NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

AI6123 Time Series Analysis Assignment 1

Liu Yaohong G2203319C

Liuy0220@e.ntu.edu.sg

The wwwusage time series data consist of the number of users connected to the internet through a server. The data are collected at a time interval of one minute and there are 100 observations. Please fit an appropriate ARIMA model for it and submit a short report including R codes, the fitted model, the diagnostic checking, AIC, etc

I. RAW DATA ANALYSIS

The target of this project is finding an appropriate ARIMA model to fit the 'wwwusage' dataset. Before we start to create a ARIMA model, we have to analyst the raw data. First of all, we need to plot the raw data (Fig.1) and calculate the statistical information of the raw data, such as minimum, maximum, mean, variance.

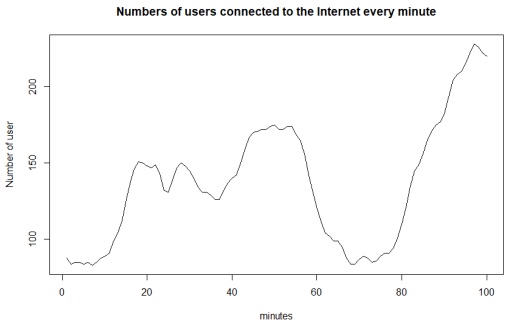

Fig.1. Raw Data Plot

From raw data set, we get the maximum and minimum value are 228 and 83. After plotting the raw data, we can see that the data falling on interval 0 to 60 has different observation comparing with other time period 60 to100. We calculate the mean of each period. The means of the 2 different time period are 135.1 and 143.902, which is different. After 70 time lag, the data seams trending up overall.

To further prove our assumption that data is not stationary, we can plot the ACF (Fig.2) and PACF(Fig.3) of the raw data. The ACF data is never cutting off or dying down. Even though PACF is cut off on lag 2, we can still say the data is non-stationary.

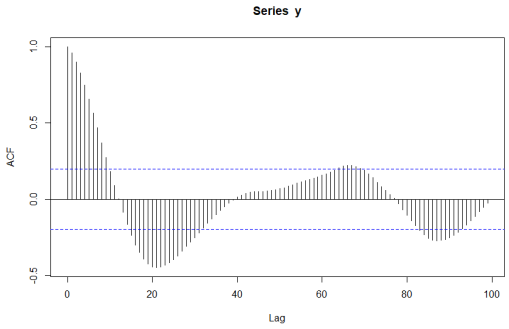

Fig.2. Raw Data ACF

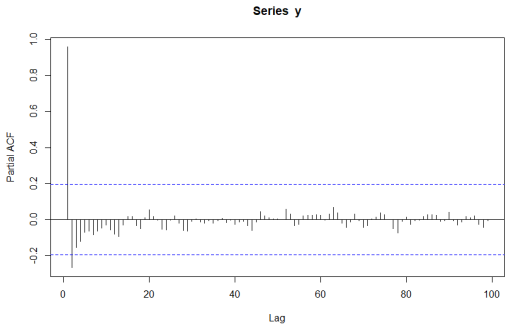

Fig.3. Raw Data PACF

To fit an appropriate AMRIMA model, we have to apply time differencing to make our data stationary.

II. FIRST-ORDER DIFFERENCING ANALYSIS

To remove the trending component of the data, we apply the first order differencing to the raw data and plot the data (Fig.4), ACF(Fig.5), PACF(Fig.6) respectively.

From Fig.4, we can see the data is not obvious trending up or down. And the

ACF is cutting off on lag 24, PACF is cut off on lag3, which means AR(3) model is better solution. Until now, we have applied 1 time differencing and AR(3) model. Therefore, the original time series data is fitted using ARIMA(3,1,0).

