STAT 153
Zhichao Yang
25779475

Midterm 2: Time Series Data Set 2 Analysis Report

Introduction

The analysis of data set q2train.csv is elaborated in this report. The original data is a non-stationary time series as indicated by the trend and seasonality. The analysis discovered that the data fit a seasonal ARIMA (2,1,1)x(1,1,1) model with a period of 52.

Trend

The original data was plotted in Figure 1. The original data has a decreasing trend, but the decreasing rate is not a constant as the trend gradually becomes horizontal. Therefore, the dataset needs to be differenced to remove the trend. In addition, the variances of the data set increase with time, especially for the first 150 data points, so a logarithms transformation was used before further analysis to stabilize variances (Figure 2). After a log transformation and a first difference, the data was plotted in Figure 3. The data points seem to be located around zero line and do not show a trend, so there is no need to do another first difference. The dataset does not show any outlier.

Figure1. Original Data Plot

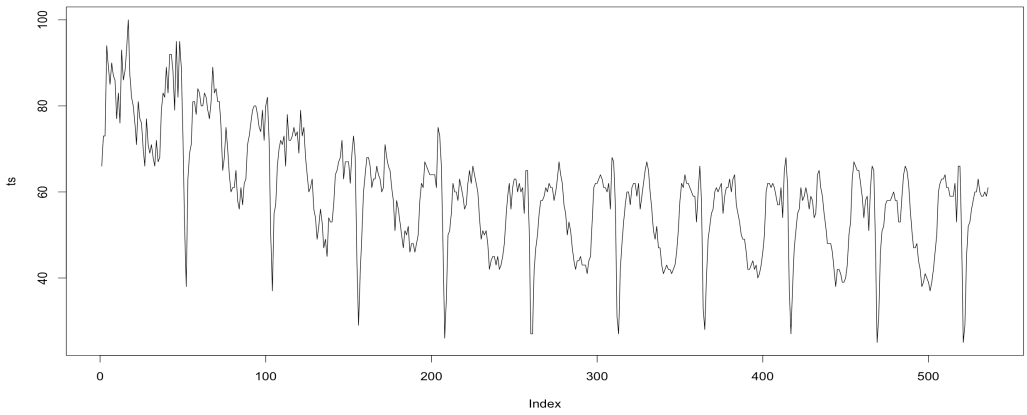


Figure2. Log Data Plot

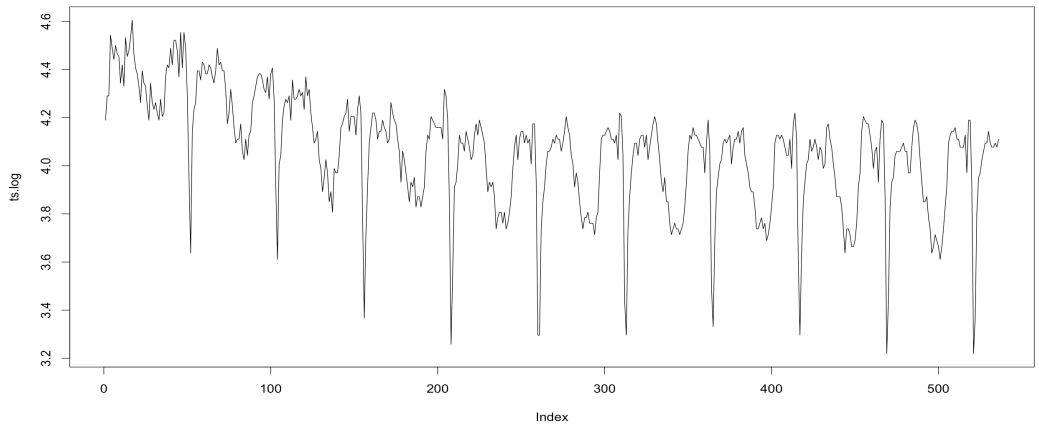
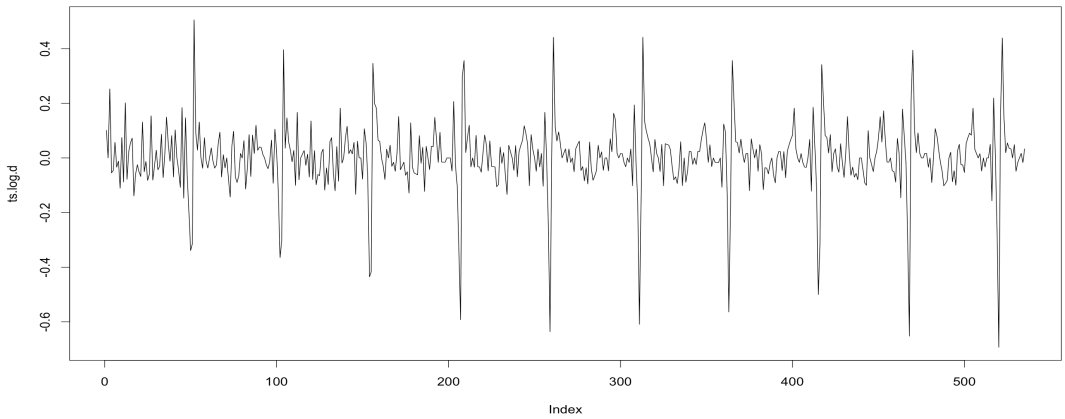


Figure3. Differenced Log Data



Seasonality

The plot for the differenced log data (Figure 3) shows that large fluctuations appear periodically, so there might be some seasonality. Then, ACF and PACF were used to examine the possible seasonality (Figure 4 and Figure 5). Both of the ACF and PACF show that there are large spikes around 52. Therefore, a period of 52 was used for a seasonal difference because it is a reasonable weekly frequency (365/7 = 52.57). The data after taking a seasonal difference looks stationary (Figure 6).

