The assignment aims to fit an appropriate ARIMA model for the 'wwwusage' time series data which consists of the number of users connected to the internet through a server.

The data

Mean of Data

```
> y=scan("C:/Users/mingy/OneDrive/Documents/AI6123/wwwusage.txt",skip=1)

> plot(1:100, y, xlim = c(0, 100), ylim = c(0, 250), xlab = "Observation
number", ylab = "Number of users connected to the internet ")

> lines(1:100, y, type="l" )
```

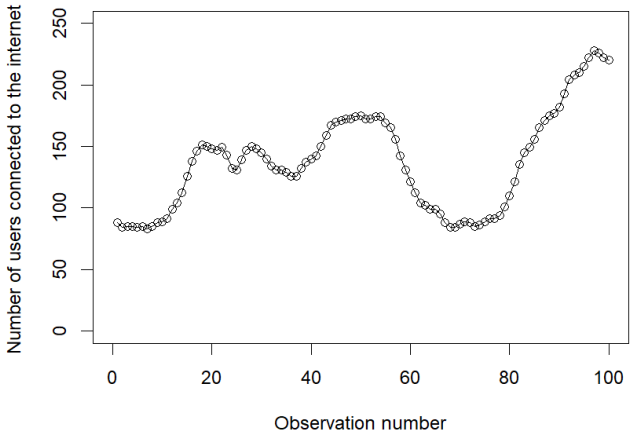

Fig 1: Plot of number of users connected to the internet against the observation number

From fig 1, we see that the mean number of users connected to the internet through a server varies drastically over the span of 100 minutes. For example, the mean number of users seemed to be around 80 from 0th to 10th minute, 150 from 10th to 55th minute, 80 from 60th to 80th minute and has an increasing trend of 80 to 220 from 80th to 100th minute. Looking at the data, it does not show any periodic trend and did not give any clear indication that the variance of the data is changing over time. Thus, box cox transformation will not be used to transform this data.

Sample ACF

```
> acf(y, lag.max=100)

> acf(y)
```

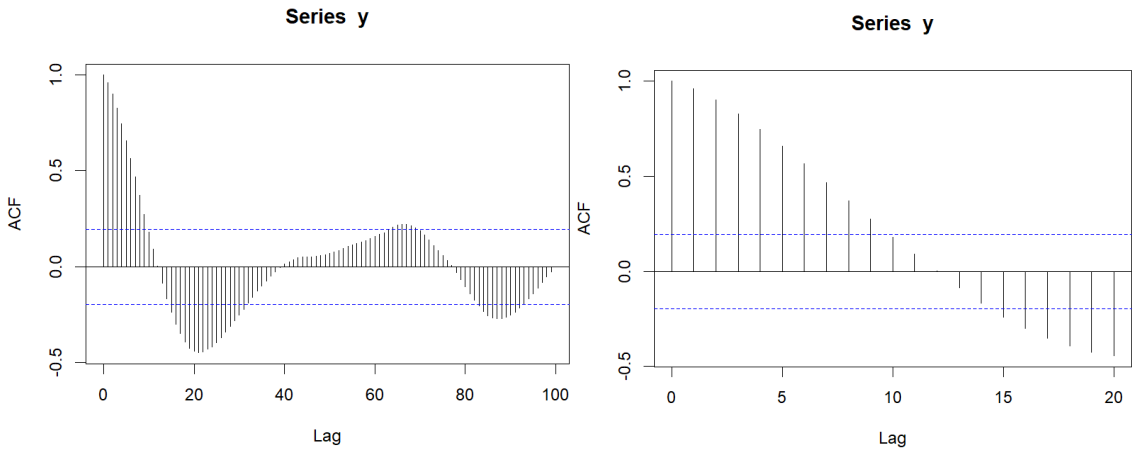

Fig 2: Sample ACF plot of the dataset

Looking at fig 2, the sample ACF is cut off at around time lag 93. That said, the boundary for considering sample ACF to be negligible is bigger as the time lag increases and it is theoretically not a straight blue line as shown in fig 2. The below is the criteria for sample ACF to be negligible:

$$|r_h| \leq 1.96\left(\frac{1 + 2\sum_{j=1}^{h} r_j^2}{n}\right)^{\frac{1}{2}} - - - equation\ 1$$

where r_h is the sample ACF of the data at time lag h.

The below is the criteria used in R Studio for ACF to be negligible:

$$|r_{hh}| \leq 1.96\left(\frac{1}{n}\right)^{\frac{1}{2}} - - - equation\ 2$$

Hence, as the time lag increases, the boundary is supposed to expand as per equation 1 instead of staying as a horizontal line as per equation 2. With this, it is possible to conclude that the sample ACF cut off at around time lag 30 or 50. It is noted that that the sample ACF die down extremely slowly. It is also possible to conclude that the sample ACF cut off at time lag around 9 as time lag 30 is far from time lag 1 and is not prominent. The covariance of X_t and X_{t+22} is very small.

