributed. Performing the test on the residuals obtained by the GARCH(2,3)-model provides a p value, 2.2e-16. Therefore, the null hypothesis is rejected, thus implying that the residuals are not normally distributed. However, when looking at the kurtosis of the residuals, one finds that the kurtosis is 5.783262. With such a value, the distribution follows a Leptokurtic distribution, which describes a more heavy tailed distribution than the normal with kurtosis 3.

When observing the Autocorrelation plots in Figure.[3], one can see that there is no serial correlation for both the residuals and squared residuals. This makes reasonable argumentation that the residuals of the chosen model are IID. This is desirable. Furthermore, from the the plot (b), there seems to exist a serial correlation. This is not desirable, as they are therefore not IID.

Furthermore, the temporal conditional variance should capture the volatility of the time-series. The conditional variance of the chosen GARCH-model and how the model captures the volatility can be seen below.

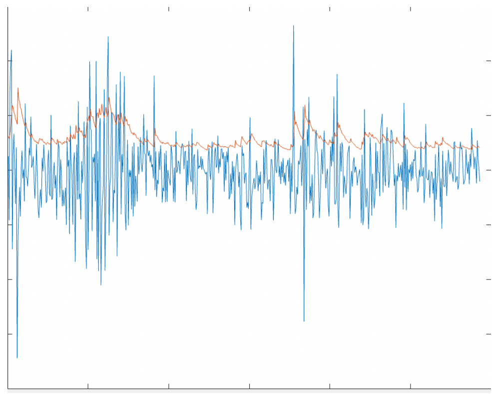


Figure 2: Plot displaying the conformity to the volatility of the returns.

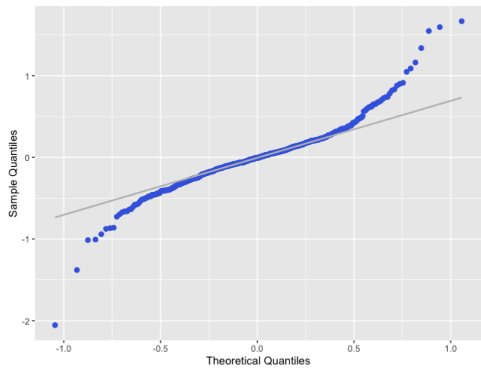


Figure 4: Plot displaying the qq-plot for the residuals obtained from the GARCH(2,3)-model.

As seen above a normal distribution conditional model may not be the best fit for the residuals obtained from the ARMA(6,5)-model. This will be further investigated in the next task.

5. Exercise 5

5.1. Problem Description

In this task further investigation regarding the residuals obtained from the fitted GARCH-model, on the residuals of the ARMA(5,6) model. The investigation that will take place in this task, is whether the normal distribution conditional assumption for the residuals is appropriate. Alternatives to the normal distribution fit on the noise is the t-distribution, that fits better for heavier tailed residuals.

5.2. Solution

To investigate the stated task, the qq-plot of the obtained residuals are shown in Figure.[4]. Observing Figure.[4], one can see that the distribution of the residuals are heavy tailed. This was also argued in the previous exercise, where the kurtosis was computed. The reason for concluding that the distribution is heavy-tailed by the qq-plot is due to the deviation in the former and latter stages of the plot. This shows lack of conformity to the normal distribution qq-plot, seen as the linear line. Heavy-tailed distribution further implies a larger distribution of outliers.

6. Excercise 6

6.1. Problem Description

Provided with the previous investigations on the model, a t-distribution may fit better to the noise at hand. A fit with the t-distribution instead of the normal distribution is to performed in Exercise 6. Furthermore, the same diagnostics as in exercise are to be performed.

6.2. Solution

Provided with the fit, seen in (Appendix A), diagnostics are to be performed to test whether the fit is more optimal than the previous fitted model. From the distribution of the quantile-plot, showing again a heavy-tailed distribution, one can conclude that the t-distribution is as better fit.