Figure4. ACF of Differenced Log Data

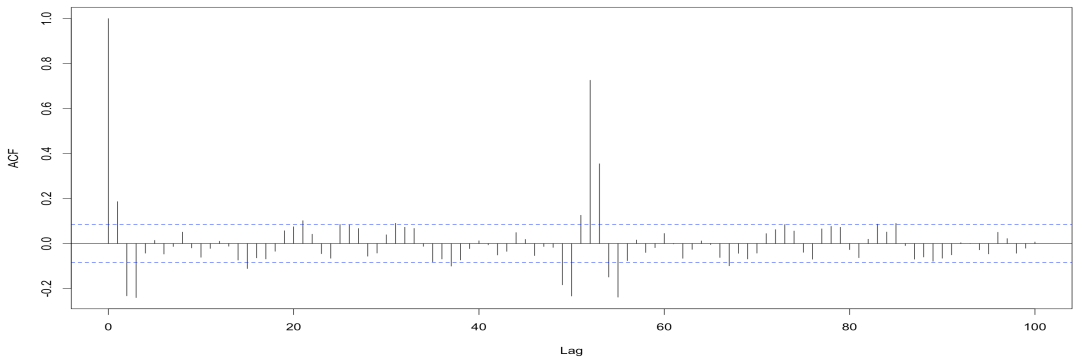


Figure5. PACF of Differenced Log Data

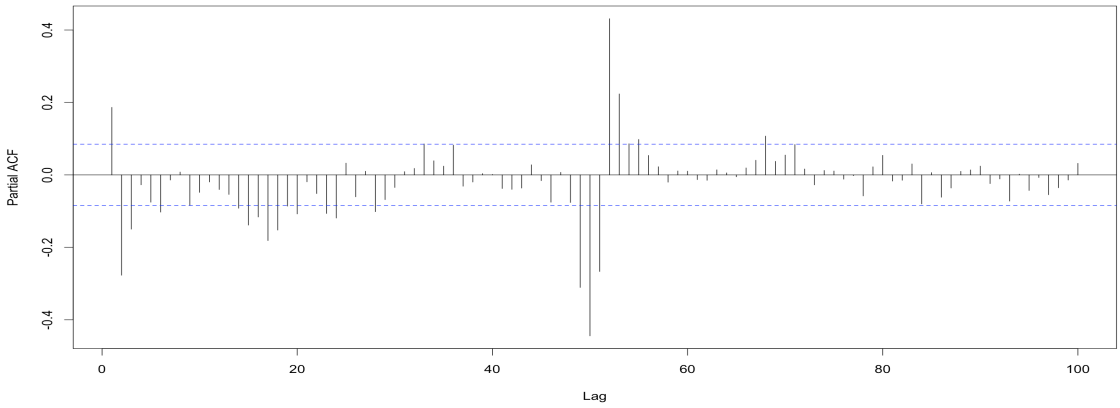
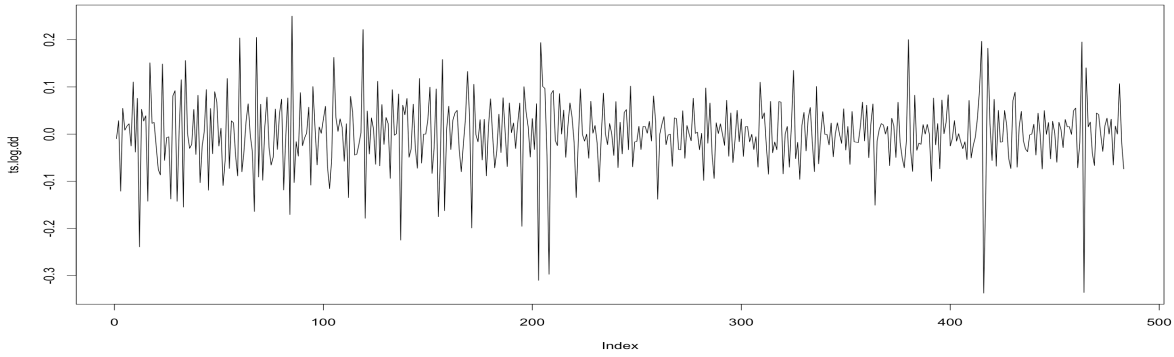


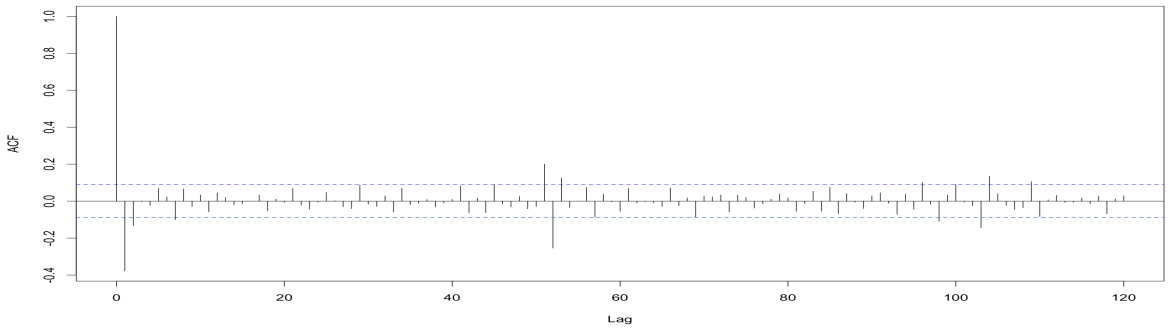
Figure6. Seasonal Differenced Log Data

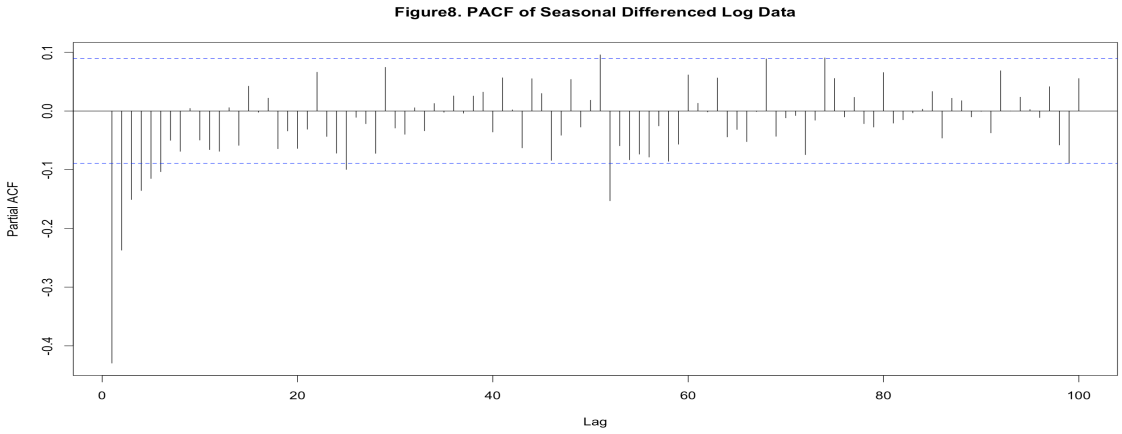


Model Fitting

The ACF and PACF of the log data after a first difference and a seasonal difference suggest that a seasonal ARIMA model might be a good fit to the log data (Figure 7 and Figure 8). In the ACF, there are seasonal peaks appears at lag 52 and 104, but the lag at 104 is relatively small. This indicates a seasonal MA(1) or MA(2). At the lower lag, there is one larger lag at lag 1, indicating a non-seasonal MA(1). In the PACF, there are two larger spikes at lag1 and lag 2, which might indicate a non-seasonal AR(2). There is only one large spike at lag 52, indicating a seasonal AR(1).

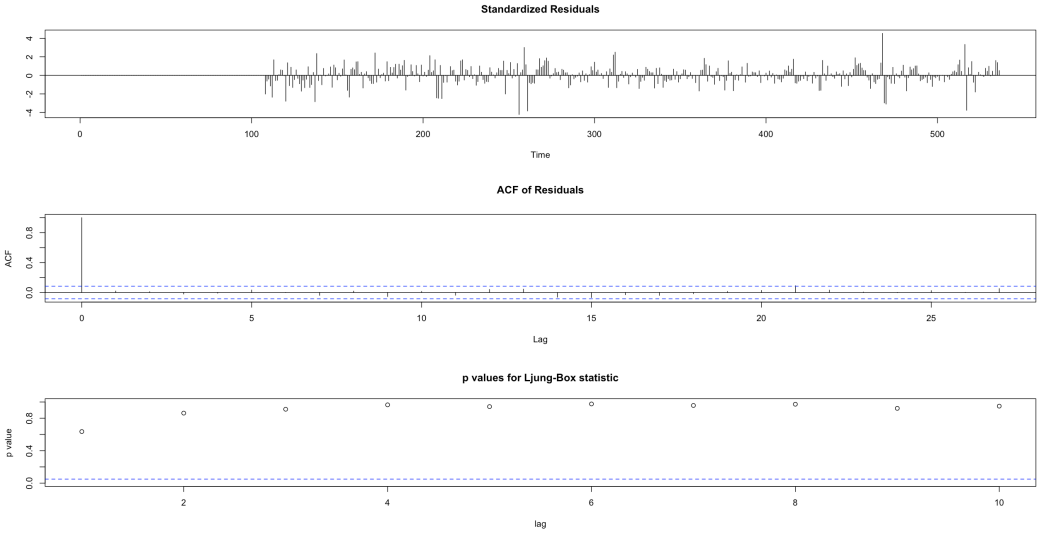
Figure7. ACF of Seasonal Differenced Log Data