Sample PACF

```
> pacf(y, lag.max=100)
```

```
> pacf(y)
```

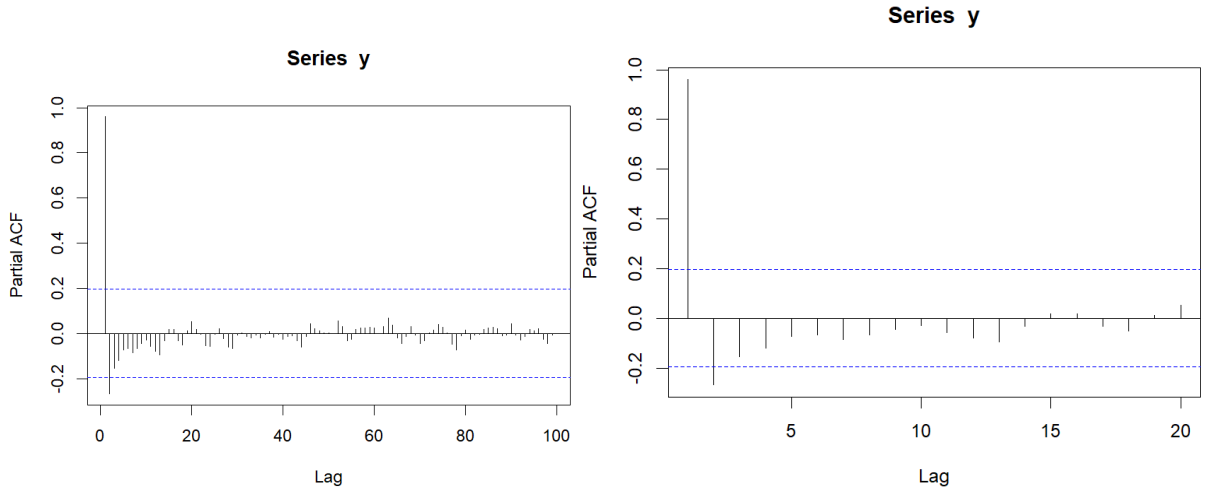

Fig 3: Sample PACF plot of the dataset

The sample PACF cut off at around time lag 1. With the sample ACF as in fig 2 dying down extremely slow and the mean seemed to change as per fig 1, it is possible that this is a non-stationary model.

Changing to a non-stationary model

We propose adjacent differencing to be conducted to the non-stationary model and not the seasonal differencing since the observations are not taken over a period of 1 year or shown patterns repeating over a yearly basis. In this report, we will conduct 1 time differencing and 2 time differencing to get more variety of proposed possible models to fit the data.

1 time differencing

Equation for models with 1 time differencing: $z_t = y_t - y_{t-1} = (B-1)y_t$

```
> dy=diff(y,lag=1,differences=1)
```

```
> plot(1:99, dy, xlim = c(0, 100), ylim = c(-15, 15), xlab = "Observation
number", ylab = "Users connected to the internet after time differencing")
```

```
> lines(1:99, dy, type="l" )
```

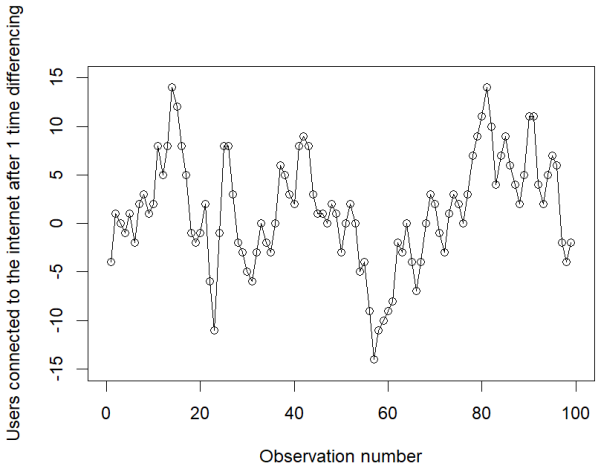

Fig 4: Data after 1 time differencing

```
> var(dy)
[1] 32.18367
```

Sample ACF (1 time differencing)

```
> acf(dy,lag.max=99)
```

```
> acf(dy)
```

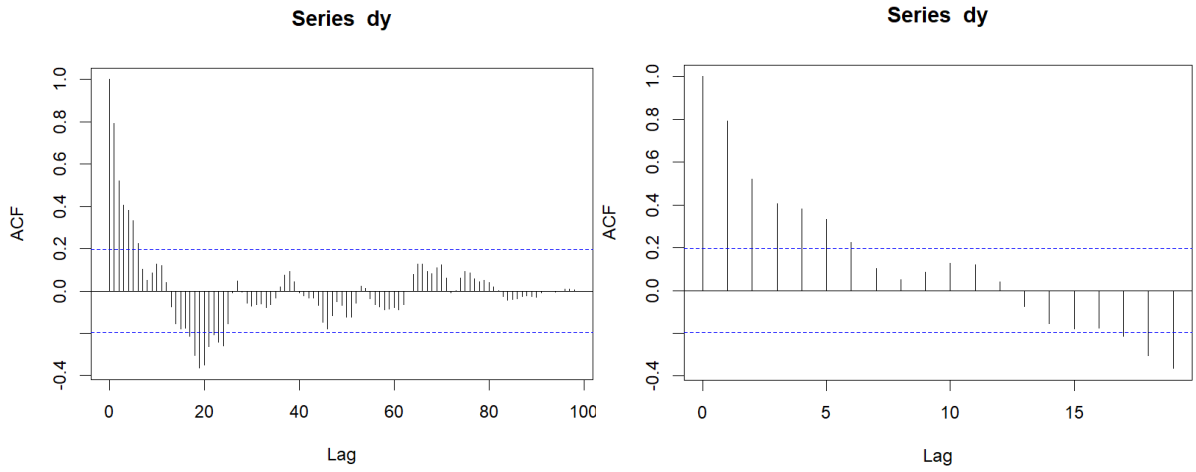

Fig 5: Sample ACF after 1 time differencing

The sample ACF cut off at around time lag 6. We disregard the sample ACF exceeding the blue horizontal threshold at time lag 17 to 24 because as mentioned above, the boundary for considering sample ACF to be negligible is bigger as the time lag increases as per equation 1. Hence, the sample ACF at time lag 17 to 24 might possibly be negligible. Moreover, the time lag around 17 to 24 is far from time lag 1 and is not prominent. The covariance of X_t and X_{t+17} or X_{t+24} is very small. A MA(6) model is a possible option.