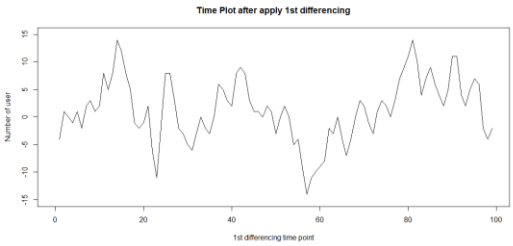

Fig.4. 1st Order Differencing Data Plot

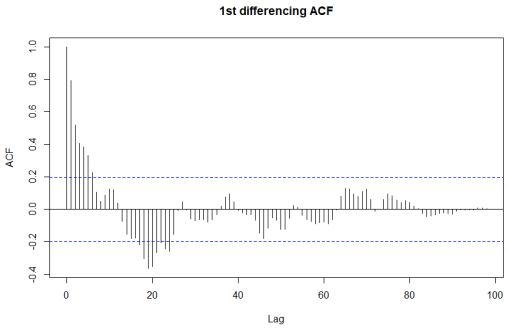

Fig.5. 1st Order Differencing ACF

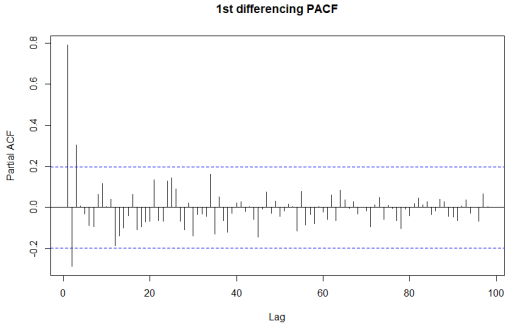

Fig.6. 1st Order Differencing PACF

Now, we have to check whether our model is adequate by using tsdiag(), shown on Fig 7.

As we can see, the values of standardized residual are randomly distributed around 0 with no discernible and trend. ACF of residuals is cut off on 0 lag which is good. Lastly, the p-values for Ljung-Box statistic are larger than 0.05, which means we can not reject the adequacy of the model by setting $\alpha = 0.05$ or we can not reject the null hypothesis $H_0 : \rho_e(k) = 0$.

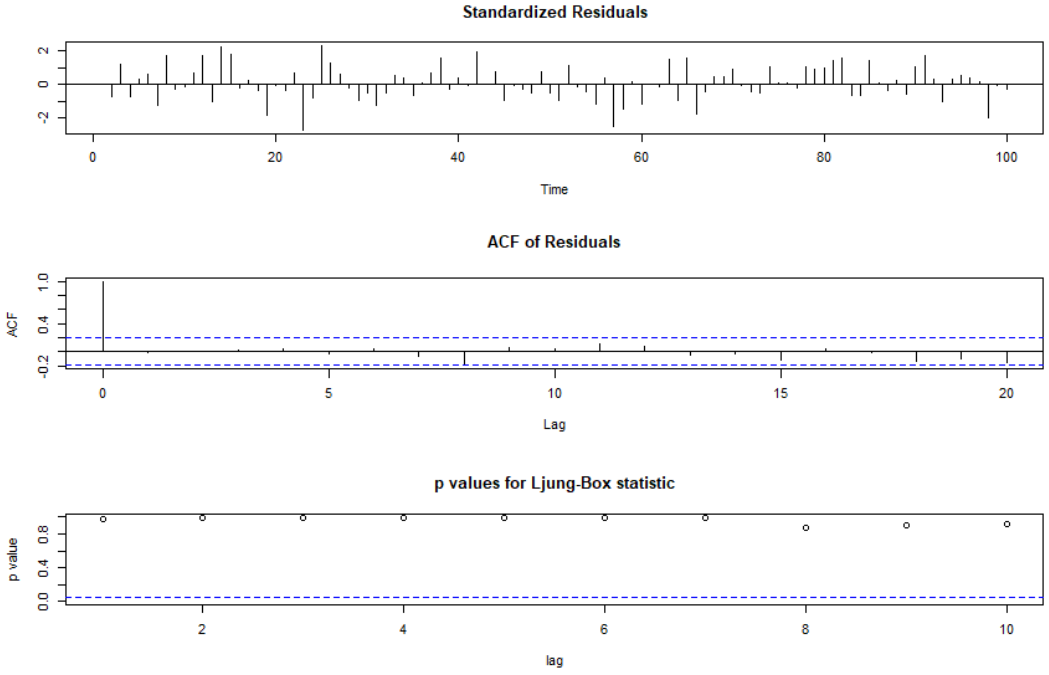

Fig.7. Diagnostic Check of ARIMA(3,1,0)

Now, let's split 10% of the data set as validation set and 90% data as training set. And we can fit the ARIMA(3,1,0) model to verify the prediction, shown on Fig.8. In our case, we used first 90 data points as training data, the last 10 data point as test set.

