Midterm 2 Project Report

Xiaoyu Sun, Rick Chen

Nov 10, 2017

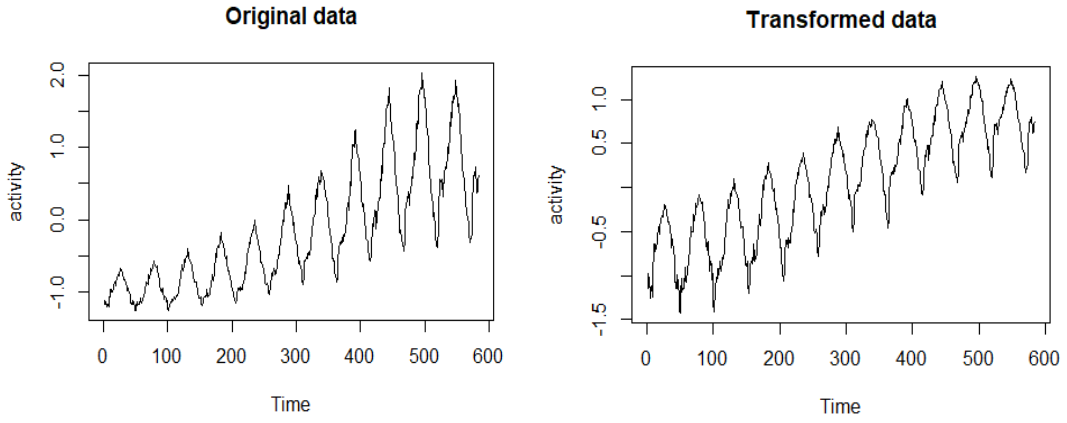
1 Introduction

In this report, we perform an analysis of the second dataset ("q2_train"). First, we use log transform to stabilize the variance. Then, we inspect the differenced data and ACF of residuals to identify candidate models for our dataset, and use several validity criteria to search for the best model. Finally, we obtain the desired predictions.

2 Building SARIMA Models

2.1 Data Transformation

As the first step of data analysis, we should construct a time plot of our data, and inspect the graph for anomalies.



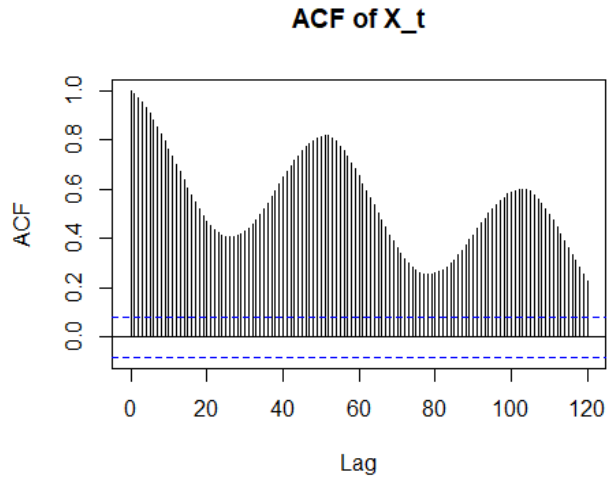
The original data, plotted above on the left, displays heteroskedasticity: the variability in the data grows with time. To stabilize the variance, we perform log transformation on the original data $\{x_t\}$. Note that some values are negative, so we need to shift them above zero before using logarithms. Based on this consideration, we obtain the transformed data $\{X_t\}$ by

$$X_t = log(x_t + 1.5).$$

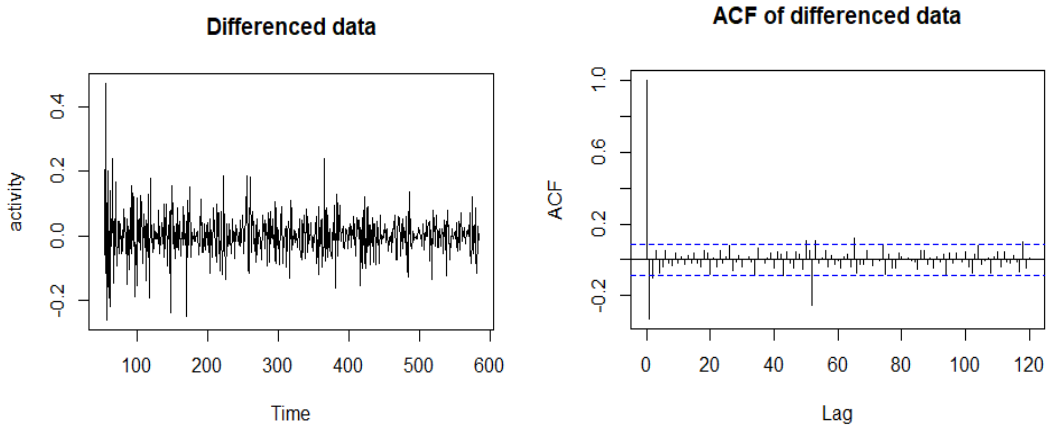
For $\{X_t\}$, heteroskedasticity is greatly reduced, as shown in the right panel above.

2.2 Order Selection

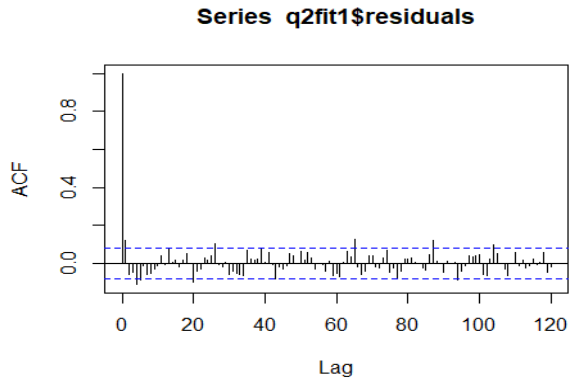
We plot the sample autocorrelation function, $\hat{\rho}(h)$, of our transformed data below.