Sample PACF (1 time differencing)

```
> pacf(dy,lag.max=99)
```

```
> pacf(dy)
```

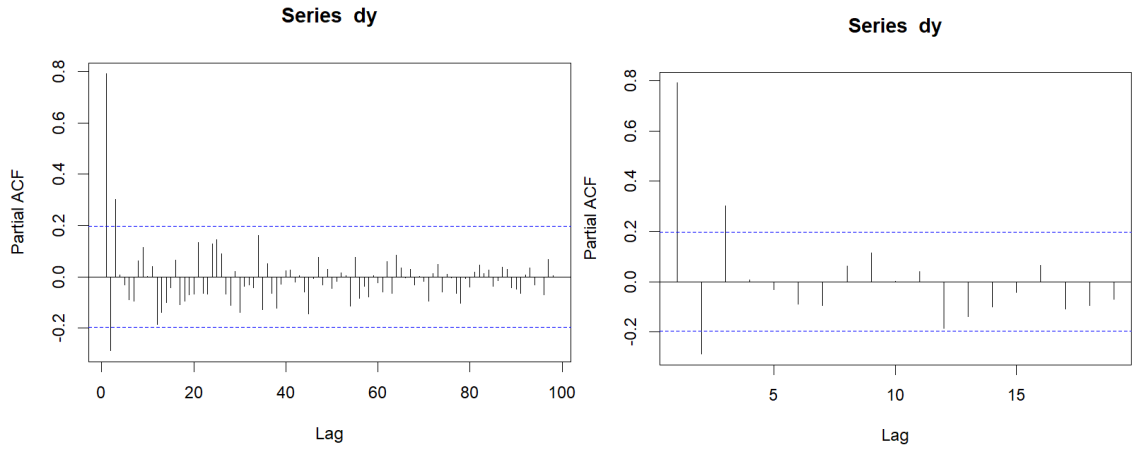

Fig 6: Sample PACF after 1 time difference

The sample PACF cut off at time lag 3. An option model might be AR(3).

2 time differencing

Equation for models with 2 time differencing: $z_t = y_t - y_{t-2} = (1 - B^2)y_t$

```
> ddy=diff(y,lag=1,differences=2)
```

```
> plot(1:98, ddy, xlim = c(0, 100), ylim = c(-15, 15), xlab = "Observation
number", ylab = "Users connected to the internet after 2 time differencing")
```

```
> lines(1:98, ddy, type="l" )
```

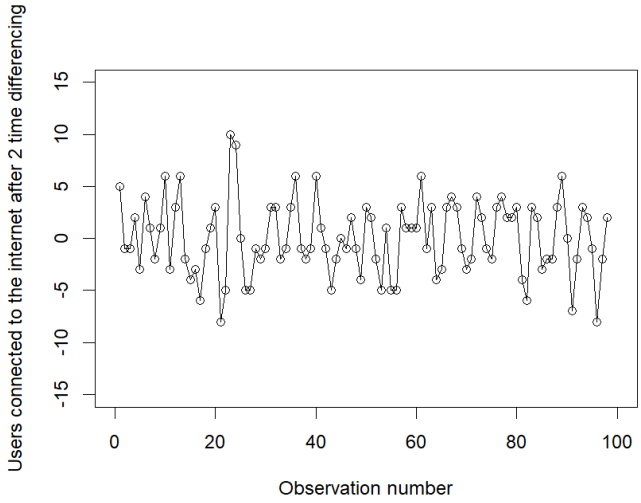

Fig 7: Data after 2 time differencing

```
> var(ddy)
13.1336
```

Sample ACF (2 time differencing)

```
> acf(ddy,lag.max=98)
> acf(ddy)
```

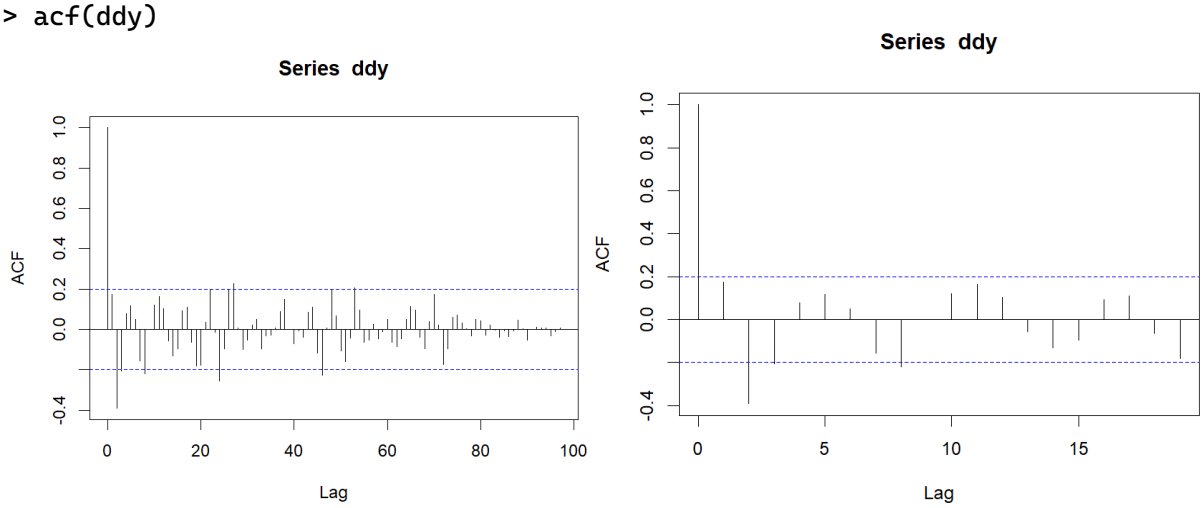

Fig 8: Sample ACF after 2 time differencing

The sample ACF cut off at around time lag 8. We disregard the sample ACF exceeding the blue horizontal threshold at time lag 24 onwards because as mentioned above, the boundary for considering sample ACF to be negligible is bigger as the time lag increases as per equation 1. Hence, the sample ACF at time lag 24 onwards might possibly be negligible. Moreover, the time lag around 24 is far from time lag 1 and is not prominent. The covariance of X_t and X_{t+24} is very small. An MA(8) model is a possible option.