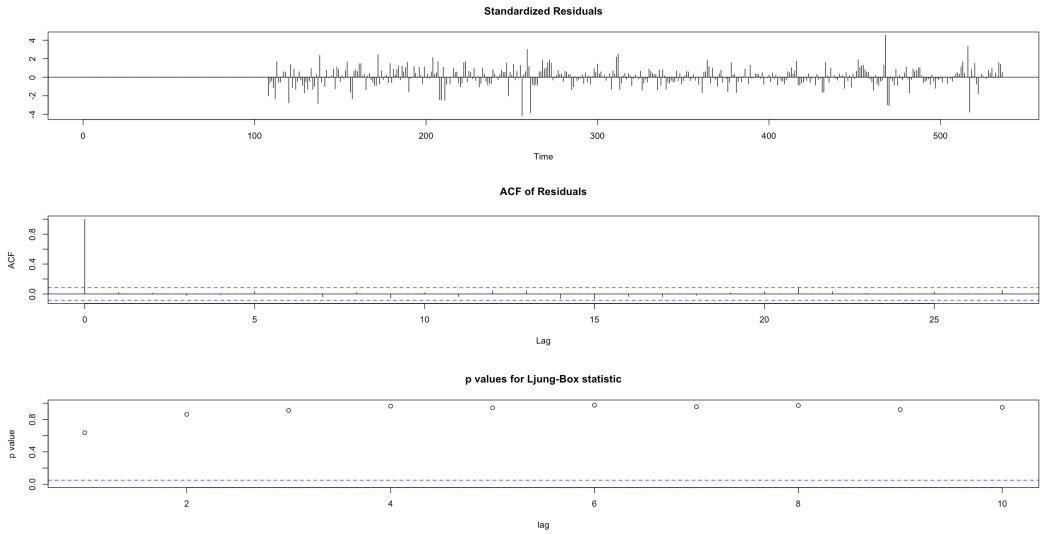
Based on the reasoning above, arima(ts.log, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 2), period = 52)) was used first, but it generated an error message. This error was avoided by choosing method = "CSS". The Ljung-Box test shows that the residuals look like a white noise. In addition, the p-values are large (Figure 9). This indicate Seasonal ARIMA(2,1,1)x(1,1,2)_52 is a reasonable model for the dataset.

Figure 9.



Since the lag 104 is reactively small in the ACF, another model Seasonal ARIMA(2,1,1)x(1,1,1)_52 was tested. This model also has a nice Ljung-Box test result. This two models gave similar Ljung-Box test result (Figure 10).

Figure 10.



Then I tried to overfit for diagnostics. Cross-validation was used for comparison. The results are listed in the following table. Generally, a model with small MSE values for all test sets is favored. However, there are some contradictions for the MSE values for different models and test sets. Therefore, it is hard to tell which one is the best model from the cross-validation result.

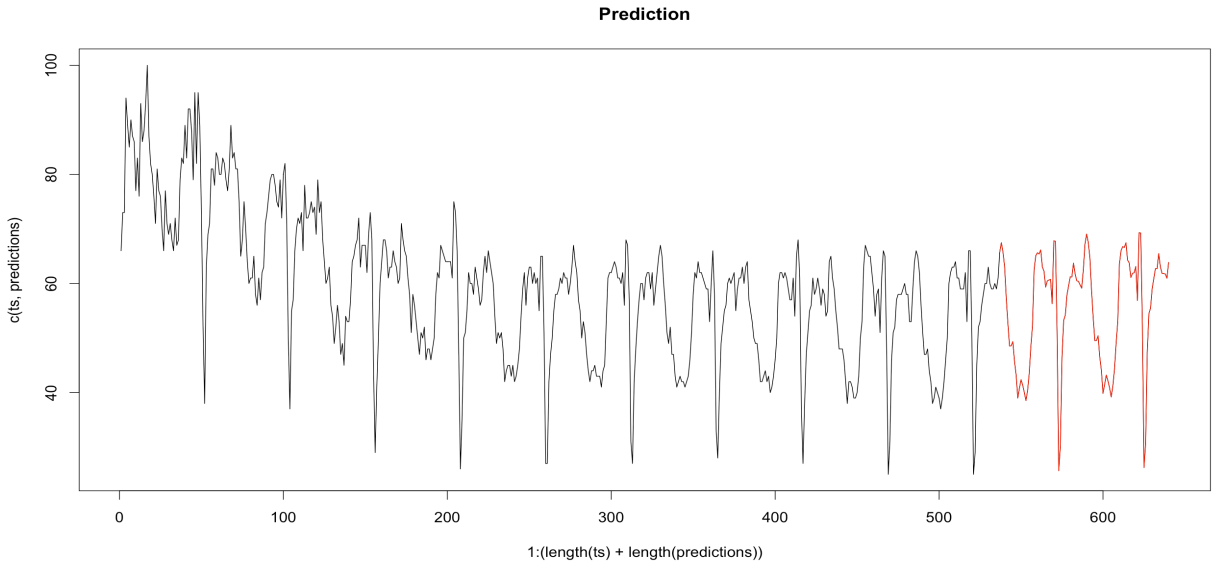
(2,1,1)x(1,1,2)_52	10.812738	13.688074	5.465700	5.034556	7.155017
(2,1,1)x(1,1,1)_52	10.064929	13.953001	5.435704	4.974302	8.711411
(2,1,1)x(1,1,3)_52	10.836691	13.755722	5.238900	5.441690	7.317123
(3,1,1)x(1,1,1)_52	9.971862	14.113133	5.277065	4.921742	7.157777
(4,1,1)x(1,1,1)_52	10.073620	13.474610	5.173049	4.863462	9.355224
(2,1,1)x(1,1,1)_52	9.773860	12.552077	5.325432	4.898208	11.245692
(2,1,1)x(2,1,1)_52	9.642972	11.804532	5.313977	5.617547	8.846564

Therefore, AIC was also used for testing and obtaining more information. In the function arima(), "CSS" is not chosen as the method because it does not generate an AIC value. The testing results are:

AIC(m3111) #-1372.11
AIC(m2111) #-1378.413
AIC(m2112) (Error message for (2,1,1)x(1,1,2)_52 with default method in arima)
AIC(m2113) (Error)
AIC(m2121) (Error)

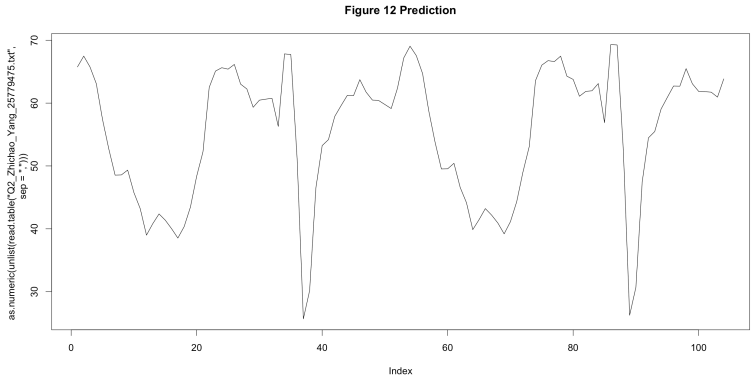
Based on the above output, I choose to use (2,1,1)x(1,1,1)_52 because it has the smallest AIC value among the AIC values I got. It has a similar residual test result as (2,1,1)x(1,1,2)_52. The MSE values are reasonable when compared to other models. In addition, this is the simplest model among these models and we don't want to overfit the data. The predictions were back transformed by exp() function. The predictions look reasonable because they have a similar seasonality as the previous data and follow a similar trend (Figure 11).