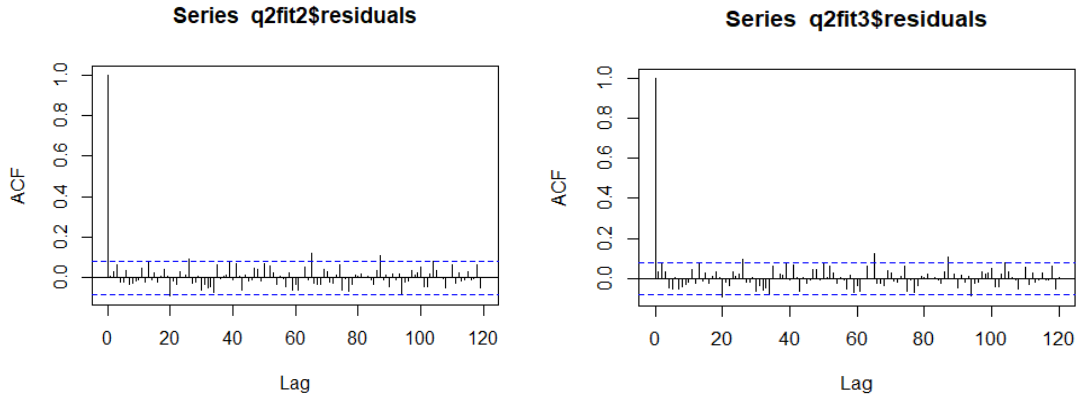
Over the first 20 lags, there is a slow decay in $\hat{\rho}(h)$; then, at around lag 52, $\hat{\rho}(h)$ attains its local maximum. This phenomenon indicates that differencing of lags 1 and 52 may both be needed. To see whether this is true, we inspect the time plot and sample ACF of $\nabla_{52}\nabla X_t = (1 - B^{52})(1 - B)X_t$, where B is the backshift operator:



After differencing, the data oscillates around zero, and it has significant autocorrelations of lags 1 and 52. The sample ACF at lags 2 and 53 narrowly exceed the blue dotted line, making it difficult to tell whether the ACF dies off after lag 1. To avoid overfitting, first we try the seasonal ARIMA model of $(0,1,1) \times (0,1,1)_{52}$.



From the correlogram above, we see that the sample ACF at lag 1 is still significant, indicating that this model is unsatisfactory. Thus, we need to increase the order of our model. As a second trial, we look at $ARIMA(1,1,1) \times (0,1,1)_{52}$.



Among the first 120 lags of ACF of the residuals (in the left panel above), about 6 of them reach or exceed the blue dotted line, i.e., the 95% confidence interval. This is a desirable outcome. On the other hand, we may also consider increasing the MA order. That is, we can also try fitting $ARIMA(0,1,2) \times (0,1,1)_{52}$ to our data. Among the first 120 lags of ACF of the residuals (in the right panel above), about 9 of them lie outside the 95% confidence interval. This percentage is slightly larger than 5%, but still acceptable.

2.3 Model Diagnostics and Comparisons

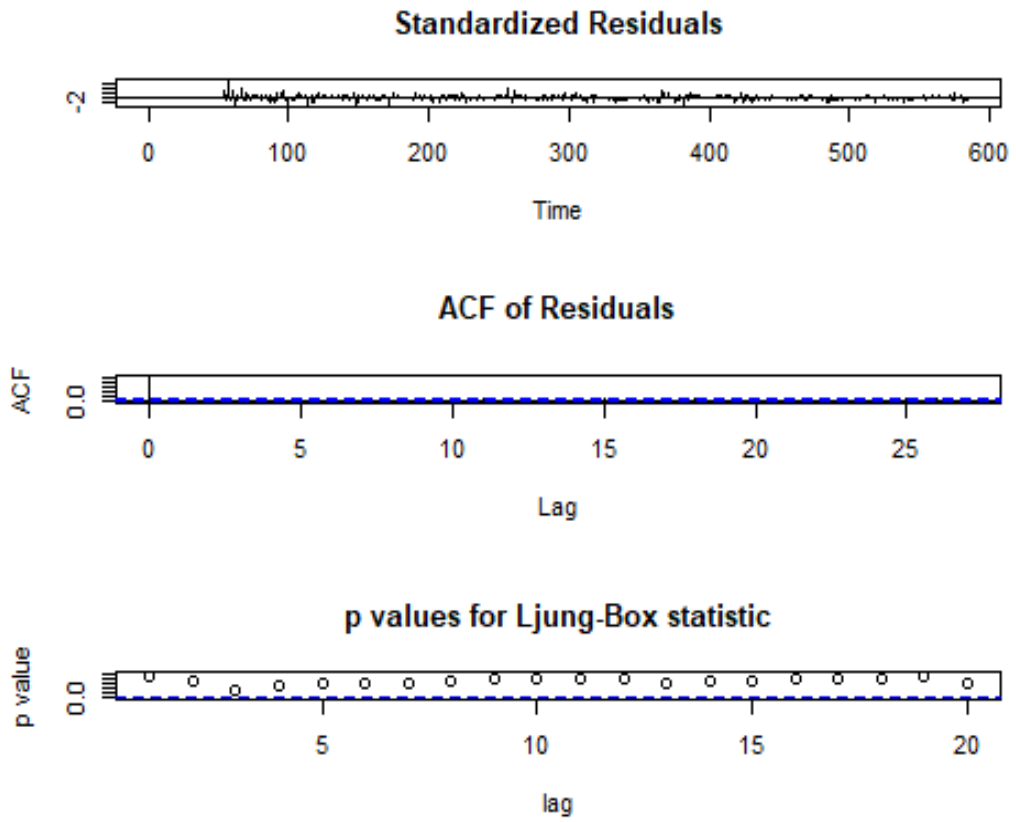
In this part, we will use several methods to determine which model to employ.

2.3.1 Internal Validity

A good model should at least be internally consistent. To check this validity, we can use two different techniques: residual analysis and overfitting.