Sample PACF (2 time differencing)

```
> pacf(ddy,lag.max=98)
> pacf(ddy)
```

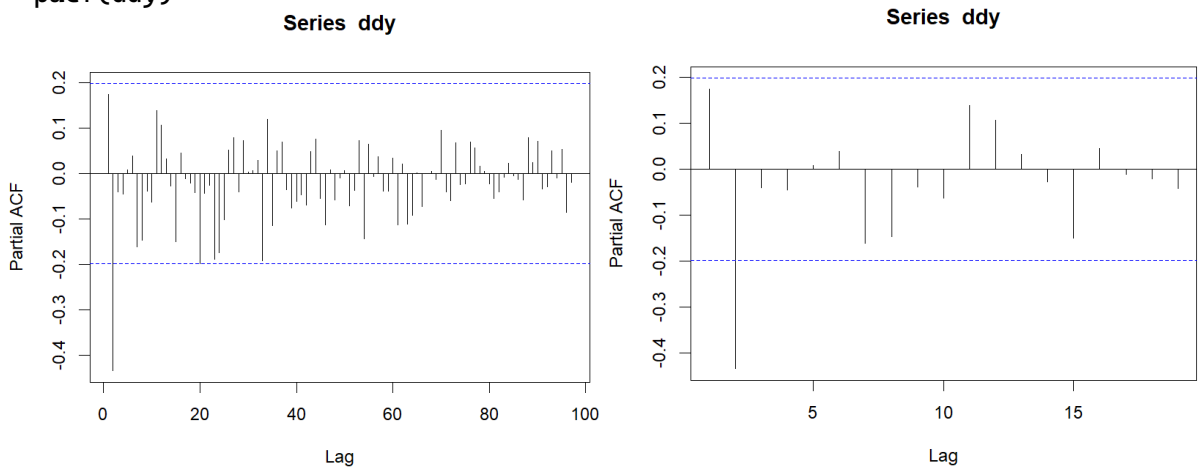

Fig 9: Sample PACF after 2 time differencing

The sample PACF cut off at time lag 2. An option model might be AR(2).

The fitted model

Variance

Table 1: Variance comparison of different transformed data

Transformation	Variance of transformed data
1 time differencing	32.18367
2 time differencing	13.1336

Looking at the variance, the data after 2 time differencing is preferred as that transformation gives a smaller variance.

1 time differencing: MA(6) model

```
> fitMA=arima(y,order=c(0,1,6))

> tsdiag(fitMA)
```

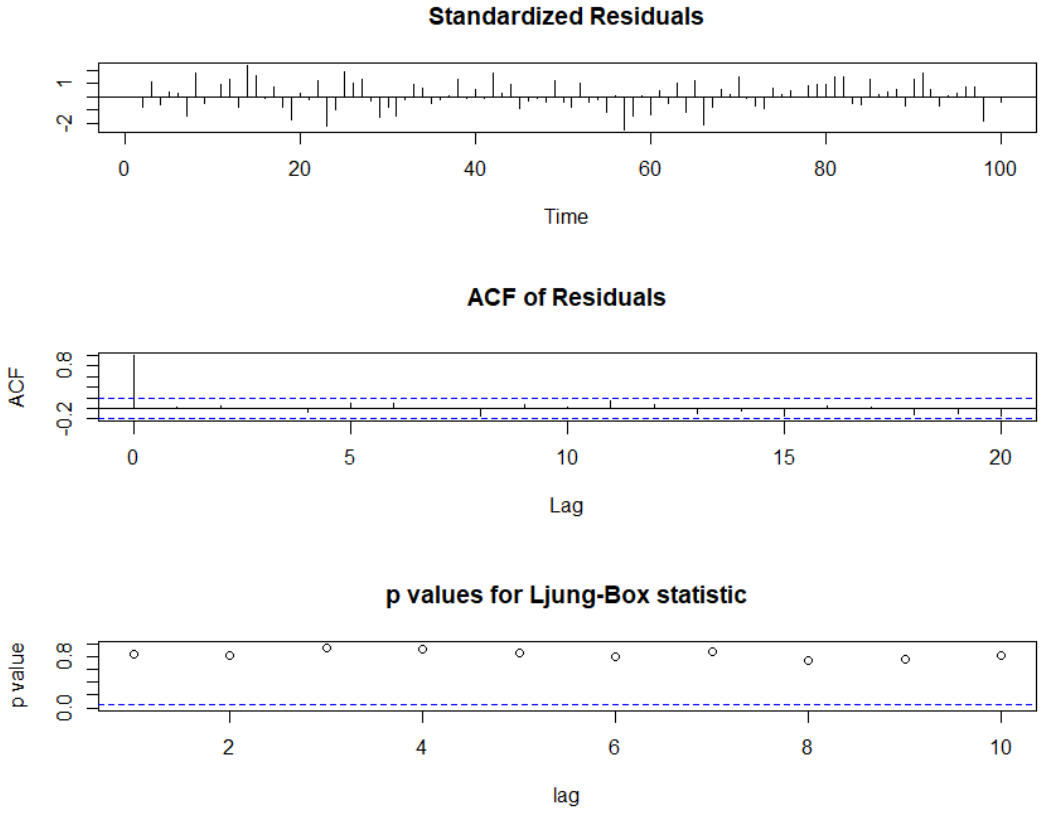

Fig 10: Adequacy check of the MA(6) model with 1 time differencing

The standardized residual of MA(6) model is scattered without any pattern or trend as seen in fig 10. This is good as the residuals should resemble the behaviour of the white noise where the covariance of Z_t and Z_s is 0 when time lag t is not equals to time lag s. The ACF of residuals are within the threshold boundaries as outlined in blue except for when time lag is 0, the ACF residual value is 1. It is not surprising that the ACF value at time lag 1 exceeds the blue outlined boundary because, ACF which is ρ_k, is $\rho_k = \frac{\gamma_k}{\gamma_0}$ and $\rho_0 = \frac{\gamma_0}{\gamma_0} = 1$. Hence, the ACF exceeding the boundary at time lag 0 can be ignored. The p values for Ljung-Box statistics are big and exceeds the blue dotted boundary which

means the p values are not negligible. This shows that the p values are big, and we fail to reject Ho where Ho is that the model is adequate. The MA(6) model with 1 time differencing is adequate.