Figure 11.



Conclusion

A seasonal ARIMA (2,1,1)x(1,1,1)_52 model is used to do prediction and the predicted values are as blow:



1	65.7567768	27	63.01896937	53	67.18553049	79	64.28629743
2	67.48269505	28	62.24826591	54	69.06196003	80	63.7693016
3	65.74563408	29	59.33464343	55	67.55489993	81	61.09037848
4	63.09592848	30	60.4693981	56	64.70025797	82	61.85453798
5	57.36052358	31	60.63163809	57	58.68701446	83	61.96338819
6	52.6745965	32	60.75218775	58	53.73854304	84	63.11121939
7	48.53704572	33	56.29205428	59	49.51958642	85	56.87984029
8	48.57297916	34	67.8322676	60	49.54369147	86	69.31970122

9	49.3427484	35	67.73668038	61	50.42116605	87	69.25558602
10	45.74937286	36	51.24223931	62	46.56640656	88	52.41707733
11	43.20427888	37	25.66874255	63	44.13821019	89	26.24049503
12	38.9870389	38	30.16369279	64	39.86579436	90	30.6986385
13	40.8115572	39	46.5133689	65	41.44888676	91	47.44035859
14	42.36650059	40	53.24011614	66	43.21517411	92	54.47897563
15	41.36474918	41	54.19051785	67	42.18227914	93	55.47731965
16	40.01605049	42	57.87234889	68	40.91693207	94	59.02955877
17	38.52191287	43	59.58203093	69	39.19206874	95	60.89843887
18	40.33914682	44	61.22332243	70	41.13342084	96	62.7208915
19	43.44680283	45	61.1907217	71	44.30063794	97	62.69897364
20	48.37622236	46	63.73304548	72	49.0479175	98	65.48567967
21	52.23488479	47	61.74986817	73	53.08118462	99	63.0743406
22	62.51534852	48	60.48706672	74	63.58633467	100	61.8663971
23	65.10144325	49	60.41189906	75	66.04065462	101	61.81593658
24	65.63094819	50	59.77445117	76	66.75886484	102	61.74288275
25	65.40299462	51	59.13125291	77	66.60662027	103	60.95288347
26	66.15397015	52	62.37329903	78	67.47181863	104	63.85332401

Appendix

R Code

Data Set 1

```
ts = as.numeric(read.csv("q1train.csv", as.is = TRUE)[,2])
plot(ts, type = 'l')
ts.log = log(ts)
plot(ts.log, type ="l")

ts.log.d = diff(ts.log)
plot(ts.log.d, type = 'l')
# If I remove the extreme value of difference
ts.log.d.remove = ts.log.d[-max(ts.log.d)]

which.max(ts.log.d)

# Data set after removing the 291st data (extreme large)
ts.log.remove = ts.log[-291]

ts.log.remove.d = diff(ts.log.remove)
plot(diff(ts.log.remove), type ="l")
acf(ts.log.remove.d, lag.max = 100)
# 52 is a large peak, 48 spike is larger
# Like a MA(1) or MA(4)
pacf(ts.log.d.remove, lag.max = 100)


ts.log.remove.dd = diff(ts.log.remove.d, 52)
plot(ts.log.remove.dd, type = 'l')
acf(ts.log.remove.dd, lag.max = 120)$acf
# Large peak at lag 1 and 52
# Maybe an Arma(0,1) x (0,1)_52 model?
pacf(ts.log.remove.dd, lag.max = 100)
# Like there is AR(1) in nonseasonal part
```

```{r}
m1101 = arima(ts.log.remove, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1), period = 52))
tsdiag(m1101)
AIC(m1101)    #-1781.313

m2101 = arima(ts.log.remove, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 1), period = 52))
tsdiag(m2101)
AIC(m2101)    #-1783.406

m3101 = arima(ts.log.remove, order = c(3, 1, 1), seasonal = list(order = c(0, 1, 1), period = 52))
```

```
tsdiag(m3101)
AIC(m3101)  #-1781.481

m4101 = arima(ts.log.remove, order = c(4, 1, 1), seasonal = list(order = c(0, 1, 1), period = 52))
tsdiag(m4101)
AIC(m4101)  # -1782.574

m5101 = arima(ts.log.remove, order = c(5, 1, 1), seasonal = list(order = c(0, 1, 1), period = 52))
tsdiag(m5101)
AIC(m5101)  #-1780.956

m2102 = arima(ts.log.remove, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 2), period = 52))
tsdiag(m2102)
AIC(m2102)  # -1790.477 (best)

m2103 = arima(ts.log.remove, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 3), period = 52))
tsdiag(m2103)
AIC(m2103)  # -1788.836

m2104 = arima(ts.log.remove, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 4), period = 52))
tsdiag(m24)
AIC(m2104)  # -1792.126


m2105 = arima(ts.log.remove, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 5), period = 52))
tsdiag(m25)
AIC(m25)  #-1794.666 (best)

m1111 = arima(ts.log.remove, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1), period = 52))
AIC(m1111)  # -1788.131

m2115 = arima(ts.log.remove, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 5), period = 52))
AIC(m2115)

m2112 = arima(ts.log.remove, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 2), period = 52))
AIC(m2112) #-1788.178

m2111 = arima(ts.log.remove, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 1), period = 52))
AIC(m2112) #

m1211 = arima(ts.log.remove, order = c(1, 1, 2), seasonal = list(order = c(1, 1, 1), period = 52))
AIC(m1211) # -1788.727

m2202 = arima(ts.log.remove, order = c(2, 1, 2), seasonal = list(order = c(0, 1, 2), period = 52))
AIC(m2202) #-1788.481

m2212 = arima(ts.log.remove, order = c(2, 1, 2), seasonal = list(order = c(1, 1, 2), period = 52))
AIC(m2212) #-1786.579
```