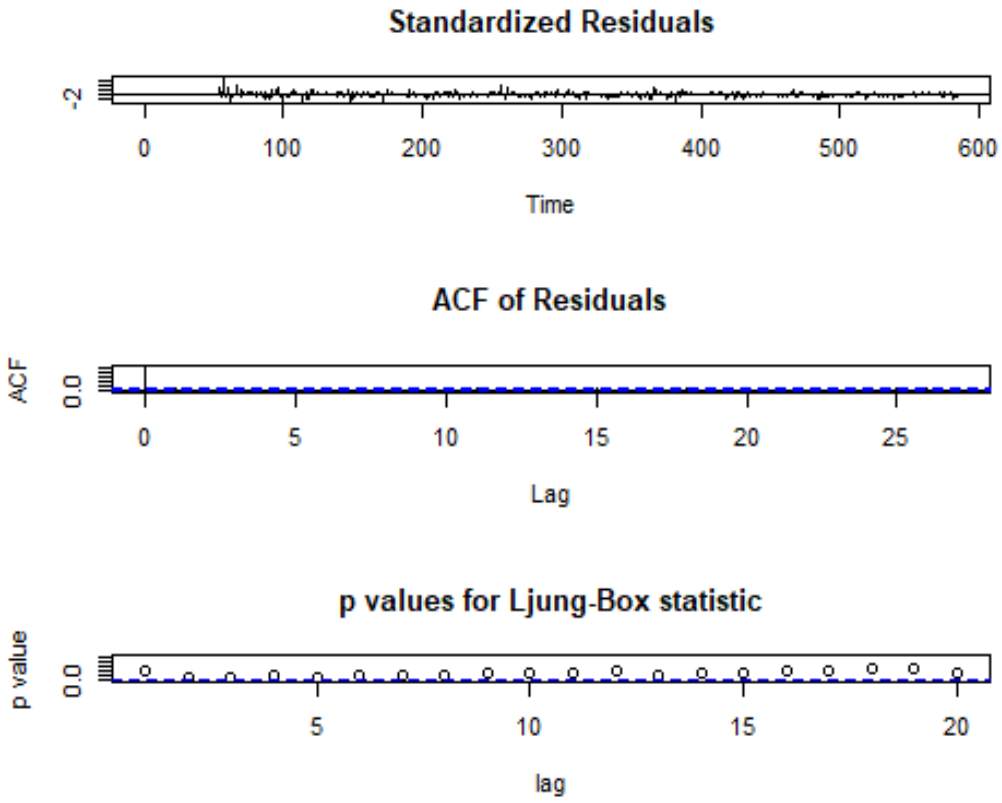
The main idea of residual analysis is that, if a model fits well, the standardized residuals should behave as an i.i.d. sequence with mean zero and variance one. Now,

we check this for $ARIMA(1,1,1) \times (0,1,1)_{52}$. The tool we use is the R function tsdiag().



Recall that we have already inspected the ACF of residuals in the last part, so it remains to interpret the top and bottom panels. The top panel, a time plot of the standardized residuals, displays no obvious patterns. However, there are outliers, with a few values exceeding 3 standard deviations in magnitude. Since their occurrence is quite rare, this should not be very surprising. The bottom panel shows the result of Ljung-Box-Pierce Test, a test which takes into consideration the magnitudes of different lags of the sample ACF of the residuals as a whole group. Since all the points are well above the blue dotted line, we would not reject the null hypothesis that the residuals are white noise.

Now, let's repeat this diagnosis for $ARIMA(0,1,2) \times (0,1,1)_{52}$.



This time, the standardized residuals behave quite similarly as the previous, again with some anomalies, especially at the beginning. We would not be very troubled by these anomalies, because our ultimate goal is to predict the values after the displayed time series, and in that scenario, the values at later years will play a bigger role. The bottom panel shows that the p-values for Ljung-Box statistic is smaller than last time, although not small enough to make us reject the null hypothesis.

Next, we use overfitting to check if the two models are internally valid. The main idea is to increment the order p or q by 1 (not simultaneously), and see whether the new parameter is different from zero, and whether the old parameters change. If the answers to both questions are negative, then our model passes this test.

```
##
## Call:
## arima(x = log.q2, order = c(1, 1, 1), seasonal = list(order = c(0,
1, 1), period = 52))
##
## Coefficients:
##          ar1      ma1    sma1
##       0.3477  -0.8491  -0.327
## s.e.  0.0592   0.0344   0.047
##
## sigma^2 estimated as 0.003339:  log likelihood = 757.15,  aic = -150
6.29
```

```
##
## Call:
## arima(x = log.q2, order = c(1, 1, 2), seasonal = list(order = c(0,
1, 1), period = 52))
##
## Coefficients:
##          ar1      ma1     ma2     sma1
##       0.4889  -1.0026  0.1155  -0.3257
## s.e.  0.1492   0.1595  0.1208   0.0468
##
## sigma^2 estimated as 0.003334:  log likelihood = 757.56,  aic = -150
5.12

##
## Call:
## arima(x = log.q2, order = c(2, 1, 1), seasonal = list(order = c(0,
1, 1), period = 52))
##
## Coefficients:
##          ar1     ar2      ma1     sma1
##       0.3534  0.0430  -0.8655  -0.3261
## s.e.  0.0574  0.0514   0.0363   0.0468
##
## sigma^2 estimated as 0.003335:  log likelihood = 757.5,  aic = -1505
```

From the output above, we see that when we increase p or q by 1 for $ARIMA(1,1,1) \times (0,1,1)_{52}$, the new parameter is less than 2 standard errors from zero, and the old parameters remain stable. For example, when we increase q by 1, for the MA1 coefficients we have

$$|-1.0026-(-0.8491)| = 0.1535 < 2 \times 0.1595.$$

For $ARIMA(0,1,2) \times (0,1,1)_{52}$, however, the output below indicates that this model is bad (note that the new AR1 coefficient is significantly nonzero, and MA1 changes greatly).

```
##
## Call:
## arima(x = log.q2, order = c(0, 1, 2), seasonal = list(order = c(0,
1, 1), period = 52))
##
## Coefficients:
##           ma1      ma2     sma1
##       -0.5190  -0.2117  -0.3259
## s.e.   0.0423   0.0445   0.0469
##
## sigma^2 estimated as 0.003369:  log likelihood = 754.85,  aic = -150
1.7

##
## Call:
```

```
## arima(x = log.q2, order = c(1, 1, 2), seasonal = list(order = c(0,
1, 1), period = 52))
##
## Coefficients:
##          ar1      ma1     ma2     sma1
##       0.4889  -1.0026  0.1155  -0.3257
## s.e.  0.1492   0.1595  0.1208   0.0468
##
## sigma^2 estimated as 0.003334:  log likelihood = 757.56,  aic = -150
5.12
```

2.3.2 Local External Validity

For this part, we use Akaike Information Criterion (AIC) to compare the two models. The last two outputs have already shown that the AIC of $ARIMA(1,1,1) \times (0,1,1)_{52}$ and $ARIMA(0,1,2) \times (0,1,1)_{52}$ is -1506.29 and -1501.7, respectively. Since the former is smaller, this criterion also favors $ARIMA(1,1,1) \times (0,1,1)_{52}$.