1 time differencing: AR(3) model

```
> fitAR=arima(y,order=c(3,1,0))

> tsdiag(fitAR)
```

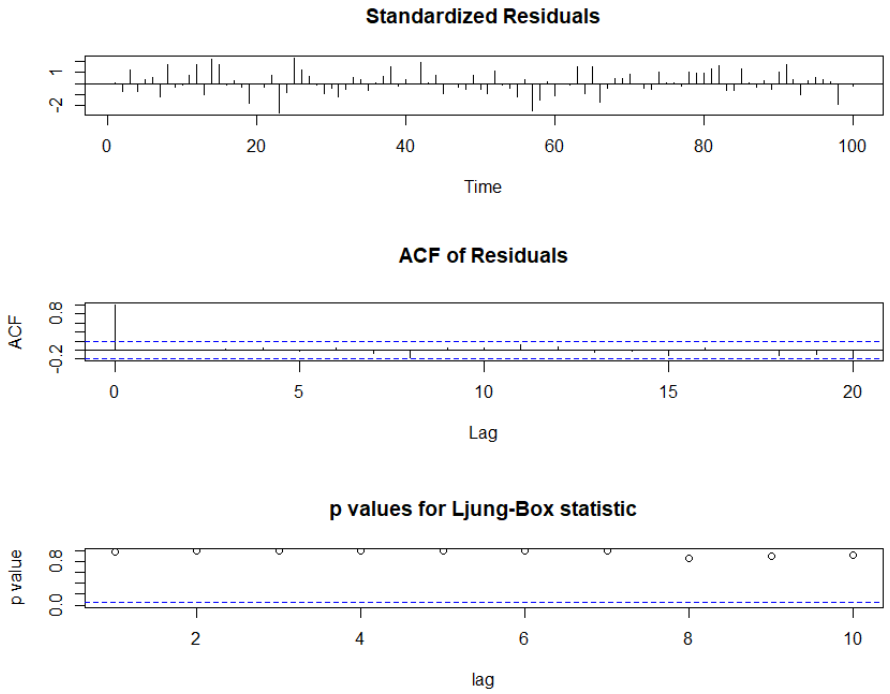

Fig 11: Adequacy check of the AR(3) model with 1 time differencing

The standardized residual of AR(3) model is scattered without any pattern or trend as seen in fig 11. All the ACF of residuals are within the threshold boundaries as outlined in blue except for the ACF when time lag=0. As explained above, ACF at time lag=0 can be ignored as ACF will always be 1 at time lag 0. The p values for Ljung-Box statistics exceeded the blue dotted boundary which means the p values are not negligible. The AR(3) model with 1 time differencing is adequate.

2 time differencing: MA(8) model

```
> fitMA2=arima(y,order=c(0,2,8))

> tsdiag(fitMA2)
```

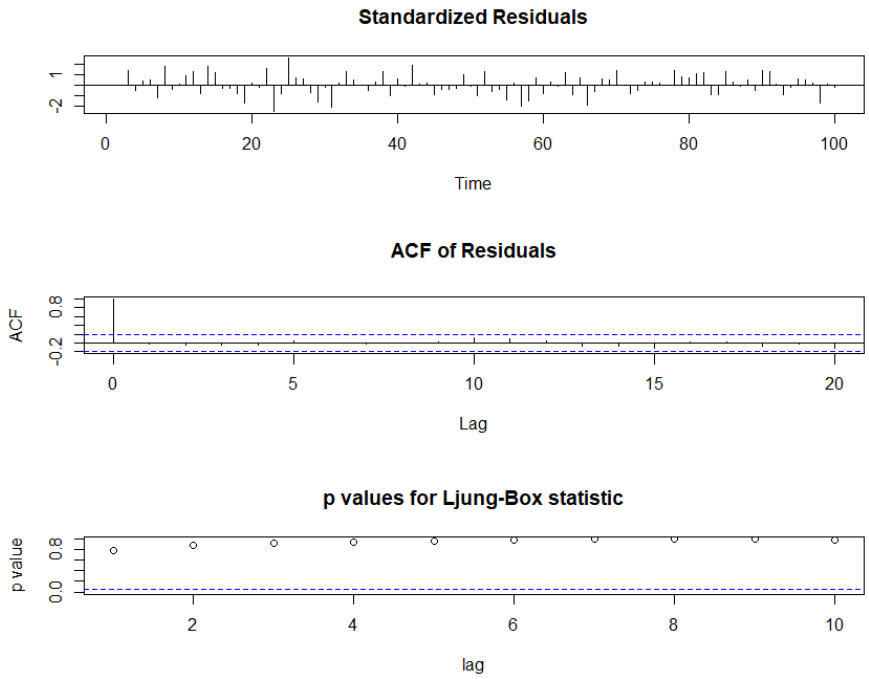

Fig 12: Adequacy check of the MA(8) model with 2 time differencing

The standardized residual of MA(8) model is scattered without any pattern or trend as seen in fig 12. The ACF of residuals are within the threshold boundaries as outlined in blue and the p values for Ljung-Box statistics are big and exceeds the blue dotted boundary which means the p values are not negligible. The MA(8) model with 2 time differencing is adequate.