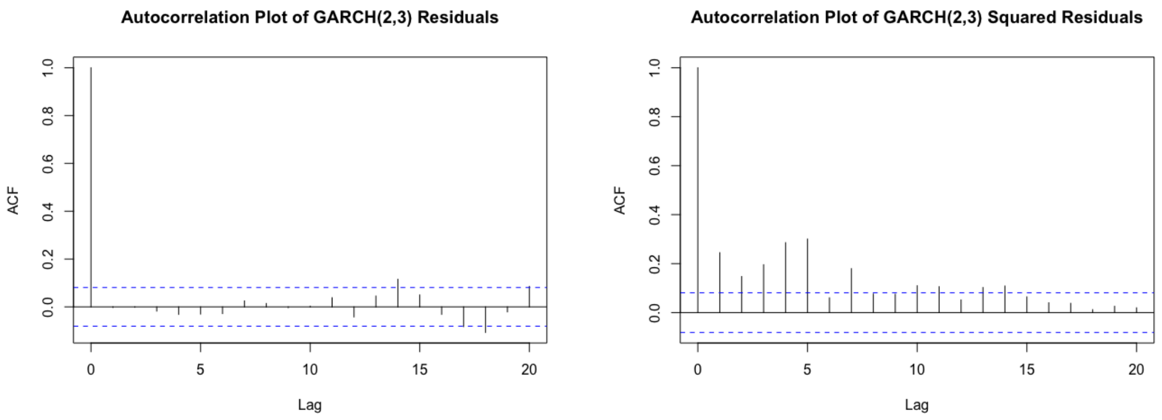
7. Excercise 2.1

7.1. Problem Description

In this task 1-step ahead-predictions are to be made via parametric bootstrapping. Note that the forecasting performed in this report is the iterative approach to h-step ahead predictions, not e.g. recursive. After obtaining the forecasts, the 2.5% and 97.5% quantiles are to be obtained and plotted for with each respective forecast.

7.2. Solution

To achieve 1-step-ahead predictions one must follow the four steps,



(a) Autocorrelation Plot of GARCH(2,3) Residuals

(b) Autocorrelation Plot of GARCH(2,3) Squared Residuals

Figure 3: Plots displaying the Autocorrelation plots for the GARCH(2,3) residuals and squared residuals.

I : Generate a sample of driving noise $Z_i = \mathcal{N}(0, \sigma = \sqrt{4.718})$.

II : Compute $\widehat{S}_{i+1}$ using $S_{i+1} = \phi S_i + Z_i$.

III : Repeat (I) and (II) 1000 times to get 1000 realizations $\widehat{S}_i^{(k)}, k = 1, ..., 1000$.

IV : Calculate the forecast of S_{i+1} by the sample average $S_i(1) := \frac{1}{K-1}\sum_{k=1}^{K}\widehat{S}_k$.

The implementation of the above algorithm for 1-step-ahead can be seen in (Appendix A). For this report, generalized functions have been manufactured. Such functions are `getQuantiles` and `getAllPredictions` and so forth and so on. The obtained quantiles and predictions are presented in Figure.[5] below.

8. 2.2.a

8.1. Problem Description

Provided with the foretasted values from the previous exercises, the measurements mean-squared- and directional error, are utilized in order to perform analysis of the forecasting results.

8.2. Solution

Using the mean-squared-error (MSE) measurement

MSE 8-1

$$\frac{1}{N-h}\sum_{j=0}^{N-h-1}(S_{1+h+j} - S_{1+j}(h))^2 \quad (3)$$

$$h = 1 \quad (4)$$

one gets the mse 5.144428. This seems plausible, due to the high variance in the quantiles.

9. Excercise 2.2.b

9.1. Problem Description

In this exercise, further analysis of the forecasts are to be performed via the direction measurement error. The directional measurement is based on the direction of movement. Such a measurement utilizes the test below.

Pearson's chi-square statistic 9-1

$$R := \sum_{i,j=1}^{2}\frac{(m_{ij} - \frac{m_{i0}m_{0j}}{m})^2}{\frac{m_{i0}m_{0j}}{m}} \quad (5)$$

9.2. Solution

The Pearson chi-square statistic tests and therefore has the null hypothesis, whether the categorical data provided is dependent or independent of each other. This meaning that the task is to test for whether the up and down movements are dependent or independent for the contingency table provided below.