2.3.3 General External Validity

Now, we design a validation experiment for the two models under inspection. For each model, first we fit it to the data from $t = 1$ to $t = 52 \times 5$, "predict" the values from $t = 52 \times 5 + 1$ to $t = 52 \times 7$, and calculate the sum squared errors between our "prediction" and true values. Next, fit the model to data from $t = 1$ to $t = 52 \times 7$, "predict" the next 104 data points, and calculate the sum squared errors. Afterwards, fit the model to data from $t = 1$ to $t = 52 \times 9$, and do the similar "prediction" and calculation. Finally, we add up the sum squared errors in the previous three steps. We expect the better model to have a lower value of errors. The result is shown below.

```
## [1] 5.981963

## [1] 6.065664
```

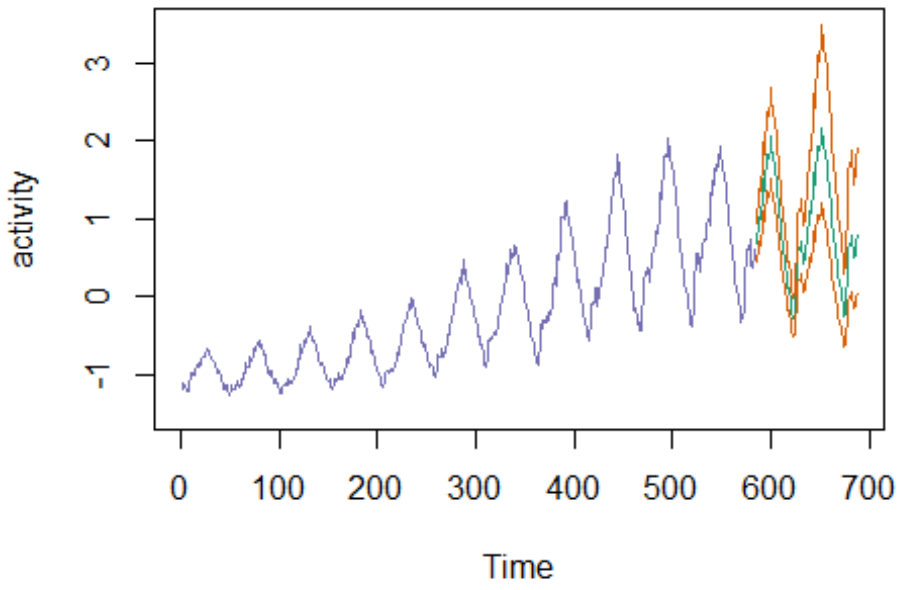
The total sum squared error of $ARIMA(1,1,1) \times (0,1,1)_{52}$ is 5.981963, smaller than that of $ARIMA(0,1,2) \times (0,1,1)_{52}$. Thus, combining the analysis in the last few parts, now we select $ARIMA(1,1,1) \times (0,1,1)_{52}$ to fit our data.

3 Prediction

Having determined the model, we can now use the predict() function to obtain the predicted values and 95% confidence intervals for the next 104 data points, i.e., from $t = 585$ to $t = 688$. For each time point, the upper bound of the confidence interval is calculated by adding the predicted value to 1.96 times the corresponding standard error. However, we must not forget that these are predictions for the transformed data. To get the final results, we need to do the following transformation for each prediction $\hat{X}_t$:

$$\hat{x}_t = exp(\hat{X}_t) - 1.5.$$

In the following graph, the green curve represents our prediction, accompanied by the confidence intervals. We document these values in a separate text file.



Appendix

Code for Dataset 1

```{r}
library(RColorBrewer)
library(ggplot2)
###Read Data
pal <- brewer.pal(8, "Dark2")
Train1 = read.csv('/Users/Rick/Desktop/q1_train.csv', header = T)
```

```{r}
plot(Train1$activity, type = 'l', col = pal)
acf(Train1$activity)
q1.diff0 = diff(Train1$activity)
q1.diff = diff(diff(Train1$activity), 52)
acf(q1.diff0, lag.max = 120)
acf(q1.diff, lag.max = 120)
```

From the plot of the data, we do observe a seasonal trend but hard to say if the trend is positive or negative, therefore, we dicided not to use the log transformation and only work with the original data. We performed diff twice so that the acf finally looks reasonable.

```{r}
model1 = arima(Train1$activity, c(0,1,1), seasonal = list(order = c(0,1,0), period = 52))
acf(model1$residuals, lag.max = 120)
model1
tsdiag(model1, gof.lag = 20)

model2 = arima(Train1$activity, c(1,1,1), seasonal = list(order = c(0,1,0), period = 52))
acf(model2$residuals, lag.max = 120)
model2
tsdiag(model2)
```