2 time differencing: AR(2) model

```
> fitAR2=arima(y,order=c(2,2,0))

> tsdiag(fitAR2)
```

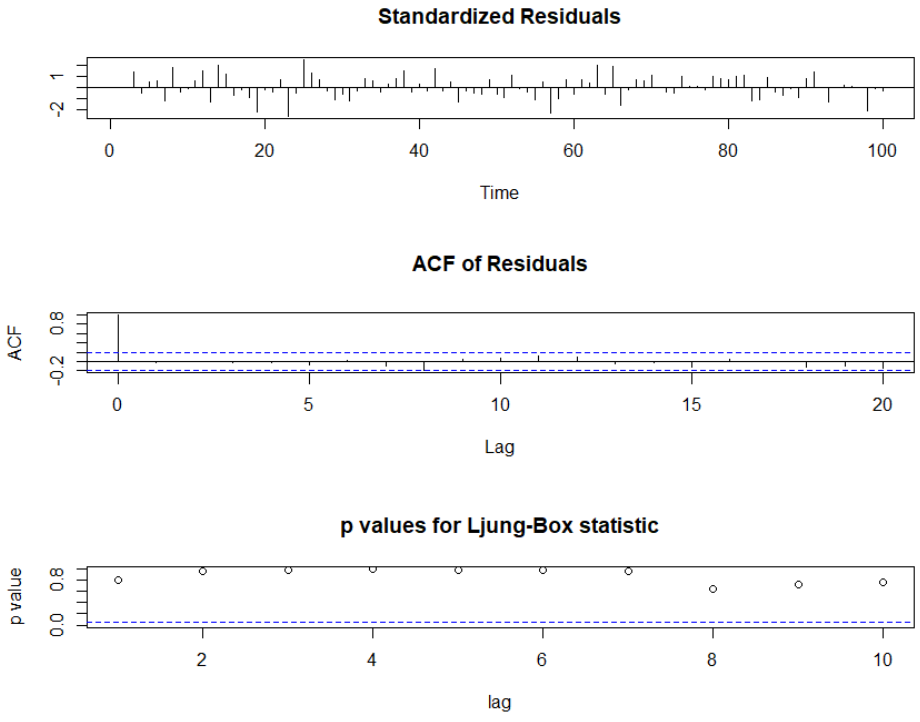

Fig 13: Adequacy check of the AR(2) model with 2 time differencing

The standardized residual of AR(2) model is scattered without any pattern or trend as seen in fig 13. The ACF of residuals are within the threshold boundaries as outlined in blue and the p values for Ljung-Box statistics are big and exceeds the blue dotted boundary which means the p values are not negligible. The AR(2) model with 2 time differencing is adequate.

AIC

We will be using Akaike's Information Criterion (AIC) statistics to evaluate which model is better. The formula for AIC is as shown:

$$AIC = n\ln[\widehat{\delta^2}] + 2(p+q) - - - equation\ 3$$

where q is the moving average process of order and p is the autoregressive process of order. $\widehat{\delta^2}$ is represented as the equation below:

$$\widehat{\delta^2} = \frac{1}{n}\sum_{j=1}^{n}\frac{(X_j - \widehat{X}_j)^2}{r_{j-1}} - - - equation\ 4$$

where r_{j-1} is a constant, independent of δ^2.

Equation 4 shows the estimated error variance. The smaller the error variance, the better as it meant that the predicted point of the model deviates less from the ground truth point. The 1st term in equation 3 accounts for the estimated error variance of the model. This implies that the smaller the AIC value, the better the model. Nevertheless, one of the ways to predict a point well is by overfitting where the models have a lot of parameters where parameters with high correlation to the output label or has little correlation to the output label are in the model. AIC accounts of overfitting as it penalises models which uses large amounts of parameters as shown in the 2nd term of equation 3. The AIC provides a quantitative measure of the trade-off between the goodness of fit of a model and its complexity.

1 time differencing: MA(6) model

```
> fitMA
```

The below is the return result for MA(6) model:

```
Call:
arima(x = y, order = c(0, 1, 6))

Coefficients:
         ma1     ma2     ma3     ma4     ma5     ma6
      1.1178  0.5529  0.2931  0.5413  0.4810  0.1852
s.e.  0.1033  0.1375  0.1317  0.1756  0.1803  0.1051

sigma^2 estimated as 9.359:  log likelihood = -252.13,  aic = 518.26
```

1 time differencing: AR(3) model

```
> fitAR

Call:
arima(x = y, order = c(3, 1, 0))

Coefficients:
         ar1      ar2     ar3
      1.1513  -0.6612  0.3407
s.e.  0.0950   0.1353  0.0941
```

```
sigma^2 estimated as 9.363:  log likelihood = -252,  aic = 511.99
```

2 time differencing: MA(8) model

```
> fitMA2
```

The below is the return result for MA(8) model:

```
Call:
arima(x = y, order = c(0, 2, 8))

Coefficients:
         ma1      ma2      ma3     ma4      ma5      ma6      ma7      ma8
      0.2042  -0.4073  -0.2512  0.1481  -0.0620  -0.1214  -0.2036  -0.3069
s.e.  0.1012   0.1119   0.1248  0.1265   0.0998   0.1174   0.1270   0.1151

sigma^2 estimated as 8.721:  log likelihood = -247.13,  aic = 512.26
```

2 time differencing: AR(2) model

```
> fitAR2
```

The below is the return result for AR(2) model:

```
Call:
arima(x = y, order = c(2, 2, 0))

Coefficients:
         ar1      ar2
      0.2579  -0.4407
s.e.  0.0915   0.0906

sigma^2 estimated as 10.13:  log likelihood = -252.73,  aic = 511.46
```