Cross Validation

```{r}
len = length(ts.log.remove)
computeCVmse_css <- function(order.totry, seasorder.totry){
  MSE <- numeric()
  for(k in 5:1){
    train.dt <-ts.log.remove[1:(534 - 52 * k)]
    test.dt <- ts.log.remove[(534 - 52 * k + 1):(534 - 52 * (k - 1))]
    mod <- arima(train.dt, order = order.totry, seasonal =
                 list(order = seasorder.totry, period = 52), method ="CSS")
    fcast <- predict(mod, n.ahead = 52)
    MSE[k] <- mean((exp(fcast$pred) - exp(test.dt))^2)
  }
  return(MSE)
}

MSE1101 = computeCVmse_css(c(1, 1, 1), c(0,1,1))
MSE2101 = computeCVmse_css(c(2, 1, 1), c(0,1,1))
MSE3101 = computeCVmse_css(c(3, 1, 1), c(0,1,1))
MSE4101 = computeCVmse_css(c(4, 1, 1), c(0,1,1))
MSE5101 = computeCVmse_css(c(5, 1, 1), c(0,1,1))
MSE2102 = computeCVmse_css(c(2, 1, 1), c(0,1,2))
MSE2103 = computeCVmse_css(c(2, 1, 1), c(0,1,3))
MSE2104 = computeCVmse_css(c(2, 1, 1), c(0,1,4))
MSE2105 = computeCVmse_css(c(2, 1, 1), c(0,1,5))
MSE1111 = computeCVmse_css(c(1, 1, 1), c(1,1,1))
MSE2112 = computeCVmse_css(c(2, 1, 1), c(1,1,2))
MSE2111 = computeCVmse_css(c(2, 1, 1), c(1,1,1))
MSE1211 = computeCVmse_css(c(1, 1, 2), c(1,1,1))
MSE2202 = computeCVmse_css(c(2, 1, 2), c(0,1,2))
MSE2212 = computeCVmse_css(c(2, 1, 2), c(1,1,2))


MSE1101
# 6.660597 12.356367  4.676739 20.673818 12.027092
MSE2101
# 5.864038 13.644713  4.580949 19.334059 18.589256
MSE3101
# 5.868461 13.531838  4.563378 18.695856 17.883119
MSE4101
# 5.998073 13.244508  4.613155 18.458088 16.165758
MSE5101
# 8.410974 11.645552  4.938578 20.186772 10.990847
MSE2102
# 6.104807 15.288871  4.243730 19.465493 17.832232
MSE2103
# 6.058176 15.104179  4.174252 21.313647 15.868488
MSE2104
# 7.151009 14.783251  4.295514 20.254258 22.180124
MSE2105
```

```
#
MSE1111
# 5.670777 13.262230  4.962748 20.250862 21.224403
MSE2112
# 5.649047 18.107482  4.721789 21.693579 18.995336
MSE2111
# 5.670729 13.367873  5.009607 19.843830 22.530886
MSE1211
# 5.678744 13.450825  4.952889 20.289051 21.282025
MSE2202
# 6.106039 15.227472  4.231530 19.477098 17.672907
MSE2212
# 5.647661 17.987404  4.534863 21.328993 20.775155
```

```{r}
predictions = exp(predict(m1101, n.ahead = 104)$pred)

## Check: Does that make sense?
plot(1:(length(ts) + length(predictions)), c(ts, predictions), type = 'l', col = 1)
points((length(ts) + 1) : (length(ts) + length(predictions)), predictions, type = 'l', col = 2)


## Let's crete the file:
write.table(predictions,
        sep = ",",
        col.names = FALSE,
        row.names = FALSE,
        file = "Q1_Zhichao_Yang_25779475.txt")
        # file = "Exercise0_Firstname_Lastname_StudentIDNumber.txt")

# A quick check, that the file is what we expect it to be:
read.table("Q1_Zhichao_Yang_25779475.txt", sep = ",")
plot(as.numeric(unlist(read.table("Q1_Zhichao_Yang_25779475.txt", sep = ","))))
```

Data Set 2

```{r}
ts = as.numeric(read.csv("q2train.csv", as.is = TRUE)[,2])
plot(ts, type = 'l', main ="Figure1. Original Data Plot")
ts.log = log(ts)
plot(ts.log, type ="l", main = "Figure2. Log Data Plot")

ts.log.d = diff(ts.log)
plot(ts.log.d, type = 'l', main = "Figure3. Differenced Log Data")
acf(ts.log.d, lag.max = 100, main = "Figure4. ACF of Differenced Log Data")
# 52 is a large peak
pacf(ts.log.d, lag.max = 100, main = "Figure5. PACF of Differenced Log Data")
```

```
ts.log.dd = diff(ts.log.d, 52)
plot(ts.log.dd, type = 'l', main = "Figure6. Seasonal Differenced Log Data")
acf(ts.log.dd, lag.max = 120, main = "Figure7. ACF of Seasonal Differenced Log Data")$acf
# Large peak at lag 1 and 52, (48)
# MA(1), seasonal MA(1)
pacf(ts.log.dd, lag.max = 105, main = "Figure8. PACF of Seasonal Differenced Log Data")
# peaks at 52, there is a seasonal AR(1)
```

```{r}
m2112 = arima(ts.log, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 2), period = 52))
# Error message
#Error in optim(init[mask], armafn, method = optim.method, hessian = TRUE,  :
 # non-finite finite-difference value [4]

m2112_css = arima(ts.log, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 2), period = 52),
method = "CSS")
tsdiag(m2112_css)
m2111_css = arima(ts.log, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 1), period = 52),
method = "CSS")
tsdiag(m2111_css)
m3111_css = arima(ts.log, order = c(3, 1, 1), seasonal = list(order = c(1, 1, 1), period = 52),
method = "CSS")
tsdiag(m3111_css)
m2113_css = arima(ts.log, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 3), period = 52),
method = "CSS")
tsdiag(m2113_css)


computeCVmse_css <- function(order.totry, seasorder.totry){
  MSE <- numeric()
  for(k in 5:1){
    train.dt <-ts.log[1:(len - 52 * k)]
    test.dt <- ts.log[(len - 52 * k + 1):(len - 52 * (k - 1))]
    mod <- arima(train.dt, order = order.totry, seasonal =
                   list(order = seasorder.totry, period = 52), method ="CSS")
    fcast <- predict(mod, n.ahead = 52)
    MSE[k] <- mean((exp(fcast$pred) - exp(test.dt))^2)
  }
  return(MSE)
}

MSE_m2112_css = computeCVmse_css(c(2, 1, 1), c(1,1,2))
MSE_m2111_css = computeCVmse_css(c(2, 1, 1), c(1,1,1))
MSE_m2113_css = computeCVmse_css(c(2, 1, 1), c(1,1,3))
MSE_m3111_css = computeCVmse_css(c(3, 1, 1), c(1,1,1))
MSE_m4111_css = computeCVmse_css(c(4, 1, 1), c(1,1,1))
MSE_m2211_css = computeCVmse_css(c(2, 1, 2), c(1,1,1))
MSE_m2121_css = computeCVmse_css(c(2, 1, 1), c(2,1,1))
```

```
MSE_m2112_css
# 10.812738 13.688074  5.465700  5.034556  7.155017
MSE_m2111_css
# 10.064929 13.953001  5.435703  4.974302  8.711411
MSE_m2113_css
# 10.836691 13.755722  5.238900  5.441690  7.317123
MSE_m3111_css
# 9.971862  14.113133  5.277065  4.921742  7.157777
MSE_m4111_css
# 10.073620 13.474610  5.173049  4.863462  9.355224
MSE_m2211_css
# 9.773860 12.552077  5.325432  4.898208 11.245692
MSE_m2121_css
#  9.642972 11.804532  5.313977  5.617547  8.846564
```

```{r}
m3111 = arima(ts.log, order = c(3, 1, 1), seasonal = list(order = c(1, 1, 1), period = 52),
optim.method ="Nelder-Mead")
AIC(m3111)    #-1372.11
m2111_NM = arima(ts.log, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 1), period = 52),
optim.method ="Nelder-Mead")
AIC(m2111_NM) #-1378.413
m2112_NM = arima(ts.log, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 2), period = 52),
optim.method ="Nelder-Mead")
# Error in optim(init[mask], armafn, method = optim.method, hessian = TRUE,  :
 #non-finite finite-difference value [4]
m2113_NM = arima(ts.log, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 3), period = 52),
optim.method ="Nelder-Mead")
# Error in optim(init[mask], armafn, method = optim.method, hessian = TRUE,  :
 #non-finite finite-difference value [4]
m2121_NM = arima(ts.log, order = c(2, 1, 1), seasonal = list(order = c(2, 1, 1), period = 52),
optim.method ="Nelder-Mead", method ="ML")
# Error in optim(init[mask], armafn, method = optim.method, hessian = TRUE,  :
 #function cannot be evaluated at initial parameters