Next, we select two models to fit the data and try to briefly pick one by looking at the residual acf, coefficient, and diagnostic plot. We can see the coefficient for Model 2 is pretty useless as the ar1 is really close to 0. So we can briefly know that Model 1 is better. However, briefly is not enough, we want to make sure our assumption by mathmatical method. So we perform Cross Validation on both models

```{r}
##USE Model 1
```

```
fit_1 = arima(Train1$activity[1:(52*5)], c(0,1,1), seasonal = list(order = c(0,1,0), period = 52))
Predict_1 = predict(fit_1, n.ahead= 104)
RSS1 = sum((Predict_1$pred - Train1$activity[(52*5+1):(52*7)])^2)

fit_2 = arima(Train1$activity[1:(52*7)], c(0,1,1), seasonal = list(order = c(0,1,0), period = 52))
Predict_2 = predict(fit_2, n.ahead= 104)
RSS2 = sum((Predict_2$pred - Train1$activity[(52*7+1):(52*9)])^2)

fit_3 = arima(Train1$activity[1:(52*9)], c(0,1,1), seasonal = list(order = c(0,1,0), period = 52))
Predict_3 = predict(fit_3, n.ahead= 104)
RSS3 = sum((Predict_3$pred - Train1$activity[(52*9+1):(52*11)])^2)

M1_RSS = RSS1+RSS2+RSS3

##USE Model 2
M2_fit_1 = arima(Train1$activity[1:(52*5)], c(1,1,1), seasonal = list(order = c(0,1,0), period = 52))
M2_Predict_1 = predict(M2_fit_1, n.ahead= 104)
M2_RSS1 = sum((M2_Predict_1$pred - Train1$activity[(52*5+1):(52*7)])^2)

M2_fit_2 = arima(Train1$activity[1:(52*7)], c(1,1,1), seasonal = list(order = c(0,1,0), period = 52))
M2_Predict_2 = predict(M2_fit_2, n.ahead= 104)
M2_RSS2 = sum((M2_Predict_2$pred - Train1$activity[(52*7+1):(52*9)])^2)

M2_fit_3 = arima(Train1$activity[1:(52*9)], c(1,1,1), seasonal = list(order = c(0,1,0), period = 52))
M2_Predict_3 = predict(M2_fit_3, n.ahead= 104)
M2_RSS3 = sum((M2_Predict_3$pred - Train1$activity[(52*9+1):(52*11)])^2)

M2_RSS = M2_RSS1+M2_RSS2+M2_RSS3

print(c(M1_RSS, M2_RSS))
```

Turns out the RSS of Model 2 is less than Model 1. So we choose Model 2 to do the prediction.

```{r}
pred =predict(model2, n.ahead = 104)
Lower_bound = pred$pred - 1.96 * pred$se
Upper_bound = pred$pred + 1.96 * pred$se

plot(Train1$activity, type = 'l', col = pal[3], xlim = c(1,689), ylim = c(-2,3.5), xlab = 'Index', ylab = 'Activity')
lines(pred$pred, type = 'l', col = pal[1])
lines(Lower_bound, col = pal[2])
lines(Upper_bound, col = pal[2])
```

```

```{r}
result = cbind(Lower_bound, pred$pred, Upper_bound)

write.table(result,'/Users/Rick/Desktop/, Q1_Rick_Chen_3032249208.txt', sep = ',', col.names = F, row.names = F)

```

Code for Dataset 2

``` {r echo=FALSE}
setwd("E:\\Berkeley\\Stat 153\\Midterm 2")
q2_train <- read.csv("q2_train.csv", header=TRUE)
q2 <- as.ts(q2_train[2])
t = 1:584
plot(q2, main = "Original data")
```

``` {r echo=FALSE}
log.q2 <- log(q2 + 1.5)
plot(log.q2, main = "Transformed data")
```

``` {r echo=FALSE}
acf(log.q2, lag.max=120, main="ACF of X_t")
```

``` {r echo=FALSE}
log.q2.diff52_1 <- diff(diff(log.q2,52))
plot(log.q2.diff52_1, main="Differenced data")
acf(log.q2.diff52_1, lag.max=120, main="ACF of differenced data")
```

``` {r echo=FALSE}
q2fit1 <- arima(log.q2, order = c(0,1,1), seasonal = list(order=c(0,1,1), period=52))
acf(q2fit1$residuals, lag.max=120)
# ACF at lag 1 is still significant. Need to increase order.
```

``` {r echo=FALSE}
q2fit2 <- arima(log.q2, order = c(1,1,1), seasonal = list(order=c(0,1,1), period=52))
acf(q2fit2$residuals,lag.max=120)
```
```

```{r echo=FALSE}
q2fit3 <- arima(log.q2, order = c(0,1,2), seasonal = list(order=c(0,1,1), period=52))
acf(q2fit3$residuals,lag.max=120)
```

```{r echo=FALSE}
tsdiag(q2fit2, gof.lag=20)
```

```{r echo=FALSE}
tsdiag(q2fit3, gof.lag=20)
```

```{r echo=FALSE}
plot(q2fit2$residuals - q2fit3$residuals)
```

```{r echo=FALSE}
q2fit4 <- arima(log.q2, order = c(1,1,2), seasonal = list(order=c(0,1,1), period=52))
q2fit5 <- arima(log.q2, order = c(2,1,1), seasonal = list(order=c(0,1,1), period=52))
q2fit2
q2fit4
q2fit5
```

```{r echo=FALSE}
q2fit3
q2fit4
```

```{r echo=FALSE}
q2fit2_1 = arima(log.q2[1:(52*5)], c(1,1,1), seasonal = list(order = c(0,1,1), period = 52))
q2predict2_1 = predict(q2fit2_1, n.ahead= 104)
SE1 = sum((q2predict2_1$pred - log.q2[(52*5+1):(52*7)])^2)

q2fit2_2 = arima(log.q2[1:(52*7)], c(1,1,1), seasonal = list(order = c(0,1,1), period = 52))
q2predict2_2 = predict(q2fit2_2, n.ahead= 104)
SE2 = sum((q2predict2_2$pred - log.q2[(52*7+1):(52*9)])^2)

q2fit2_3 = arima(log.q2[1:(52*9)], c(1,1,1), seasonal = list(order = c(0,1,1), period = 52))
q2predict2_3 = predict(q2fit2_3, n.ahead= 104)
SE3 = sum((q2predict2_3$pred - log.q2[(52*9+1):(52*11)])^2)
```

```
q2fit2_SE = SE1+SE2+SE3

q2fit3_1 = arima(log.q2[1:(52*5)], c(0,1,2), seasonal = list(order = c(0,1,1), period = 52))
q2predict3_1 = predict(q2fit3_1, n.ahead= 104)
se1 = sum((q2predict3_1$pred - log.q2[(52*5+1):(52*7)])^2)

q2fit3_2 = arima(log.q2[1:(52*7)], c(0,1,2), seasonal = list(order = c(0,1,1), period = 52))
q2predict3_2 = predict(q2fit3_2, n.ahead= 104)
se2 = sum((q2predict3_2$pred - log.q2[(52*7+1):(52*9)])^2)

q2fit3_3 = arima(log.q2[1:(52*9)], c(0,1,2), seasonal = list(order = c(0,1,1), period = 52))
q2predict3_3 = predict(q2fit3_3, n.ahead= 104)
se3 = sum((q2predict3_3$pred - log.q2[(52*9+1):(52*11)])^2)

q2fit3_SE = se1+se2+se3

q2fit2_SE
q2fit3_SE
```

```
```{r echo=FALSE}
library(RColorBrewer)
pal <- brewer.pal(8, "Dark2")
q2future =predict(q2fit2, n.ahead = 104)
Lower_bound = q2future$pred - 1.96 * q2future$se
Upper_bound = q2future$pred + 1.96 * q2future$se