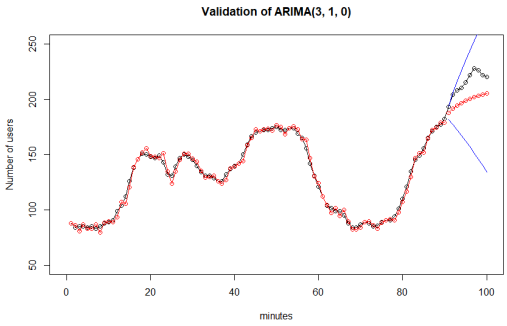

Fig.8. Black dot is raw data and red dot is ARIMA(3,1,0) prediction

Figure.8 displays the raw data as black dots and the ARIMA(3,1,0) predictions as red dots. The blue lines represent the 95% confidence interval of our prediction falling within that area. Upon comparison with the ground truth data, the prediction in the test set did not perform very well and was not sufficiently fitting.

III. SECOND-ORDER DIFFERENCING ANALYSIS

At present, we are unable to determine the appropriate differencing order. As a next step, we can try applying second-order differencing, similar to the first-order differencing, and examine the resulting data. The 2nd order differencing data plotting, ACF, PACF shown on Fig.9, Fig.10 and Fig.11 respectively.

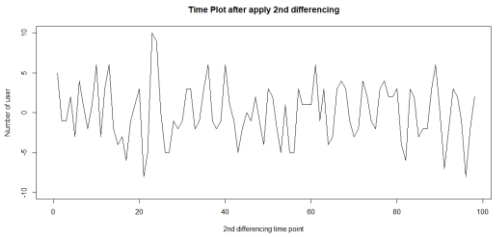

Fig.9. 2nd Order Differencing Data Plot

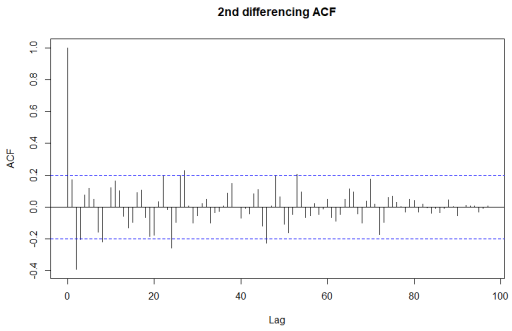

Fig.10. 2nd Order Differencing ACF

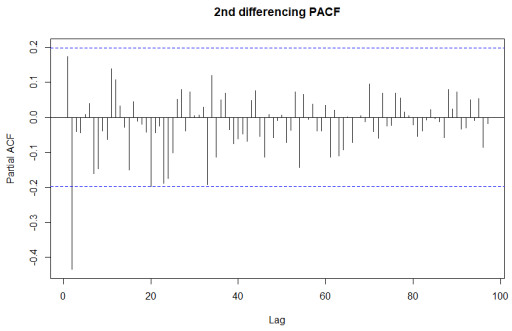

Fig.11. 2nd Order Differencing ACF

After applied 2nd order differencing, we can clearly see the mean of data is around 0 as a constant. The ACF plot shows data ACF cut off on lag 25 and PACF plot shows the data PACF cut off on lag 2. Thus, we can use ARIMA(2,2,0) model.

To test the model, we can apply tsdiag() function again. As we can see from Fig.12, the values of standardized residual are randomly distributed around 0 with no discernible and trend. ACF of residuals is cut off on 0 lag which is good. Lastly, the p-values for Ljung-Box statistic are larger than 0.05, which means we cannot reject the adequacy of the model by setting $\alpha = 0.05$.

To validate ARIMA(2,2,0) model, we used first 90 data points as training data, the last 10 data point as test set.