```

```{r}
predictions = exp(predict(m2111, n.ahead = 104)$pred)

## Check: Does that make sense?
plot(1:(length(ts) + length(predictions)), c(ts, predictions), type = 'l', col = 1, main =
"Prediction")
points((length(ts) + 1) : (length(ts) + length(predictions)), predictions, type = 'l', col = 2)


## Let's crete the file:
write.table(predictions,
        sep = ",",
```

```
        col.names = FALSE,
        row.names = FALSE,
        file = "Q2_Zhichao_Yang_25779475.txt")
        # file = "Q2_Zhichao_Yang_25779475.txt")

# A quick check, that the file is what we expect it to be:
read.table("Q2_Zhichao_Yang_25779475.txt", sep = ",")
plot(as.numeric(unlist(read.table("Q2_Zhichao_Yang_25779475.txt", sep = ","))), type ="l",
main ="Figure 12 Prediction")
```

Data Set 3

```{r}
ts = as.numeric(read.csv("q3train.csv", as.is = TRUE)[,2])
plot(ts, type = 'l')

ts.d = diff(ts)
plot(ts.d, type = 'l')
# There is one extremeley value around 500. Remove it
which.max(ts.d)
max(ts.d)

ts.remove = ts[-497]
ts.log.remove = log(ts.remove)
ts.log.remove.d = diff(ts.log.remove)
plot(ts.log.remove.d, type ="l")

acf(ts.log.remove.d, lag.max = 100)
# 52 is a large peak
pacf(ts.log.remove.d, lag.max = 100)

ts.log.remove.dd = diff(ts.log.remove.d, 52)
plot(ts.log.remove.dd, type = 'l')
acf(ts.log.remove.dd, lag.max = 120)$acf
# Large peak at lag 1 and 52, MA(1)
# Maybe an Arma(0,1) x (0,1)_52 model?
pacf(ts.log.remove.dd, lag.max = 106)



m0111 = arima(ts.log.remove, order = c(0, 1, 1), seasonal = list(order = c(1, 1, 1), period = 52))
tsdiag(m0111) #BAD
AIC(m0111)    #-1538.018

m1111 = arima(ts.log.remove, order = c(1, 1, 1 ), seasonal = list(order = c(1, 1, 1), period = 52))
tsdiag(m1111)
AIC(m1111)   # -1567.402
```

```
m2111 = arima(ts.log.remove, order = c(2, 1, 1 ), seasonal = list(order = c(1, 1, 1), period = 52))
# Error

m3111 = arima(ts.log.remove, order = c(3, 1, 1), seasonal = list(order = c(1, 1, 1), period = 52))
tsdiag(m3111)
AIC(m3111)   # -1563.78

m4111 = arima(ts.log.remove, order = c(4, 1, 1), seasonal = list(order = c(1, 1, 1), period = 52))
tsdiag(m4111)
AIC(m4111)   #-1562.955

m7111 = arima(ts.log.remove, order = c(7, 1, 1), seasonal = list(order = c(1, 1, 1), period = 52))
tsdiag(m7111)
AIC(m7111)   #-1555.764

m0101 = arima(ts.log.remove, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 52))
tsdiag(m0101)
AIC(m0101)    #-1536.685

m1211 = arima(ts.log.remove, order = c(1, 1, 2), seasonal = list(order = c(1, 1, 1), period = 52))
tsdiag(m1211)
AIC(m1211)   #-1565.44

m1311 = arima(ts.log.remove, order = c(1, 1, 3), seasonal = list(order = c(1, 1, 1), period = 52))
tsdiag(m1311)
AIC(m1311)  # -1563.704

m1101 = arima(ts.log.remove, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1), period = 52))
tsdiag(m1101)
AIC(m1101)    #-1568.635

m1121 = arima(ts.log.remove, order = c(1, 1, 1), seasonal = list(order = c(2, 1, 1), period = 52))
# Error

m1110 = arima(ts.log.remove, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 0), period = 52))
tsdiag(m1110)
AIC(m1110)  # -1537.098

m1113 = arima(ts.log.remove, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 3), period = 52))
tsdiag(m1113)
AIC(m1113)  #

m3101 = arima(ts.log.remove, order = c(3, 1, 1), seasonal = list(order = c(0, 1, 1), period = 52))
tsdiag(m3101)
AIC(m3101)    #-1565.044

m2211 = arima(ts.log.remove, order = c(2, 1, 2 ), seasonal = list(order = c(1, 1, 1), period = 52))
tsdiag(m2211)
AIC(m2211)   # -1565.361
```

```

Cross Validation
```{r}
len <- length(ts.log.remove)
# we have a period of 52 so let's try to predict entire periods:

computeCVmse <- function(order.totry, seasorder.totry){
 MSE <- numeric()
 for(k in 5:1){
  train.dt <-ts.log.remove[1:(len - 52 * k)]
  test.dt <- ts.log.remove[(len - 52 * k + 1):(len - 52 * (k - 1))]
  mod <- arima(train.dt, order = order.totry, seasonal =
           list(order = seasorder.totry, period = 52), method ="CSS")
  fcast <- predict(mod, n.ahead = 52)
  MSE[k] <- mean((exp(fcast$pred) - exp(test.dt))^2)
 }
 return(MSE)
}

MSE0111 <- computeCVmse(c(0, 1, 1), c(1,1,1))
MSE1111 <- computeCVmse(c(1, 1, 1), c(1,1,1))
MSE2111 <- computeCVmse(c(2, 1, 1), c(1,1,1))
MSE3111 <- computeCVmse(c(3, 1, 1), c(1,1,1))
MSE4111 <- computeCVmse(c(4, 1, 1), c(1,1,1))
MSE7111 <- computeCVmse(c(7, 1, 1), c(1,1,1))
MSE0101 <- computeCVmse(c(0, 1, 1), c(0,1,1))
MSE1211 <- computeCVmse(c(1, 1, 2), c(1,1,1))
MSE1311 <- computeCVmse(c(1, 1, 3), c(1,1,1))
MSE1101 <- computeCVmse(c(1, 1, 1), c(0,1,1))
MSE1121 <- computeCVmse(c(1, 1, 1), c(2,1,1))
MSE1110 <- computeCVmse(c(1, 1, 1), c(1,1,0))
MSE1113 <- computeCVmse(c(1, 1, 1), c(1,1,3))
MSE3101 <- computeCVmse(c(3, 1, 1), c(0,1,1))
MSE2211 <- computeCVmse(c(2, 1, 2), c(1,1,1))