q2future_original <- exp(q2future$pred) - 1.5
Lower_bound_original <- exp(Lower_bound) - 1.5
Upper_bound_original <- exp(Upper_bound) - 1.5
plot(q2, type = 'l', col = pal[3], xlim = c(1,688), ylim = c(-1.5,3.5))
lines(q2future_original, type = 'l', col = pal[1])
lines(Lower_bound_original, col = pal[2])
lines(Upper_bound_original, col = pal[2])
```

```
```{r echo=FALSE}
results <- cbind(Lower_bound_original, q2future_original, Upper_bound_original)
write.table(results, "E:\\Berkeley\\Stat 153\\Midterm 2\\Train2.txt", sep=',', col.names = F, row.names = F)
```
```

## Code for Dataset 3
```

```{r}
library(RColorBrewer)
library(ggplot2)
###Read Data
pal <- brewer.pal(8, "Dark2")
Train3 = read.csv('/Users/Rick/Desktop/q3_train.csv', header = T)
```

```{r}
plot(Train3$activity, type = 'l', col = pal)
acf(Train3$activity)
adjusted_data = log(Train3$activity + 1.5)
plot(adjusted_data, type = 'l', col = pal)
acf(adjusted_data)
q1.diff = diff(diff(adjusted_data), 52)
plot(q1.diff, type = 'l', col = pal)
acf(q1.diff, lag.max = 120)
```

We can inspect seasonal trend from the original data and the acf plot has a clearly negative trend, therefore we decide to perform log transformation and difference on the dataset.

```{r}
model0 = arima(adjusted_data , c(1,1,1), seasonal = list(order = c(0,1,1), period = 52))
acf(model0$residuals, lag.max = 120)
model0
tsdiag(model0, gof.lag = 20)

model1 = arima(adjusted_data , c(2,1,1), seasonal = list(order = c(0,1,1), period = 52))
acf(model1$residuals, lag.max = 120)
model1
tsdiag(model1, gof.lag = 20)

model2 = arima(adjusted_data , c(2,1,2), seasonal = list(order = c(0,1,1), period = 52))
acf(model2$residuals, lag.max = 120)
model2
tsdiag(model2)

model3 = arima(adjusted_data , c(2,1,3), seasonal = list(order = c(0,1,1), period = 52))
acf(model3$residuals, lag.max = 120)
model3
tsdiag(model2)

```

Next, we tried different models on the adjusted data. From the diagnostic plots, we can see that the p-value for Model0 is pretty low, and the coefficients for Model 1 and 2 are not sufficient because the ar2 value for both Model 1 and 2 are too close to 0. Therefore, we can almost decide the only Model that is good for prediction is Model 3. However, we want to make sure that our assumption is right, so we use the cross validation.

```{r}
##CV on Model 0
M0_fit_1 = arima(adjusted_data[1:(52*5)] , c(1,1,1), seasonal = list(order = c(0,1,1), period = 52))
M0_Predict_1 = predict(M0_fit_1, n.ahead= 104)
M0_RSS1 = sum((M0_Predict_1$pred - adjusted_data[(52*5+1):(52*7)])^2)

M0_fit_2 = arima(adjusted_data[1:(52*7)], c(1,1,1), seasonal = list(order = c(0,1,1), period = 52))
M0_Predict_2 = predict(M0_fit_2, n.ahead= 104)
M0_RSS2 = sum((M0_Predict_2$pred - adjusted_data[(52*7+1):(52*9)])^2)

M0_fit_3 = arima(adjusted_data[1:(52*9)], c(1,1,1), seasonal = list(order = c(0,1,1), period = 52))
M0_Predict_3 = predict(M0_fit_3, n.ahead= 104)
M0_RSS3 = sum((M0_Predict_3$pred - adjusted_data[(52*9+1):(52*11)])^2)

M0_RSS = M0_RSS1+M0_RSS2+M0_RSS3

##CV on Model 1
M1_fit_1 = arima(adjusted_data[1:(52*5)] , c(2,1,1), seasonal = list(order = c(0,1,1), period = 52))
M1_Predict_1 = predict(M1_fit_1, n.ahead= 104)
M1_RSS1 = sum((M1_Predict_1$pred - adjusted_data[(52*5+1):(52*7)])^2)

M1_fit_2 = arima(adjusted_data[1:(52*7)], c(2,1,1), seasonal = list(order = c(0,1,1), period = 52))
M1_Predict_2 = predict(M1_fit_2, n.ahead= 104)
M1_RSS2 = sum((M1_Predict_2$pred - adjusted_data[(52*7+1):(52*9)])^2)

M1_fit_3 = arima(adjusted_data[1:(52*9)], c(2,1,1), seasonal = list(order = c(0,1,1), period = 52))
M1_Predict_3 = predict(M1_fit_3, n.ahead= 104)
M1_RSS3 = sum((M1_Predict_3$pred - adjusted_data[(52*9+1):(52*11)])^2)

M1_RSS = M1_RSS1+M1_RSS2+M1_RSS3

##CV on Model 2
M2_fit_1 = arima(adjusted_data[1:(52*5)] , c(2,1,2), seasonal = list(order = c(0,1,1), period = 52))
M2_Predict_1 = predict(M2_fit_1, n.ahead= 104)
M2_RSS1 = sum((M2_Predict_1$pred - adjusted_data[(52*5+1):(52*7)])^2)
```

```
M2_fit_2 = arima(adjusted_data[1:(52*7)], c(2,1,2), seasonal = list(order = c(0,1,1), period = 52))
M2_Predict_2 = predict(M2_fit_2, n.ahead= 104)
M2_RSS2 = sum((M2_Predict_2$pred - adjusted_data[(52*7+1):(52*9)])^2)

M2_fit_3 = arima(adjusted_data[1:(52*9)], c(2,1,2), seasonal = list(order = c(0,1,1), period = 52))
M2_Predict_3 = predict(M2_fit_3, n.ahead= 104)
M2_RSS3 = sum((M2_Predict_3$pred - adjusted_data[(52*9+1):(52*11)])^2)

M2_RSS = M2_RSS1+M2_RSS2+M2_RSS3

##USE Model 3
M3_fit_1 = arima(adjusted_data[1:(52*5)] , c(2,1,3), seasonal = list(order = c(0,1,1), period = 52))
M3_Predict_1 = predict(M3_fit_1, n.ahead= 104)
M3_RSS1 = sum((M3_Predict_1$pred - adjusted_data[(52*5+1):(52*7)])^2)

M3_fit_2 = arima(adjusted_data[1:(52*7)], c(2,1,3), seasonal = list(order = c(0,1,1), period = 52))
M3_Predict_2 = predict(M3_fit_2, n.ahead= 104)
M3_RSS2 = sum((M3_Predict_2$pred - adjusted_data[(52*7+1):(52*9)])^2)

M3_fit_3 = arima(adjusted_data[1:(52*9)], c(2,1,3), seasonal = list(order = c(0,1,1), period = 52))
M3_Predict_3 = predict(M3_fit_3, n.ahead= 104)
M3_RSS3 = sum((M3_Predict_3$pred - adjusted_data[(52*9+1):(52*11)])^2)

M3_RSS = M3_RSS1+M3_RSS2+M3_RSS3


print(c(M0_RSS, M1_RSS,M2_RSS,M3_RSS))
```