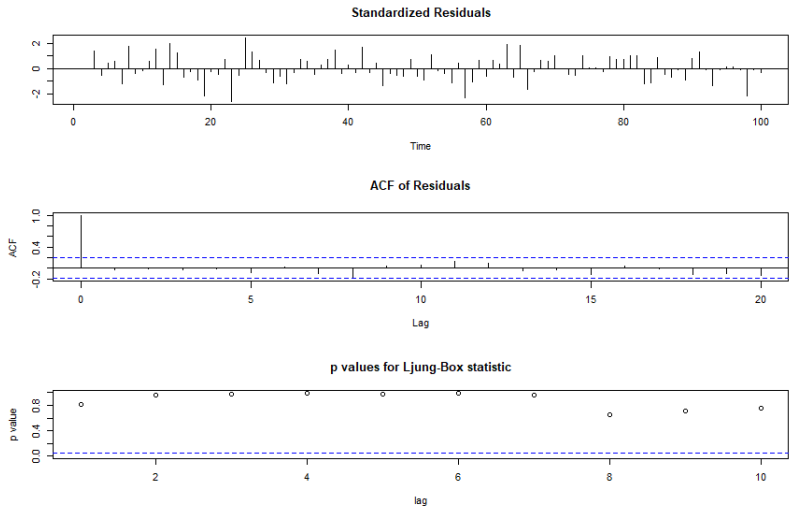

Fig.12. Diagnostic Check of ARIMA(2,2,0)

It is evident from Figure 13 that the next 10-step prediction is trending upward. However, the last three time points of the raw data (black dots) indicate a downward trend. Unfortunately, our ARIMA(2,2,0) model's prediction values (red dots) continue to rise, which is not indicative of good performance.

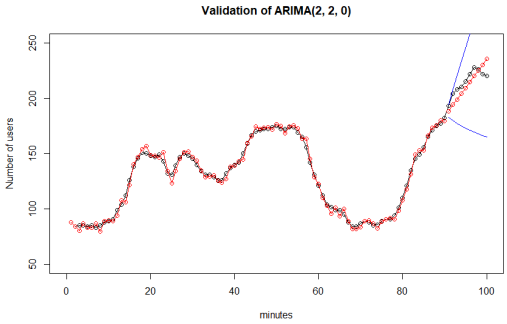

Fig.13. Black dot is raw data and red dot is ARIMA(2,2,0) prediction

IV. SEQUENTIAL SEARCH

Fig.14 indicates that the standardized residuals exhibit a random distribution around 0, without any discernible trend. Furthermore, the ACF plot of the residuals shows cutoff at 0 lag, which is a positive indication. Finally, the p-values for the Ljung-Box statistic exceed 0.05, indicating that we cannot reject the adequacy of the model at α=0.05.

To validate the ARIMA(5,2,5) model, the first 90 data points were utilized as training data, while the last 10 data points were used as a test set.

p \ q	0	1	2	3	4	5	6
0	NA	549.8055	519.8749	520.2717	519.3800	518.8573	518.2574
1	529.2378	514.2995	516.2519	514.5763	515.1001	516.2762	517.9788
2	522.1782	516.2914	517.3604	515.7733	513.2413	518.0892	520.0533
3	511.9940	513.9377	515.6208	514.4139	514.7583	516.4277	514.1192
4	513.9298	515.9558	516.1818	519.0777	515.3952	NA	517.2763
5	515.8617	517.6386	513.5433	521.6405	**511.1393**	512.7706	514.7702
6	517.4507	519.6317	518.5071	NA	NA	NA	NA

Table.1. AIC values of d equals to 1

p \ q	0	1	2	3	4	5	6
0	NA	523.9024	517.2141	512.3330	513.8051	515.7849	517.3000
1	530.4436	523.5921	513.1887	513.8513	515.7958	513.8984	515.1896
2	511.4645	513.2557	515.1303	515.7331	513.8702	514.6680	516.9475
3	513.2912	510.7123	512.6767	514.4996	NA	513.5214	515.7518
4	515.1390	517.1386	514.5294	515.0273	517.8548	519.9093	517.0725
5	517.1372	518.7936	516.3985	518.3985	514.1490	**509.8135**	511.6112
6	518.7514	520.0481	518.5307	516.4733	516.0612	512.4001	512.1558