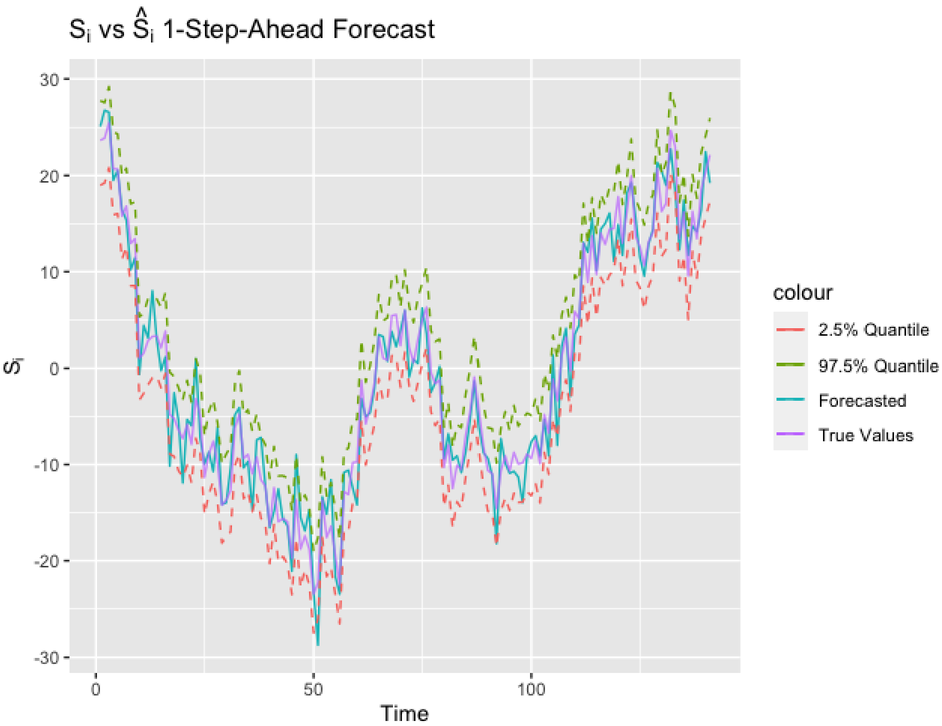


Figure 5: Plot showing the predicted values (Blue transparent line), true values (Purple line), 97.5% quantile (Dashed green line), 2.5% (Dashed red line).

	Predictions		Total
Actual	up	down	
up	56	18	74
down	13	53	66
Total	69	71	140

Table 1: 2x2 Contingency Table (confusion matrix) for hit and misses regarding the directions of the true and predicted series.

Provided with Table.[1], one can use Pearson's chi-sqaure statistic to test for dependency, resulting the table below.

χ	df	p value
41.525	1, p-value	1.164e-10

Table 2: Pearson chi-squared statistic results.

With such a result, one can therefore not reject the null hypothesis. This coincides with the computed MSE, due to the forecasts and therefore the MSE being independent to the true values.

10. Exercise 2.3

10.1. Problem Description

Provided with the framework used to compute 1-step-ahead forecasts, 10-step-ahead forecasts will now be computed and analyzed.

10.2. Solution

The code to produce Figure.[6], can be seen in Appendix.[Appendix A]. From Figure.[6], one can see how the movements regarding the predictions are more erratic than the one-step forecasts. This is understandable, due multi-step forecasts using estimated, i.e. predicted values, to reach and predict the tenth step value. This meaning that the further one predicts step-wise, the deviation to the actual values will increase. Furthermore, from the shown figure, one can see how the behavior of the movement is still showing in the predicted values. However, there seems to be a lag regarding the movement, in the predicted values path, compared to the true values movement.

The resulting MSE for the 10-step-ahead prediction was 75.87852. This is considerably larger than the 1-step-ahead forecasts, which is understandable. This being due to, as stated previously, to loss of information for each forecast value used to predict the tenth value. Furthermore, to clarify, as the length of the forecast values will be ten less than the true values, modifications to the upper limit of the sum was carried out. Additional, modifications due to the different length of the data can be further seen in (Appendix A).

One can further investigate the performance of the 10-step-ahead forecast by the directional measurement. The resulting contingency table can be seen in Table.[3], providing the Chi-Square statistic result provided in Table.[4]. From the results provided by the table below, one can see how a multi-step forecast deteriorates the quality of the forecasts, regarding error metrics. This provides that one should take caution when relying on forecasts.