We can see that Model 3 has the samllest RSS, therefore we choose Model 3 and do the prediction

```{r}
pred =predict(model3, n.ahead = 104)
Lower_bound = pred$pred - 1.96 * pred$se
Upper_bound = pred$pred + 1.96 * pred$se



plot(adjusted_data, type = 'l', col = pal[3], xlim = c(1,689), ylim = c(-1.5,2), xlab = 'Index', ylab = 'Activity')
lines(pred$pred, type = 'l', col = pal[1])
lines(Lower_bound, col = pal[2])
lines(Upper_bound, col = pal[2])
```

```

**Here is the prediction plot of adjusted data, but what we want is the prediction for the original data, so we need to convert the data back to original.(See code below)**

```{r}
pred_original = exp(pred$pred) - 1.5
Lower_original = exp(Lower_bound) - 1.5
Upper_original = exp(Upper_bound) - 1.5
plot(Train3$activity, type = 'l', col = pal[3], xlim = c(1,689), ylim = c(-1.5,5.2), xlab = 'Index', ylab = 'Activity')
lines(pred_original, type = 'l', col = pal[1])
lines(Lower_original, col = pal[2])
lines(Upper_original, col = pal[2])
```

```{r}
result = cbind(Lower_original, pred_original, Upper_original)

write.table(result,'/Users/Rick/Desktop/Q3_Rick_Chen_3032249208.txt', sep = ',', col.names = F, row.names = F)

```

Code for Dataset 4

``` {r}
setwd("E:\\Berkeley\\Stat 153\\Midterm 2")
q4_train <- read.csv("q4_train.csv", header=TRUE)
q4 <- as.ts(q4_train[2])
t = 1:584
log.q4 <- log(q4 + 3)
plot(log.q4, main = "Transformed data")
acf(log.q4, lag.max=120)
```

``` {r echo=FALSE}
log.q4.diff52_1 <- diff(diff(log.q4,52))
plot(log.q4.diff52_1, main="Differenced data")
acf(log.q4.diff52_1, lag.max=120, main="ACF of differenced data")
```

``` {r}

```
q4fit1 <- arima(log.q4, order = c(0,1,1), seasonal = list(order=c(0,1,1), period=52))
acf(q4fit1$residuals, lag.max=120)
```

``` {r}
tsdiag(q4fit1, gof.lag=20)
```

``` {r}
q4fit2 <- arima(log.q4, order = c(0,1,1), seasonal = list(order=c(1,1,0), period=52))
acf(q4fit2$residuals, lag.max=120)
```

``` {r}
tsdiag(q4fit2 ,gof.lag=20)
```

``` {r}
q4fit3 <- arima(log.q4, order = c(0,1,1), seasonal = list(order=c(1,1,1), period=52))
acf(q4fit3$residuals, lag.max=120)
tsdiag(q4fit3, gof.lag=20)
```

``` {r}
q4fit4 <- arima(log.q4, order = c(0,1,1), seasonal = list(order=c(0,1,2), period=52))
acf(q4fit4$residuals, lag.max=120)
tsdiag(q4fit4, gof.lag=20)
# good
```

``` {r}
q4fit5 <- arima(log.q4, order = c(0,1,1), seasonal = list(order=c(2,1,0), period=52))
acf(q4fit5$residuals, lag.max=120)
tsdiag(q4fit5, gof.lag=20)
# so-so
```

``` {r}
q4fit1
q4fit3
q4fit4 # good
```

``` {r}
```

```
q4fit2
q4fit3
q4fit5
```

```{r}
q4fit1_1 = arima(log.q4[1:(52*5)], c(0,1,1), seasonal = list(order = c(0,1,1), period = 52))
q4predict1_1 = predict(q4fit1_1, n.ahead= 104)
SE4_1 = sum((q4predict1_1$pred - log.q4[(52*5+1):(52*7)])^2)

q4fit1_2 = arima(log.q4[1:(52*7)], c(0,1,1), seasonal = list(order = c(0,1,1), period = 52))
q4predict1_2 = predict(q4fit1_2, n.ahead= 104)
SE4_2 = sum((q4predict1_2$pred - log.q4[(52*7+1):(52*9)])^2)

q4fit1_3 = arima(log.q4[1:(52*9)], c(0,1,1), seasonal = list(order = c(0,1,1), period = 52))
q4predict1_3 = predict(q4fit1_3, n.ahead= 104)
SE4_3 = sum((q4predict1_3$pred - log.q4[(52*9+1):(52*11)])^2)

q4fit1_SE = SE4_1+SE4_2+SE4_3

q4fit4_1 = arima(log.q4[1:(52*5)], c(0,1,1), seasonal = list(order = c(0,1,2), period = 52))
q4predict4_1 = predict(q4fit4_1, n.ahead= 104)
se4_1 = sum((q4predict4_1$pred - log.q4[(52*5+1):(52*7)])^2)

q4fit4_2 = arima(log.q4[1:(52*7)], c(0,1,1), seasonal = list(order = c(0,1,2), period = 52))
q4predict4_2 = predict(q4fit4_2, n.ahead= 104)
se4_2 = sum((q4predict4_2$pred - log.q4[(52*7+1):(52*9)])^2)

q4fit4_3 = arima(log.q4[1:(52*9)], c(0,1,1), seasonal = list(order = c(0,1,2), period = 52))
q4predict4_3 = predict(q4fit4_3, n.ahead= 104)
se4_3 = sum((q4predict4_3$pred - log.q4[(52*9+1):(52*11)])^2)

q4fit4_SE = se4_1+se4_2+se4_3

q4fit1_SE
q4fit4_SE
```

```{r}
# The last part finally suggests that we should use q4fit4.
q4future =predict(q4fit4, n.ahead = 104)
q4_Lower_bound = q4future$pred - 1.96 * q4future$se
q4_Upper_bound = q4future$pred + 1.96 * q4future$se
```

```
q4future_original <- exp(q4future$pred) - 3
q4_Lower_bound_original <- exp(q4_Lower_bound) - 3
q4_Upper_bound_original <- exp(q4_Upper_bound) - 3
plot(q4, type = 'l', col = pal[3], xlim = c(1,688), ylim = c(-2,3.5))
lines(q4future_original, type = 'l', col = pal[1])
lines(q4_Lower_bound_original, col = pal[2])
lines(q4_Upper_bound_original, col = pal[2])
```

```
{r}
results_4 <- cbind(q4_Lower_bound_original, q4future_original, q4_Upper_bound_original)
write.table(results_4, "E:\\Berkeley\\Stat 153\\Midterm 2\\Train4.txt", sep=',', col.names = F, row.names = F)
```