Looking at the AIC from the 4 models, the AR(2) model with 2 time differencing has the smallest AIC. Hence, AR(2) model with 2 time differencing is the best model. The fitted model is as follows:

$$(1 - B^2)y_t = Z_t + 0.2579Z_{t-1} - 0.4407Z_{t-2}$$

Comparing top 2 models with the data

```
#graph without zooming in

> plot(1:100, y, xlim = c(0, 100), ylim = c(0, 250), xlab = "Observation
number", ylab = "Number of users connected to the internet ")

> lines(1:100, y, type="l", col = "black")

> lines(1:100, y - fitAR2$residuals, type="l", col = "red")

> lines(1:100, y - fitAR$residuals, type="l", col = "blue")

> legend("topleft", legend = c("Original Data", "Time Diff 1 AR(3)", "Time
Diff 2 AR(2)"), fill = c("black","blue", "red"))

#graph after zooming in

> plot(1:100, y, xlim = c(0, 100), ylim = c(80, 230), xlab = "Observation
number", ylab = "Number of users connected to the internet ")

> lines(1:100, y, type="l", col = "black")
```

```
> lines(1:100, y - fitAR2$residuals, type="l", col = "red")

> lines(1:100, y - fitAR$residuals, type="l", col = "blue")

> legend("topleft", legend = c("Original Data", "Time Diff 1 AR(3)", "Time
Diff 2 AR(2)"), fill = c("black","blue", "red"))
```

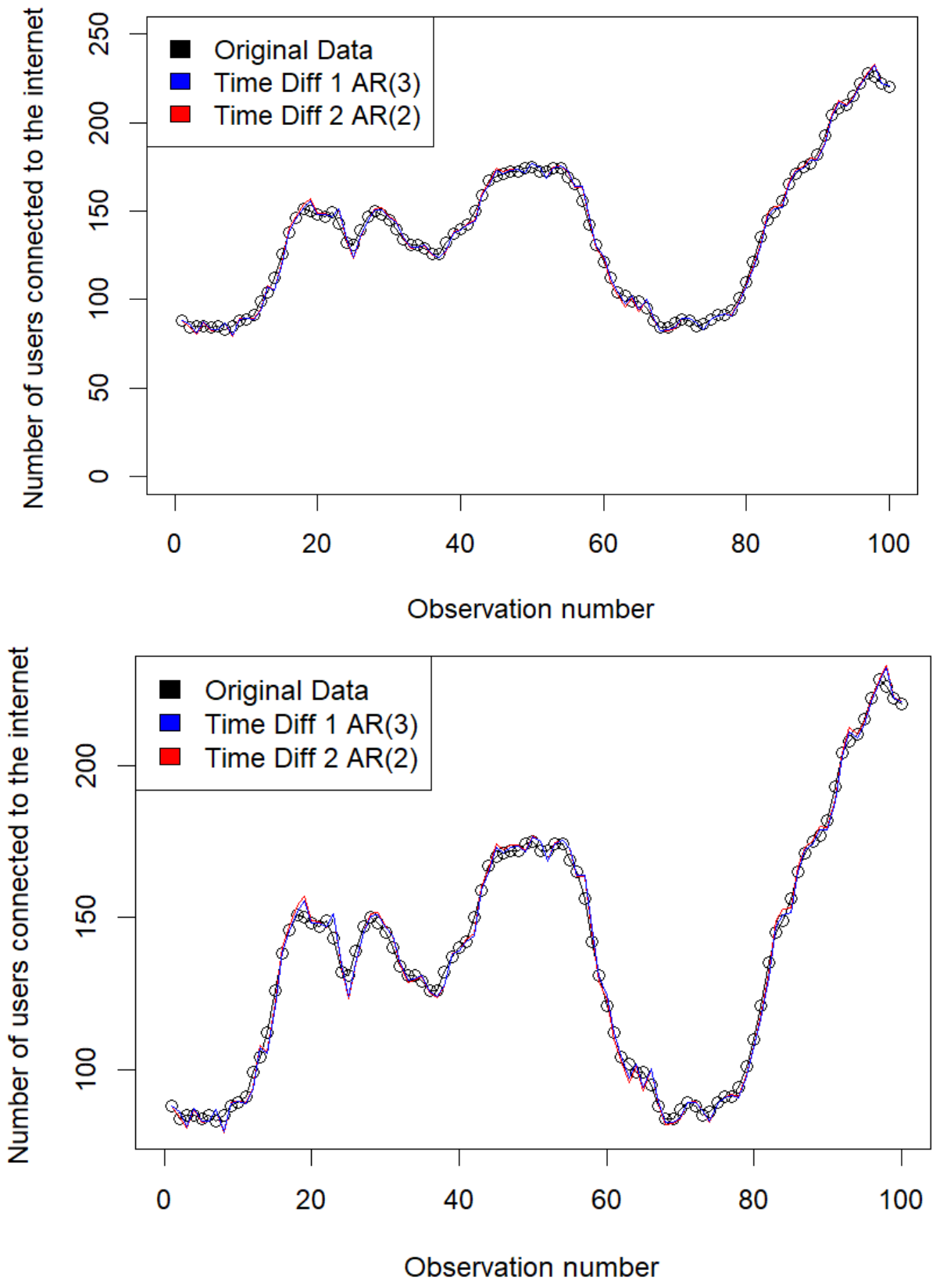

Fig 14: Comparing the predicted AR(2) model with 2 time differencing, predicted AR(3) model with 1 time differencing and the observations from the dataset

Visually from looking fig 14, we can see that the predicted AR(2) model with 2 time differencing and AR(3) model with 1 time differencing fit the observations in the dataset to a very good extent as both the red line and blue line lie very closely to the black line which is plotted from the values from the observations of the dataset.