	Predictions		Total
Actual	up	down	
up	35	32	67
down	26	40	66
Total	69	71	140

Table 3: 2x2 Contingency Table (confusion matrix) for hit and misses regarding the directions of the true and predicted series.

χ	df	p value
129	1	¡ 2.2e-16

Table 4: Pearson chi-squared statistic results.

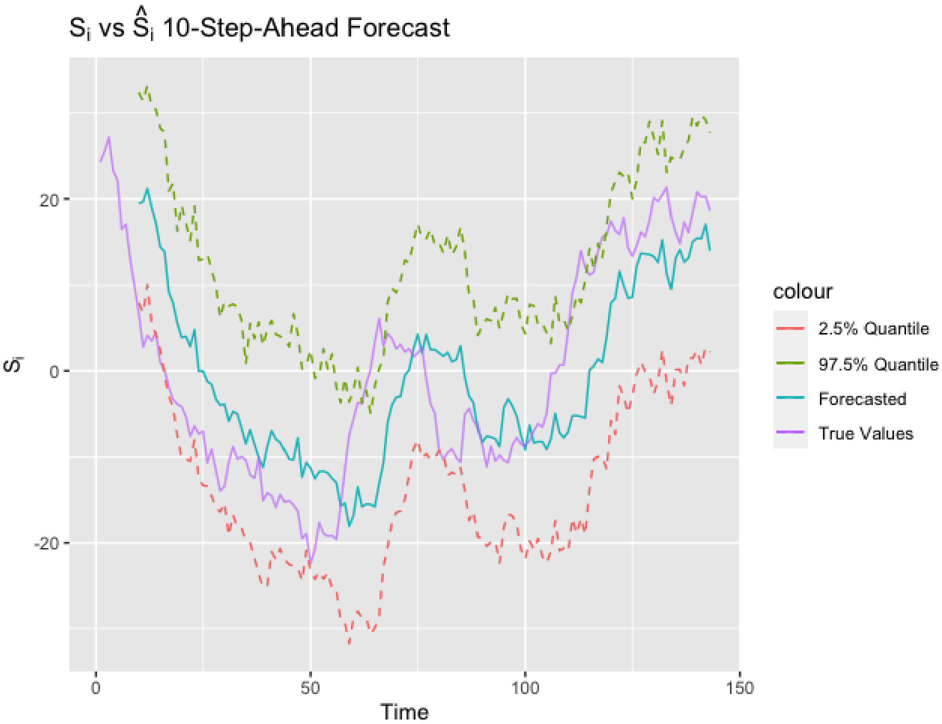


Figure 6: Plot showing the predicted values (Blue transparent line), true values (Purple line), 97.5% quantile (Dashed green line), 2.5% (Dashed red line).

Appendix A. Code

Appendix A.1. Excercise 1.1

```
df <- read_csv("denmark_covid_logrets.csv")

model <- arima(df$x, order=c(6,0,5), include.mean = FALSE)

model.resid <- resid(model)
residuals.2 <- model.resid^2
```

Appendix A.2. Excercise 1.2

```
#2

acf(model.resid, lag=20)
acf(model.resid, lag = 20, type=c('partial'))

Box.test(model.resid, lag = 20, type=c('Ljung-Box'))
```

Appendix A.3. Excercise 1.3

```
#3

m <- matrix(NULL*10,10,10)

for (i in 1:10){
  for (j in 1:10){

    GFSpec=ugarchspec(variance.model = list(model="sGARCH",garchOrder=c(i,j)),
                      mean.model = list(armaOrder=c(0,0),include.mean=TRUE))

    GF=ugarchfit(spec=GFSpec, data = model.resid,
                 solver="gosolnp",solver.control = list(tol=1e-7))


    BIC <- -2*GF@fit$LLH + (i+j+2*log(nrow(df)))
    m[i,j] <- BIC

  }
}

c <-  which(m == min(m), arr.ind = TRUE)
m[c] #best is therefore garch(2,3)
```