Code for Dataset 5

```
{r}
setwd("E:\\Berkeley\\Stat 153\\Midterm 2")
q5_train <- read.csv("q5_train.csv", header=TRUE)
q5 <- as.ts(q5_train[2])
t = 1:584
plot(q5)
acf(q5, lag.max=120)
```

```
{r}
q5_diff1_52 <- diff(diff(q5),52)
acf(q5_diff1_52, lag.max=120)
```

```
{r}
q5fit1 <- arima(q5, order = c(0,1,1), seasonal = list(order=c(0,1,0), period=52))
acf(q5fit1$residuals, lag.max=120)
tsdiag(q5fit1, gof.lag=20)
```

```
{r}
q5fit2 <- arima(q5, order = c(0,1,1), seasonal = list(order=c(0,1,1), period=52))
acf(q5fit2$residuals, lag.max=120)
tsdiag(q5fit2, gof.lag=20)
```

```
{r}
q5fit1
q5fit3 <- arima(q5, order = c(1,1,1), seasonal = list(order=c(0,1,0), period=52))
```

```
q5fit3
q5fit4 <- arima(q5, order = c(0,1,2), seasonal = list(order=c(0,1,0), period=52))
q5fit4
```

```{r}
q5fit1
q5fit2
q5fit5 <- arima(q5, order = c(0,1,1), seasonal = list(order=c(1,1,0), period=52))
q5fit5
```

```{r}
tsdiag(q5fit5, gof.lag = 20)
```

```{r}
q5fit1_1 = arima(q5[1:(52*5)], c(0,1,1), seasonal = list(order = c(0,1,0), period = 52))
q5predict1_1 = predict(q5fit1_1, n.ahead= 104)
SE5_1 = sum((q5predict1_1$pred - q5[(52*5+1):(52*7)])^2)

q5fit1_2 = arima(q5[1:(52*7)], c(0,1,1), seasonal = list(order = c(0,1,0), period = 52))
q5predict1_2 = predict(q5fit1_2, n.ahead= 104)
SE5_2 = sum((q5predict1_2$pred - q5[(52*7+1):(52*9)])^2)

q5fit1_3 = arima(q5[1:(52*9)], c(0,1,1), seasonal = list(order = c(0,1,0), period = 52))
q5predict1_3 = predict(q5fit1_3, n.ahead= 104)
SE5_3 = sum((q5predict1_3$pred - q5[(52*9+1):(52*11)])^2)

q5fit1_SE = SE5_1+SE5_2+SE5_3

q5fit5_1 = arima(q5[1:(52*5)], c(0,1,1), seasonal = list(order = c(1,1,0), period = 52))
q5predict5_1 = predict(q5fit5_1, n.ahead= 104)
se5_1 = sum((q5predict5_1$pred - q5[(52*5+1):(52*7)])^2)

q5fit5_2 = arima(q5[1:(52*7)], c(0,1,1), seasonal = list(order = c(1,1,0), period = 52))
q5predict5_2 = predict(q5fit5_2, n.ahead= 104)
se5_2 = sum((q5predict5_2$pred - q5[(52*7+1):(52*9)])^2)

q5fit5_3 = arima(q5[1:(52*9)], c(0,1,1), seasonal = list(order = c(1,1,0), period = 52))
q5predict5_3 = predict(q5fit5_3, n.ahead= 104)
se5_3 = sum((q5predict5_3$pred - q5[(52*9+1):(52*11)])^2)

q5fit5_SE = se5_1+se5_2+se5_3
```

```
q5fit1_SE
q5fit5_SE
```

```{r}
q5future =predict(q5fit5, n.ahead = 104)
q5_Lower_bound = q5future$pred - 1.96 * q5future$se
q5_Upper_bound = q5future$pred + 1.96 * q5future$se
plot(q5, type = 'l', col = pal[3], xlim = c(1,688), ylim = c(-3,4))
lines(q5future$pred, type = 'l', col = pal[1])
lines(q5_Lower_bound, col = pal[2])
lines(q5_Upper_bound, col = pal[2])
```

```{r}
results_5 <- cbind(q5_Lower_bound, q5future$pred, q5_Upper_bound)
write.table(results_5, "E:\\Berkeley\\Stat 153\\Midterm 2\\Train5.txt", sep=',', col.names = F, row.names = F)
```