MSE0111
# 11.636928 21.246921 13.524106 38.715634  6.841146
MSE1111 ####
# 11.385174 23.866787 11.279660 38.787082  7.625344
MSE2111
# 11.253564 23.421892 11.158787 38.304872  7.365775
MSE3111
# 11.247768 23.493219 14.532471 40.638774  6.200849
MSE4111
# 11.230966 23.325376  9.487191 40.811542  6.191650
MSE7111
```
```

```
# 11.485249 23.498027 12.806927 39.663210  6.783688
MSE0101
# 10.287546 12.509839 14.446796 39.537310  4.303717
MSE1211
# 11.375466 23.899877 11.311860 38.386829  7.685642
MSE1311
# 11.24684  22.09681  11.93494  39.59851   7.32042
MSE1101 ####
# 10.662259 14.697330 12.354186 38.909947  4.334845
MSE1121
# 11.659999 20.950694 11.189214 38.869594  7.461068
MSE1110
# 12.118878 32.732837  9.114776 37.894979  6.659545
MSE1113
# 10.307880 18.039492 12.442655 41.444642  7.907723
MSE3101
# 10.588371 15.265414 13.268514 40.054294  4.197017
MSE2211
# 11.250743 21.942052 11.501945 39.760867  7.485821
```

```{r}
predictions = exp(predict(m1101, n.ahead = 104)$pred)
## Check: Does that make sense?
plot(1:(length(ts) + length(predictions)), c(ts, predictions), type = 'l', col = 1)
points((length(ts) + 1) : (length(ts) + length(predictions)), predictions, type = 'l', col = 2)
# Great :)

## Let's crete the file:
write.table(predictions,
        sep = ",",
        col.names = FALSE,
        row.names = FALSE,
        file = "Q3_Zhichao_Yang_25779475.txt")
        # file = "Exercise0_Firstname_Lastname_StudentIDNumber.txt")
# Q3_Zhichao_Yang_25779475
# A quick check, that the file is what we expect it to be:
read.table("Q3_Zhichao_Yang_25779475.txt", sep = ",")
plot(as.numeric(unlist(read.table("Q3_Zhichao_Yang_25779475.txt", sep = ","))))
```

Data Set 4

```{r}
ts = as.numeric(read.csv("q4train.csv", as.is = TRUE)[,2])
plot(ts, type = 'l')

ts.log = log(ts)
plot(ts.log, type ="l")
```

```
ts.log.d = diff(ts.log)
plot(ts.log.d, type = 'l')

acf(ts.log.d, lag.max = 100)
# 52 is a large peak, 48 spike is larger
# Like a MA(1) or MA(4)
pacf(ts.log.d, lag.max = 100)


ts.log.dd = diff(ts.log.d, 52)
plot(ts.log.dd, type = 'l')
acf(ts.log.dd, lag.max = 158)$acf
# Large peak at lag 1, a smaller peak at lag 52, MA(1) or with seasonal MA(1)
pacf(ts.log.dd, lag.max = 110)
# Seasonal AR, AR(3)

m3101 = arima(ts.log, order = c(3, 1, 1), seasonal = list(order = c(0, 1, 1), period = 52))
tsdiag(m3101)
AIC(m3101)    #-1402.89

m2101 = arima(ts.log, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 1), period = 52))
tsdiag(m2101)
AIC(m2101)    #-1405.392

m3111 = arima(ts.log, order = c(3, 1, 1), seasonal = list(order = c(1, 1, 1), period = 52))
tsdiag(m3111)
AIC(m3111)   #-1401.488

m3112 = arima(ts.log, order = c(3, 1, 1), seasonal = list(order = c(1, 1, 2), period = 52))
tsdiag(m3112)
AIC(m3112) #-1399.796

m2111 = arima(ts.log, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 1), period = 52))
tsdiag(m2111)
AIC(m2111) # -1404.084

m2112 = arima(ts.log, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 2), period = 52))
# Error

m1111 = arima(ts.log, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1), period = 52))
# Error

m1112 = arima(ts.log, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 2), period = 52))
# Error

m0101 = arima(ts.log, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 52))
AIC(m0101)
tsdiag(m0101)

m1101 = arima(ts.log, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1), period = 52))
```

```
AIC(m1101) # -1407.277
tsdiag(m1101)

```

Cross Validation

```{r}
len <- length(ts.log)
# we have a period of 52 so let's try to predict entire periods:

computeCVmse <- function(order.totry, seasorder.totry){
  MSE <- numeric()
  for(k in 5:1){
    train.dt <-ts.log[1:(len - 52 * k)]
    test.dt <- ts.log[(len - 52 * k + 1):(len - 52 * (k - 1))]
    mod <- arima(train.dt, order = order.totry, seasonal =
                 list(order = seasorder.totry, period = 52), method = "CSS")
    fcast <- predict(mod, n.ahead = 52)
    MSE[k] <- mean((exp(fcast$pred) - exp(test.dt))^2)
  }
  return(MSE)
}

MSE3101 = computeCVmse(c(3, 1, 1), c(0,1,1))
MSE2101 = computeCVmse(c(2, 1, 1), c(0,1,1))
MSE3111 = computeCVmse(c(3, 1, 1), c(1,1,1))
MSE3112 = computeCVmse(c(3, 1, 1), c(1,1,2))
MSE2111 = computeCVmse(c(2, 1, 1), c(1,1,1))
MSE2112 = computeCVmse(c(2, 1, 1), c(1,1,2))
MSE1111 = computeCVmse(c(1, 1, 1), c(1,1,1))
MSE1112 = computeCVmse(c(1, 1, 1), c(1,1,2))
MSE1121 = computeCVmse(c(1, 1, 1), c(2,1,1))
MSE1101 = computeCVmse(c(1, 1, 1), c(0,1,1))
MSE1102 = computeCVmse(c(1, 1, 1), c(0,1,2))
MSE1201 = computeCVmse(c(1, 1, 2), c(0,1,1))
MSE0101 = computeCVmse(c(0, 1, 1), c(0,1,1))

MSE3101
# 48.321611 29.462174  7.936428  9.343114 11.433418
MSE2101
# 47.098052 30.639286  8.365653  9.268876 11.494813
MSE3111
# 48.177622 30.549521  7.894156  9.330762 12.284385
MSE3112
# 48.174077 30.613445  7.883054  9.678646 21.076463
MSE2111
# 48.495996 30.074450  7.859315  9.663337 11.747275
MSE2112
# 48.641351 30.204707  7.840029 10.069648 20.552462
```

```
MSE1111
# 42.405070 31.968461  8.391786 10.186023 11.529298
MSE1112
# 42.453116 32.143098  8.351001 10.312267 18.713868
MSE1121
# 40.055110 30.607209  9.685140  9.961085 17.703409
MSE1101
# 42.240579 31.420035  8.501979  9.814225 11.329723
MSE1102
# 41.967970 32.348763  8.267529 10.637154 19.769704
MSE1201
# 46.878945 31.154393  8.474460  9.528338 11.340364
MSE0101
# 41.944615 31.320226  8.517714 10.491552 11.280750
```

```{r}
predictions = exp(predict(m0101, n.ahead = 104)$pred)

## Check: Does that make sense?
plot(1:(length(ts) + length(predictions)), c(ts, predictions), type = 'l', col = 1)
points((length(ts) + 1) : (length(ts) + length(predictions)), predictions, type = 'l', col = 2)

## Let's crete the file:
write.table(predictions,
        sep = ",",
        col.names = FALSE,
        row.names = FALSE,
        file = "Q4_Zhichao_Yang_25779475.txt")
        # file = "Exercise0_Firstname_Lastname_StudentIDNumber.txt")

# A quick check, that the file is what we expect it to be:
read.table("Q4_Zhichao_Yang_25779475.txt", sep = ",")
plot(as.numeric(unlist(read.table("Q4_Zhichao_Yang_25779475.txt", sep = ","))))
```