Appendix A.4. Excercise 1.4

```r
# 4

#including the mean is the offset term that should be included
GFSpec=ugarchspec(variance.model = list(model="sGARCH",garchOrder=c(2,3)),
                  mean.model = list(armaOrder=c(0,0),include.mean=TRUE),
                     distribution.model = "norm")

GF=ugarchfit(spec=GFSpec, data = model.resid,
             solver="gosolnp",solver.control = list(tol=1e-7))
resids <- GF@fit$residuals
resids2 <- (GF@fit$residuals)^2

kurtosis(resids)


shapiro.test(resids)

Box.test(resids2, type = c('Ljung'))
acf(resids, lag = 20, type = c('correlation'),
   main="Autocorrelation Plot of GARCH(2,3) Residuals")
acf(resids2, lag = 20, type = c('correlation'),
   main="Autocorrelation Plot of GARCH(2,3) Squared Residuals")

pacf(resids, lag = 20)
pacf(resids2, lag = 20)


kurtosis(resids)


shapiro.test(resids)

Box.test(resids2, type = c('Ljung'))
acf(resids, lag = 20, type = c('correlation'),
   main="Autocorrelation Plot of GARCH(2,3) Residuals")
acf(resids2, lag = 20, type = c('correlation'),
   main="Autocorrelation Plot of GARCH(2,3) Squared Residuals")


data.frame(resids) %>% ggplot(mapping = aes(sample = resids)) +
  stat_qq_point(size = 2, colour="Royalblue")  +
  stat_qq_line(colour="grey") +
  xlab("Theoretical Quantiles") +
  ylab("Sample Quantiles")
```

Appendix A.5. Excercise 1.5

```r
library('ggplot2')

resids.table %>% ggplot(mapping = aes(sample = resids2)) +
  stat_qq_point(size = 2, colour="Royalblue")  +
  stat_qq_line(colour="grey") +
  xlab("Theoretical Quantiles") +
  ylab("Sample Quantiles")
```

Appendix A.6. Excercise 1.6

```
# 6

GFSpec=ugarchspec(variance.model = list(model="sGARCH",garchOrder=c(2,3)),
                  mean.model = list(armaOrder=c(0,0),include.mean=TRUE),
                     distribution.model = "std")

GF=ugarchfit(spec=GFSpec, data = model.resid,
             solver="gosolnp",solver.control = list(tol=1e-7))
resids <- GF@fit$residuals

resids2 <- resid(GF@fit)
acf(resids, lag = 20, type = c('correlation'))
acf(resids2, lag = 20, type = c('correlation'))

pacf(resids, lag = 20)

library('ggplot2')

data.frame(resids) %>% ggplot(mapping = aes(sample = resids)) +
  stat_qq_point(size = 2, colour="Royalblue")  +
  stat_qq_line(colour="grey") +
  xlab("Theoretical Quantiles") +
  ylab("Sample Quantiles")
```

Appendix A.7. Excercise 2.1

```
#assumptions
psi <- 0.9827
sigma2 <- 4.718

#1

testdata <- read_csv("test_data.csv")
St <- colnames(testdata)
St <- as.numeric(St)
St <- append(St, testdata$`24.2876668779738`)
testdata <- as.data.frame(St)

lag.s <- lag(testdata$St, 1)
testdata['S_(t-1)'] <- lag.s

#Parametric Bootstrapping
S.sim <- testdata[2:142,]
for (n in seq(1,1000)){
  Z <- rnorm(141, mean = 0, sd = sigma2)
  S.sim[n] <- psi* S.sim$`S_(t-1)` + Z
}


forecastAverage <- function(i,df,predsstart){
  lst <- NULL
  #df <- as.data.frame(df)
  for (j in seq(predsstart,ncol(df))){
```

```r
    lst <- append(lst, df[i,predsstart])
  }
  return(mean(lst))
}

averages <- NULL

for(i in seq(1,141)){
  averages <- append(averages, forecastAverage(i,S.sim, predsstart = 3))
}

averages <- data.frame(averages)

averages['S_(t-1)'] <- S.sim$`S_(t-1)`


getAllPredictions <- function(df,i,predsstart){
  lst <- NULL
  for (j in seq(predsstart, ncol(df))){
    lst <- append(lst, df[i,j])
  }
  return(lst)
}

table.of.predictions <- getAllPredictions(S.sim,1,predsstart = 3)
table.of.predictions <- data.frame(table.of.predictions)
for (i in seq(2,141)){
  table.of.predictions[glue("{i}")] <- getAllPredictions(S.sim, i, predsstart = 3)
}


getQuantiles <- function(lst.of.quantiles,df,i){
  lst <- NULL
  for (quant in lst.of.quantiles){
    # this is wrong, getting multiple quantiles
    lst <- append(lst,quantile(df[,i], quant))
  }
  return(lst)
}

lst <- NULL
for (i in seq(1,ncol(table.of.predictions))){
  lst <- append(lst, getQuantiles(c(0.025, 0.975), table.of.predictions, i))
}

quantile.25 <- NULL
for (i in seq(1,282,2)){
  quantile.25 <- append(quantile.25,lst[i])
}

quantile.75 <- NULL
for (i in seq(2,282,2)){
  quantile.75 <- append(quantile.75,lst[i])
}

averages %>% ggplot(aes(x = 1:141, y = averages)) +
```

```r
  geom_line(aes(colour="Forecasted")) +
  geom_line(aes(x = 1:141, y =`S_(t-1)`, colour = "True Values"), alpha = 0.7) +
  geom_line(aes(x= 1:141, y = quantile.25 , colour = "2.5% Quantile"),
            size = 0.5,
            linetype = "dashed") +
  geom_line(aes(x = 1:141, y = quantile.75, colour = "97.5% Quantile"),
            size = 0.5,
            linetype = "dashed") +
 # geom_point(aes(x = 1:141, y = quantile.25, colour = "2.5% Quantile")) +
 # geom_point(aes(x = 1:141, y = quantile.75, colour = "97.5% Quantile")) +
  labs(x = "Time", y = TeX("$S_i$")) +
  ggtitle(TeX("$S_i$ vs $\\hat{S}_{i}$ 1-Step-Ahead Forecast"))
```