Table.2. AIC values of d equals to 2

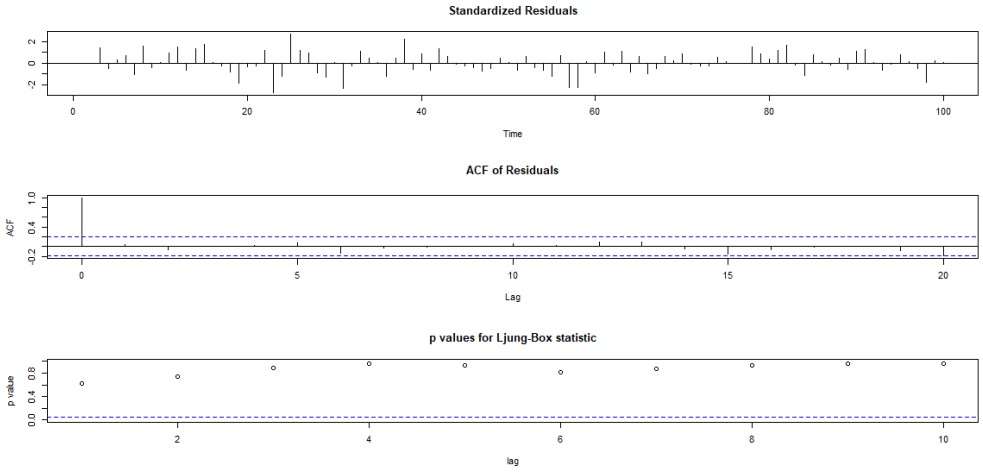

Fig.14. Diagnostic Check of ARIMA(5,2,5)

From Section III, we know the PACF cuts off on leg 2 after second order differencing. Now, we use tsidiag() function to check ARIMA(5,2,5) model, shown on Fig.14.

Fig.14 indicates that the standardized residuals exhibit a random distribution around 0, without any discernible trend. Furthermore, the ACF plot of the residuals shows cutoff at 0 lag, which is a positive indication. Finally, the p-values for the Ljung-Box statistic exceed 0.05, indicating that we cannot reject the adequacy of the model at α=0.05.

To validate the ARIMA(5,2,5) model, the first 90 data points were utilized as training data, while the last 10 data points were used as a test set.

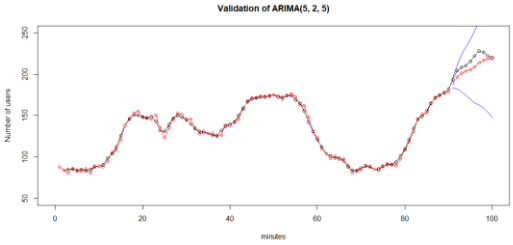

Fig.15. Black dot is raw data and red dot is ARIMA(5,2,5) prediction

On Fig.15, the slot for our prediction is gradually decreasing and approaches zero at the final data point. The ground truth data also decreases during the last three data points, indicating that our model's prediction aligns with the actual trend of the data. From the prediction, ARIMA(5,2,5) model has better performing than ARIMA(3,1,0) and ARIMA(3,1,0).

V. AUTO ARIMA BY R

This section we discuss the auto ARIMA fitting provide by R. By using auto.arima() function, we get $p = 1$, $d = 1, q = 1$, which is ARIMA(1,1,1) model. The diagnostic checking is shown on Fig.16.

Figure 16 demonstrates that the standardized residuals are randomly distributed around zero, without any apparent pattern. Additionally, the ACF plot of the residuals exhibits a cutoff at 0 lag, which is a positive indication. The p-values for the Ljung-Box statistic are greater than 0.05, indicating that we cannot reject the model's adequacy at α=0.05.

To confirm the validity of the ARIMA(1,1,1) model, we used the first 90 data points as training data and the last 10 data points as a test set.