Data Set 5

```{r}
ts = as.numeric(read.csv("q5train.csv", as.is = TRUE)[,2])
plot(ts, type = 'l')
# Non-constant variance, so take a log
ts.log = log(ts)
plot(ts.log, type ="l")

ts.log.d = diff(ts.log)
plot(ts.log.d, type = 'l')
acf(ts.log.d, lag.max = 100)$acf
```

```
#  52 is very big
pacf(ts.log.d, lag.max = 100)

ts.log.dd = diff(ts.log.d, 52)
plot(ts.log.dd, type = 'l')
acf(ts.log.dd, lag.max = 110)
#MA(1) seasonal , MA(1) non-seasonal
pacf(ts.log.dd, lag.max = 110)
#AR(1) seasonal, AR(2)? non-seasonal
```

```{r}
m2111 = arima(ts.log, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 1), period = 52),
optim.method="Nelder-Mead")
tsdiag(m2111)
AIC(m2111)   # -1616.428
BIC(m2111)   # -1591.348

m1111 = arima(ts.log, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1), period = 52),
optim.method="Nelder-Mead")
# ERROR

m1101 = arima(ts.log, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1), period = 52))
tsdiag(m1101)
AIC(m1101)   #-1617.638
BIC(m1101)   # -1600.918

m1102 = arima(ts.log, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 2), period = 52))
tsdiag(m1102)
AIC(m1102)   #-1617.819 (best)
BIC(m1102)   #-1596.919 (best)

m1103 = arima(ts.log, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 3), period = 52))
tsdiag(m1103)
AIC(m1103)   #-1617.091
BIC(m1103)   #-1592.011

m2102 = arima(ts.log, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 2), period = 52))
tsdiag(m2102)
AIC(m2102)   #-1617.018
BIC(m2102)   #-1591.938

m3102 = arima(ts.log, order = c(3, 1, 1), seasonal = list(order = c(0, 1, 2), period = 52))
tsdiag(m3102)
AIC(m3102)   #-1615.02
BIC(m3102)   #-1585.76

m1202 = arima(ts.log, order = c(1, 1, 2), seasonal = list(order = c(0, 1, 2), period = 52))
tsdiag(m1202)
AIC(m1202)   # -1617.687
```

```
BIC(m1202)   # -1592.607
```

```{r}
len <- length(ts.log)
computeCVmse <- function(order.totry, seasorder.totry){
  MSE <- numeric()
  for(k in 5:1){
    train.dt <-ts.log[1:(len - 52 * k)]
    test.dt <- ts.log[(len - 52 * k + 1):(len - 52 * (k - 1))]
    mod <- arima(train.dt, order = order.totry, seasonal =
                 list(order = seasorder.totry, period = 52), method ="CSS")
    fcast <- predict(mod, n.ahead = 52)
    MSE[k] <- mean((exp(fcast$pred) - exp(test.dt))^2)
  }
  return(MSE)
}

MSE2111=  computeCVmse(c(2, 1, 1), c(1,1,1))
MSE1111 = computeCVmse(c(1, 1, 1), c(1,1,1))
MSE1101 = computeCVmse(c(1, 1, 1), c(0,1,1))
MSE1102 = computeCVmse(c(1, 1, 1), c(0,1,2))
MSE1103 = computeCVmse(c(1, 1, 1), c(0,1,3))
MSE2102 = computeCVmse(c(2, 1, 1), c(0,1,2))
MSE3102 = computeCVmse(c(3, 1, 1), c(0,1,2))
MSE1202 = computeCVmse(c(1, 1, 2), c(0,1,2))
MSE3202 = computeCVmse(c(3, 1, 2), c(0,1,2))

MSE3111 = computeCVmse(c(3, 1, 1), c(1,1,1))
MSE4111 = computeCVmse(c(3, 1, 1), c(1,1,1))
MSE1112 = computeCVmse(c(1, 1, 1), c(1,1,2))
MSE1113 = computeCVmse(c(1, 1, 1), c(1,1,3))
MSE1114 = computeCVmse(c(1, 1, 1), c(1,1,4))
MSE1121 = computeCVmse(c(1, 1, 1), c(2,1,1))

MSE0111 = computeCVmse(c(0, 1, 1), c(1,1,1))


MSE2111
# 11.619969  7.971270  7.217957  4.777166  2.599799
MSE1111
# 11.473611  7.931312  6.837845  4.571931  2.581373
MSE1101
# 11.270861  8.127022  6.205084  4.155766  2.729998
MSE1102 #(BEST FROM LAST)
# 11.276550  8.049701  6.301963  4.016296  2.510855
MSE1103
# 11.177003  8.170938  6.325626  3.618124  2.507304
MSE2102
# 11.232946  8.186069  6.093097  3.918444  2.512841
```

MSE3102 ######
11.231380 8.201410 6.139533 3.808250 2.474115
MSE1202
11.316043 8.037518 6.291410 3.999733 2.518553
MSE3202
11.237958 9.853694 5.941698 3.819835 2.472283
MSE3111
11.539641 7.919424 7.107281 4.532110 2.574115
MSE4111
11.539641 7.919424 7.107281 4.532110 2.574115
MSE1112
11.484920 8.029074 6.929456 4.541597 2.946189
MSE1113
11.320388 8.106017 7.157839 3.828187 2.802412
MSE1114
#
MSE1121
11.759157 8.286234 8.517560 4.791592 2.882346
MSE0111
11.339994 8.024564 6.712422 4.550746 2.552194
```

```{r}
predictions = exp(predict(m1102, n.ahead = 104)$pred)

predictions2 = exp(predict(m3102, n.ahead = 104)$pred)

## Check: Does that make sense?
plot(1:(length(ts) + length(predictions)), c(ts, predictions), type = 'l', col = 1)
points((length(ts) + 1) : (length(ts) + length(predictions)), predictions, type = 'l', col = 2)
# Great :)

plot(1:(length(ts) + length(predictions2)), c(ts, predictions2), type = 'l', col = 1)
points((length(ts) + 1) : (length(ts) + length(predictions2)), predictions2, type = 'l', col = 2)


## Let's crete the file:
write.table(predictions,
        sep = ",",
        col.names = FALSE,
        row.names = FALSE,
        file = "Q5.txt")
        # file = "Exercise0_Firstname_Lastname_StudentIDNumber.txt")

# A quick check, that the file is what we expect it to be:
read.table("Q5_Zhichao_Yang_25779475.txt", sep = ",")
```
```

```
plot(as.numeric(unlist(read.table("Q5_Zhichao_Yang_25779475.txt", sep = ","))))
```

```
## Let's crete the file:
write.table(predictions2,
        sep = ",",
        col.names = FALSE,
        row.names = FALSE,
        file = "Q5_Zhichao_Yang_25779475.txt")
        # file = "Exercise0_Firstname_Lastname_StudentIDNumber.txt")

# A quick check, that the file is what we expect it to be:
read.table("Q5_Zhichao_Yang_25779475.txt", sep = ",")
plot(as.numeric(unlist(read.table("Q5_Zhichao_Yang_25779475.txt", sep = ","))))
```