Appendix A.8. Excercise 2.2

```r
# 2. a

mse <-(1/(length(testdata-1)))*sum((averages$`S_(t-1)` - averages$averages)^2)


getDifferential <- function(vector1, i){
  difference <- vector1[i] - vector1[i-1]
  if(difference<0){
    return('-')
  } else if (difference == 0){
    return('0')
  } else {
    return('+')
  }
}



M11 <- function(predicted,true){
  s <- 0
  for(i in seq(2,length(predicted))){
    if((getDifferential(predicted, i) == "+") & (getDifferential(true,i) == "+")){
      s <- s + 1
    }
  }
  return(s)
}

M21 <- function(predicted, true){
  s <- 0
  for(i in seq(2,length(predicted))){
    if((getDifferential(predicted,i)=="+") & (getDifferential(true,i) == "-")){
      s <- s + 1
    }
  }
  return(s)
}

M12 <- function(predicted, true){
  s <- 0
```

```
  for(i in seq(2,length(predicted))){
    if((getDifferential(predicted,i)=="-") & (getDifferential(true,i) == "+")){
      s <- s + 1
    }
  }
  return(s)
}

M22 <- function(predicted, true){
  s <- 0
  for(i in seq(2,length(predicted))){
    if((getDifferential(predicted,i)=="-") & (getDifferential(true,i) == "-")){
      s <- s + 1
    }
  }
  return(s)
}

m11 <- M11(averages$averages, averages$`S_(t-1)`)
m21 <- M21(averages$averages, averages$`S_(t-1)`)
m12 <- M12(averages$averages, averages$`S_(t-1)`)
m22 <- M22(averages$averages, averages$`S_(t-1)`)

m20 <- m21 + m22
m10 <- m11 + m12
m01 <- m11 + m21
m02 <- m12 + m22

contigency.table <- data.frame(cbind(c(m11,m21),c(m12,m22)))
contigency.table <- contigency.table %>% rename('Up'='X1', 'Down'='X2')


createDummyvariables <- function(vector){
  lst <- NULL
  for (i in seq(2,length(vector))){
    if(getDifferential(vector,i)=="+"){
      lst <- append(lst,1)
    } else {
      lst <- append(lst,0)
    }
  }
  return(lst)
}

predictions <- createDummyvariables(averages$averages)
true <- createDummyvariables(averages$`S_(t-1)`)

chisq.test(predictions, true)
```