Predicted data

Using the best model we have, AR(2) with 2 time differencing, we predict the 101th and 102th point.

```
> predict(fitAR2, n.ahead= 2)
$pred
Time Series:
Start = 101
End = 102
Frequency = 1
[1] 219.3972 218.2732

$se
Time Series:
Start = 101
End = 102
Frequency = 1
[1] 3.182263 7.858337
```

Using the second best model we have, AR(3) with 1 time differencing, we predict the 101th and 102th point.

```
> predict(fitAR, n.ahead= 2)
$pred
Time Series:
Start = 101
End = 102
Frequency = 1
[1] 219.6608 219.2299

$se
Time Series:
Start = 101
End = 102
Frequency = 1
[1] 3.059957 7.259431
```

Table 2: Predicted value of 101th and 102th point from the top 2 models

Model	101th point	101th point after round off	102th point	102th point after round off
AR(2) time differencing 2	219.3972	220	218.2732	219
AR(3) time differencing 1	219.6608	220	219.2299	220

```
# graph without zoom in

> ytd1<-c(219.6608, 219.2299)
> ytd2<-c(219.3972, 218.2732)
> plot(1:100, y, xlim = c(0, 100), ylim = c(0, 250), xlab = "Observation
number", ylab = "Number of users connected to the internet ")
> lines(1:100, y, type="l", col = "black")
> lines(101:102, ytd1, type="l", col = "blue")
> lines(101:102, ytd2, type="l", col = "red")

# graph with zoom in

> plot(1:100, y, xlim = c(0, 100), ylim = c(80, 230), xlab = "Observation
number", ylab = "Number of users connected to the internet ")
> lines(1:100, y, type="l", col = "black")
```

```
> lines(101:102, ytd1, type="l", col = "blue")
> lines(101:102, ytd2, type="l", col = "red")
> legend("topleft", legend = c("Original Data", "Time Diff 1 AR(3)", "Time
Diff 2 AR(2)"), fill = c("black","blue", "red"))
```

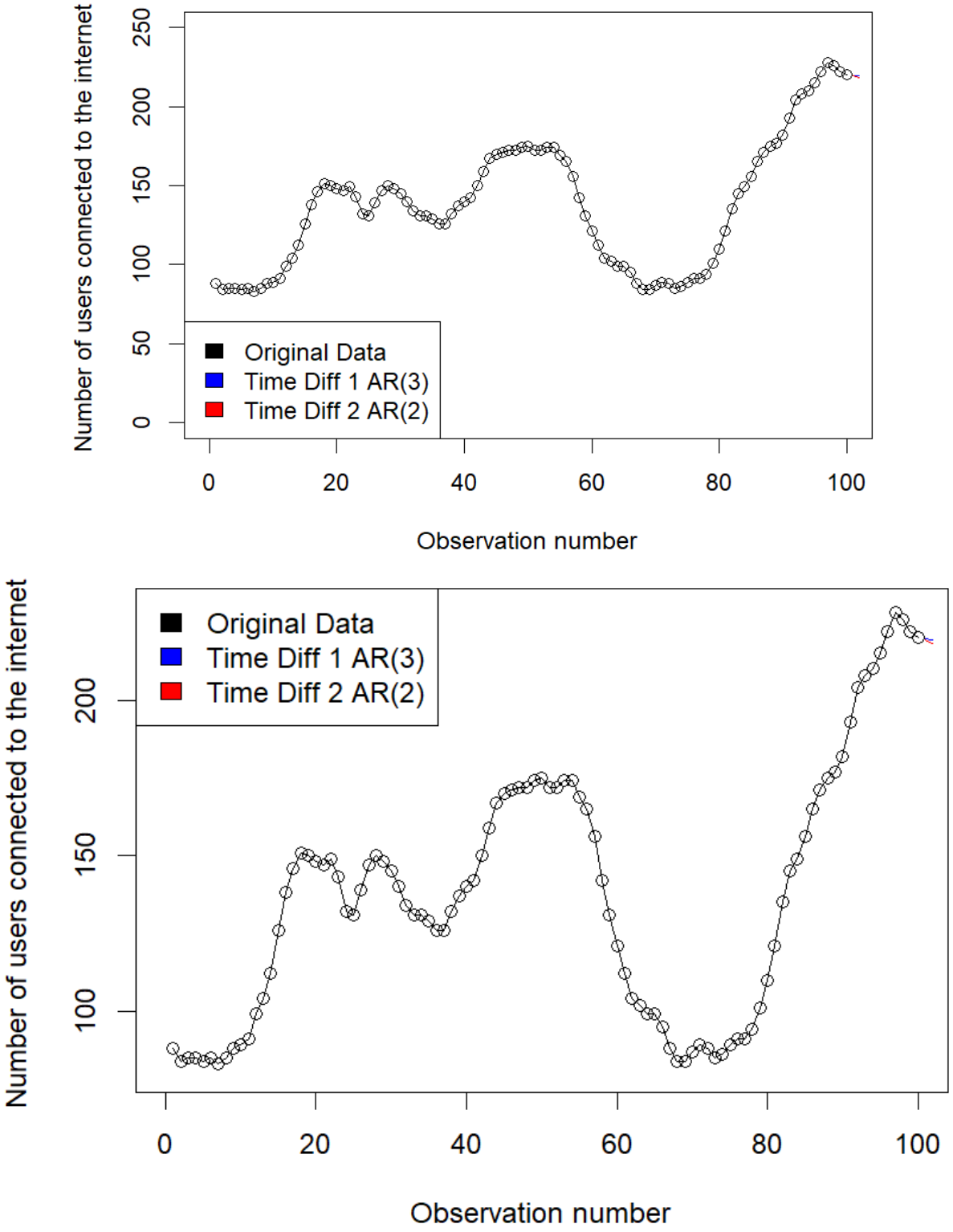

Fig 15: Graph containing the 101th and 102th predicted points

From table 2, we see that both models predicted a similar number of people connected to the internet at time lag 101 and 102. As number of people cannot be in decimals, the number of people is rounded up to the nearest integer and both models gave a rounded off value of 220 people at time

point 101. AR(2) model gave a rounded off value of 219 people at time lag 102 while AR(3) model gave a rounded off value of 220 people at time lag 102.

Figure 15 is plotted with the predicted values in decimals as what was returned from R Studio instead of the rounded off predicted values. In fig 15, we see that the AR(2) model predicted a slightly bigger decrease in the number of users connected to the internet than the AR(3) model from time lag 101 to time lag 102.