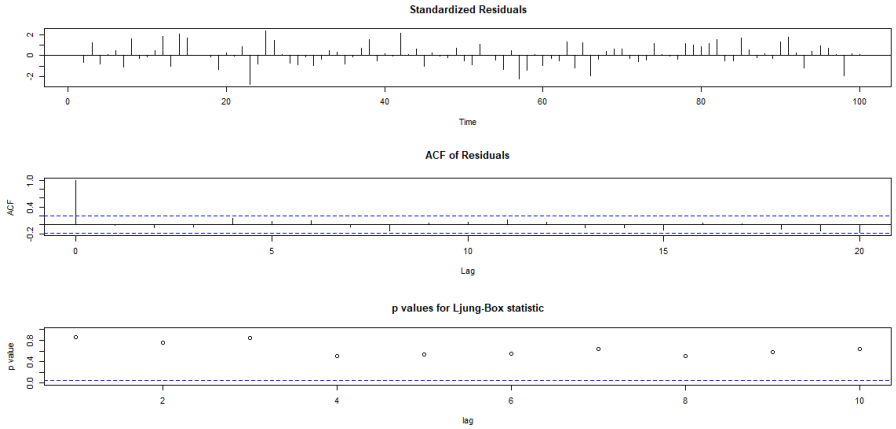

Fig.17. Diagnostic Check of Auto ARIMA

The plot of the validation data set demonstrates that the Auto ARIMA model is effectively capturing the features of the training data set. However, the prediction begins to trend downward at data point 93, while our ground truth data also trends downward during the last three data points. Comparing the Auto ARIMA model's prediction with the ground truth data reveals that it is not a good prediction.

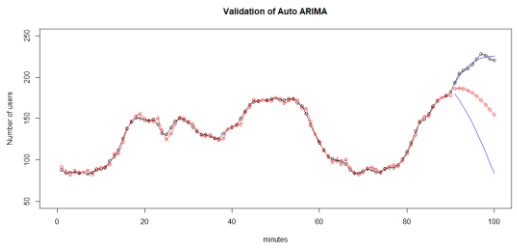

Fig.18. Black dot is raw data and red dot is ARIMA(1,1,1) prediction

VI. CONCLUSION

Until now, we have created 4 models for 'wwwusage' data set. The AICs and RMSE of 4 model are shown on table.3.

	ARIMA(3,1,0)	ARIMA(2,2,0)	ARIMA(5,2,5)	AutoARIMA
AIC	511.944	511.4645	**509.8135**	514.2995
RMSE	17.411	**8.384**	21.360	43.966

Table.3. AICs of ARIAM model

Although ARIMA(2,2,0) has the lowest RMSE, it failed to capture the downward trend of the raw data during the last three data points, as our investigation revealed. On the other hand, the ARIMA(5,2,5) model accurately predicted the downward trend of the raw data and had the lowest AIC value. Therefore, we conclude that the ARIMA(5,2,5) model is the best choice for the 'wwwusage' data set.