Appendix A.9. Excercise 2.3

```
multistep <- function(h,series,start, psi, sigma2){
  s <- (psi)*series[start-9] + rnorm(1,mean=0, sd = sqrt(sigma2))
  for (j in seq(0,h)){
    s <- (psi)*s + rnorm(1,mean = 0, sd = sqrt(sigma2))
```

```
  }
  return(s)
}


df.series <- testdata
series <- testdata$St


nstepaheadpreds <- data.frame(rnorm(134,0,0))
for (n in seq(1,1000)){
  preds <- NULL
  for (j in seq(10,length(series))){
    preds <- append(preds,multistep(10, series=series,start=j,psi = psi, sigma2 =
       sigma2 ))
  }
  nstepaheadpreds[glue("{n}")] <- preds
}

nstepaheadpreds <- nstepaheadpreds[, 2:134]

averages.nstepahead <- NULL

for(i in seq(1,134)){
  averages.nstepahead <- append(averages.nstepahead,
     forecastAverage(i,nstepaheadpreds, 1))
}

averages.nstepahead <- data.frame(averages.nstepahead)


vec <- rep(NA,9)
vec <- append(vec,averages.nstepahead)
averages.nstepahead <- data.frame(vec)


averages.nstepahead['St'] <- testdata$St


table.multi <- getAllPredictions(nstepaheadpreds,1,predsstart = 1)
table.multi <- data.frame(table.multi)
for (i in seq(2,134)){
  table.multi[glue("{i}")] <- getAllPredictions(nstepaheadpreds, i, predsstart = 1)
}

lst <- NULL
for (i in seq(1,ncol(table.multi))){
  lst <- append(lst, getQuantiles(c(0.025, 0.975), table.multi, i))
}

quantile.25 <- NULL
for (i in seq(1,268,2)){
  quantile.25 <- append(quantile.25,lst[i])
}
```

```r
temp <- rep(NA,9)
temp <- append(temp, quantile.25[1:134])
quantile.25 <- temp


quantile.75 <- NULL
for (i in seq(2,268,2)){
  quantile.75 <- append(quantile.75,lst[i])
}

temp <- rep(NA,9)
temp <- append(temp, quantile.75[1:134])
quantile.75 <- temp


df.series %>% ggplot(aes(x = 1:143, y = averages.nstepahead$vec)) +
  geom_line(aes(colour="Forecasted")) +
  geom_line(aes(x = 1:143, y = St, colour = "True Values"), alpha = 0.7) +
  geom_line(aes(x= 1:143, y = quantile.25 , colour = "2.5% Quantile"),
          size = 0.5,
          linetype = "dashed") +
  geom_line(aes(x = 1:143, y = quantile.75, colour = "97.5% Quantile"),
          size = 0.5,
          linetype = "dashed") +
  # geom_point(aes(x = 1:141, y = quantile.25, colour = "2.5% Quantile")) +
  # geom_point(aes(x = 1:141, y = quantile.75, colour = "97.5% Quantile")) +
  labs(x = "Time", y = TeX("$S_i$")) +
  ggtitle(TeX("$S_i$ vs $\\hat{S}_{i}$ 10-Step-Ahead Forecast"))



mse <- (1/(133-1))*sum((averages.nstepahead$St[10:143] -
   averages.nstepahead$vec[10:143])^2)



m11 <- M11(averages.nstepahead$vec[10:nrow(averages.nstepahead)],
           averages.nstepahead$St[10:nrow(averages.nstepahead)])

m21 <- M21(averages.nstepahead$vec[10:nrow(averages.nstepahead)],
            averages.nstepahead$St[10:nrow(averages.nstepahead)])

m12 <- M12(averages.nstepahead$vec[10:nrow(averages.nstepahead)],
           averages.nstepahead$St[10:nrow(averages.nstepahead)])

m22 <- M22(averages.nstepahead$vec[10:nrow(averages.nstepahead)],
           averages.nstepahead$St[10:nrow(averages.nstepahead)])

m20 <- m21 + m22
m10 <- m11 + m12
m01 <- m11 + m21
m02 <- m12 + m22

contigency.table <- data.frame(cbind(c(m11,m21),c(m12,m22)))
contigency.table <- contigency.table %>% rename('Up'='X1', 'Down'='X2')
```

```r
predictions <-
   createDummyvariables(averages.nstepahead$vec[10:nrow(averages.nstepahead)])
true <- createDummyvariables(averages.nstepahead$vec[10:nrow(averages.nstepahead)])

chisq.test(predictions, true)
```