VII. APPENDIX

```
library(forecast)
library(tseries)

location="C:/Users/liu92/Documents/MSAI/TimeSeries/Assignment1/wwwusage.txt"
y <- scan(location, skip = 1)
max(y)
min(y)
mean(y)
```

```
acf(y, lag.max = 100)
pacf(y, lag.max = 100)
ts.plot(y, gpars=list(xlab="minutes"
                      ,ylab="Number of user"
                      ,main='Numbers of users connected to the Internet every
minute')

##D1
d1<-diff(y)
max(d1)
min(d1)

ts.plot(d1,xlim = c(0,99), ylim=c(-15, 15)
        ,gpars=list(xlab="1st differencing time point", ylab="Number of user"
                    ,main='Time Plot after apply 1st differencing'))
acf(d1, lag.max = 99,main='1st differencing ACF')
pacf(d1, lag.max = 99,main='1st differencing PACF')
ar.yw(d1, order.max = 10)
fit_d1 <- arima(y, order= c(3,1,0))
tsdiag(fit_d1)

trainSet <- y[1:90]
testSet <- y[91:100]
fit_d1_train = arima(trainSet, order= c(3,1,0))
forecast1 = predict(fit_d1_train, n.ahead=10)


plot(c(y),
     main = "Validation of ARIMA(3, 1, 0)",
     xlab = "minutes",
     ylab = "Number of users",
     type = "o",
     xlim = c(0,100),
     ylim = c(50,250))
lines(1:90, trainSet-fit_d1_train$residuals, type="o", col="red")
lines(91:100, forecast1$pred, type="o", col="red")
lines(91:100, forecast1$pred-1.96*forecast$se, col="blue")
lines(91:100, forecast1$pred+1.96*forecast$se, col="blue")

##D2
d2<-diff(d1)
max(d2)
min(d2)
```

```
ts.plot(d2,xlim = c(0,98), ylim=c(-10, 10)
              ,gpars=list(xlab="2nd differencing time point", ylab="Number of user"
                                ,main='Time Plot after apply 2nd differencing'))
acf(d2, lag.max = 98,main='2nd differencing ACF')
pacf(d2, lag.max = 98,main='2nd differencing PACF')
fit_d2 <- arima(y, order= c(2,2,0))

fit_d2_train = arima(trainSet, order= c(2,2,0))
forecast2 = predict(fit_d2_train, n.ahead=10)

plot(c(y),
        main = "Validation of ARIMA(2, 2, 0)",
        xlab = "minutes",
        ylab = "Number of users",
        type = "o",
        xlim = c(0,100),
        ylim = c(50,250))
lines(1:90, trainSet-fit_d2_train$residuals, type="o", col="red")
lines(91:100, forecast2$pred, type="o", col="red")
lines(91:100, forecast2$pred-1.96*forecast$se, col="blue")
lines(91:100, forecast2$pred+1.96*forecast$se, col="blue")


#sequential test
aicd1 <- matrix(NA, 7, 7, dimnames = list(p = 0:6, q = 0:6))
aicd2 <- matrix(NA, 7, 7, dimnames = list(p = 0:6, q = 0:6))
for(p in 0:6){
   for(q in 0:6){
      tryCatch({
         if(!(p == 0 & q == 0)){
            aicd1[p+1, q+1] <-AIC(arima(y, c(p, 1, q)))}
      }, error = function(e) {
         print(paste("An error occurred:", e$message))
         NaN
      }, warning = function(w) {
         print(paste("A warning occurred:", w$message))
         NaN
      }, finally = {
         print("Done.")
      })
   }}
for(p in 0:6){
   for(q in 0:6){
```

```
      tryCatch({
        if(!(p == 0 & q == 0)){
          aicd2[p+1, q+1] <-AIC(arima(y, c(p, 2, q)))}
      }, error = function(e) {
        print(paste("An error occurred:", e$message))
        NaN
      }, warning = function(w) {
        print(paste("A warning occurred:", w$message))
        NaN
      }, finally = {
        print("Done.")
      })
  }}



min(aics_d1, na.rm = TRUE)#511.1393
min(aics_d2, na.rm = TRUE)#509.8135
arima525= arima(y, order= c(5,2,5))
tsdiag(arima525)

arima525_train = arima(trainSet, order= c(5,2,5))
forecast3 = predict(arima525_train, n.ahead=10)

plot(c(y),
       main = "Validation of ARIMA(5, 2, 5)",
       xlab = "minutes",
       ylab = "Number of users",
       type = "o",
       xlim = c(0,100),
       ylim = c(50,250))
lines(1:90, trainSet-arima525_train$residuals, type="o", col="red")
lines(91:100, forecast3$pred, type="o", col="red")
lines(91:100, forecast3$pred-1.96*forecast$se, col="blue")
lines(91:100, forecast3$pred+1.96*forecast$se, col="blue")


##auto fit arima
autoarima <- auto.arima(y)
autoarima
autoarima_train = auto.arima(trainSet)
forecast4 = predict(autoarima_train, n.ahead=10)

plot(c(y),
```

```
        main = "Validation of ARIMA(1, 1, 1)",
        xlab = "minutes",
        ylab = "Number of users",
        type = "o",
        xlim = c(0,100),
        ylim = c(50,250))
lines(1:90, trainSet-autoarima_train$residuals, type="o", col="red")
lines(91:100, forecast4$pred, type="o", col="red")
lines(91:100, forecast4$pred-1.96*forecast$se, col="blue")
lines(91:100, forecast4$pred+1.96*forecast$se, col="blue")

##AIC Checking, Accuracy checking
AIC(fit_d1)
AIC(fit_d2)
AIC(arima525)
AIC(autoarima)
accuracy(testSet,forecast1$pred)
accuracy(testSet,forecast2$pred)
accuracy(testSet,forecast3$pred)
accuracy(testSet,forecast4$pred)
```