AI6123 Time Series Analysis
Assignment 1

Zheng, Weixiang
G2103278G

Load data

In the first section, we use `scan` function to load the data under the same folder. Because the first entry is "x", we skip the first entry by specifying `skip=1`. Then we print out the minimum, the maximum and the average value of the time series. Finally we print out the time series on the plot.
"A time series is said to be stationary if there is no systematic change in mean (no trend), if there is no systematic change in variance if strictly periodic variations have been removed." We can see clearly that it is **NOT Stationary** since there is a obvious upward trend at the end of the time series.

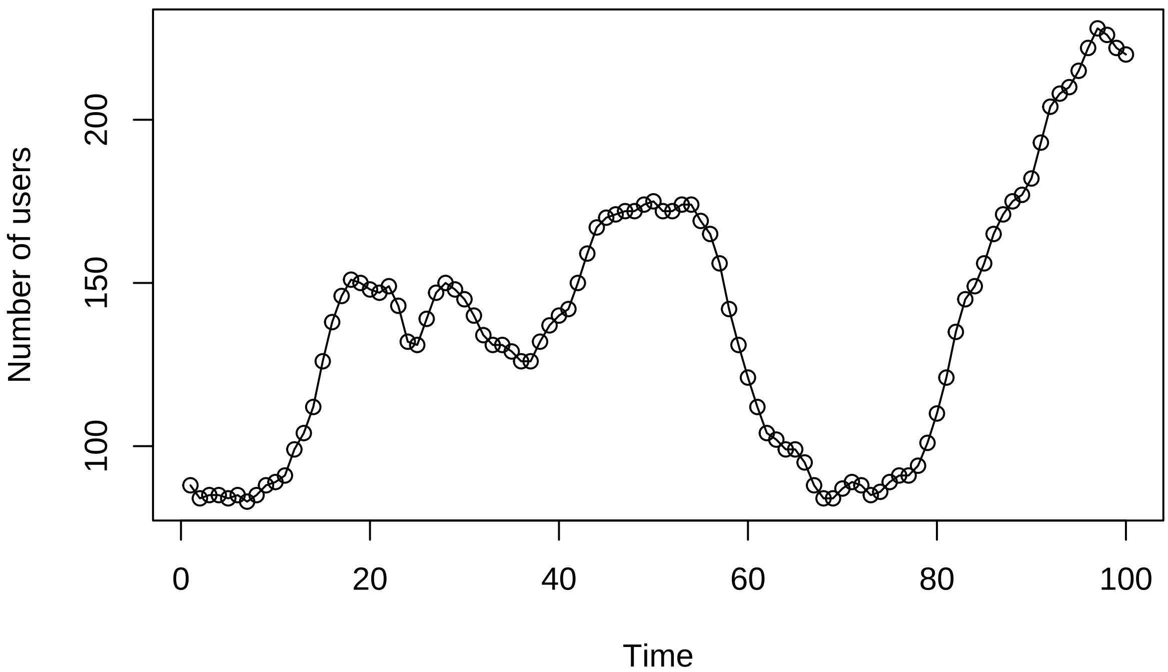
```
data = scan("./wwwusage.txt",skip=1)
min(data);max(data);mean(data)
```

```
## [1] 83
```

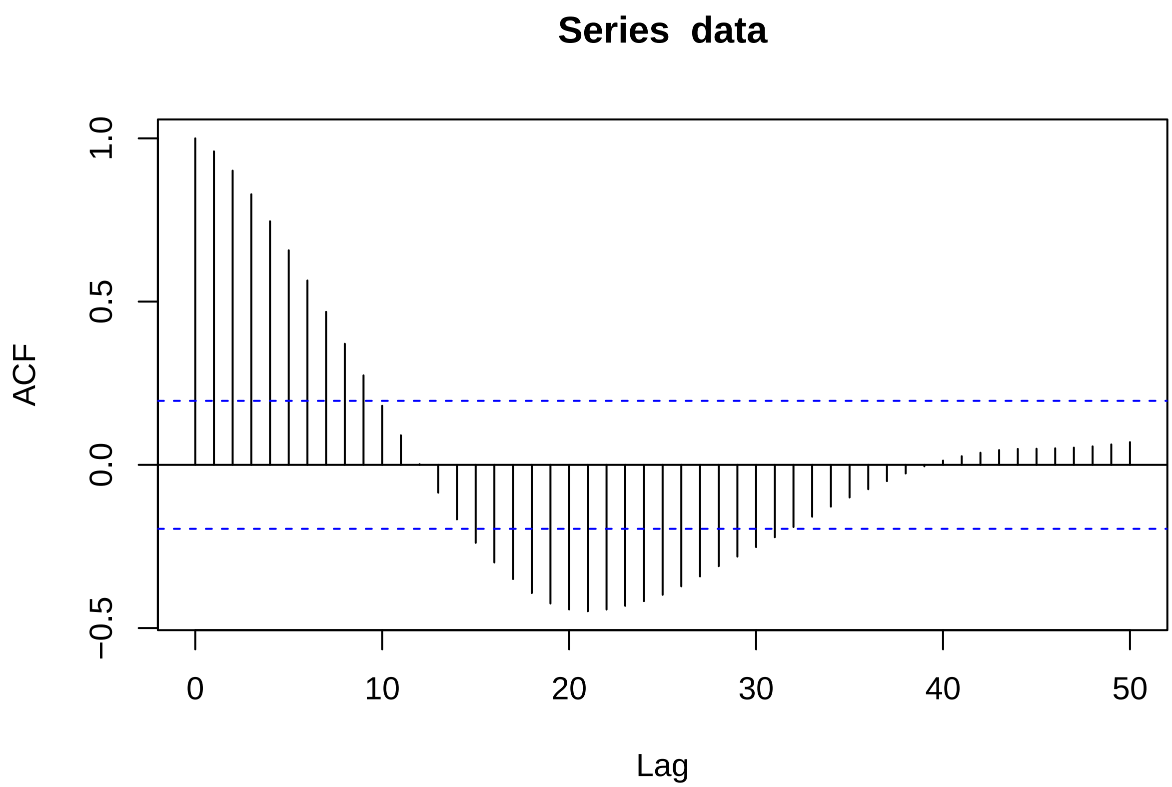
```
## [1] 228
```

```
## [1] 137.08
```

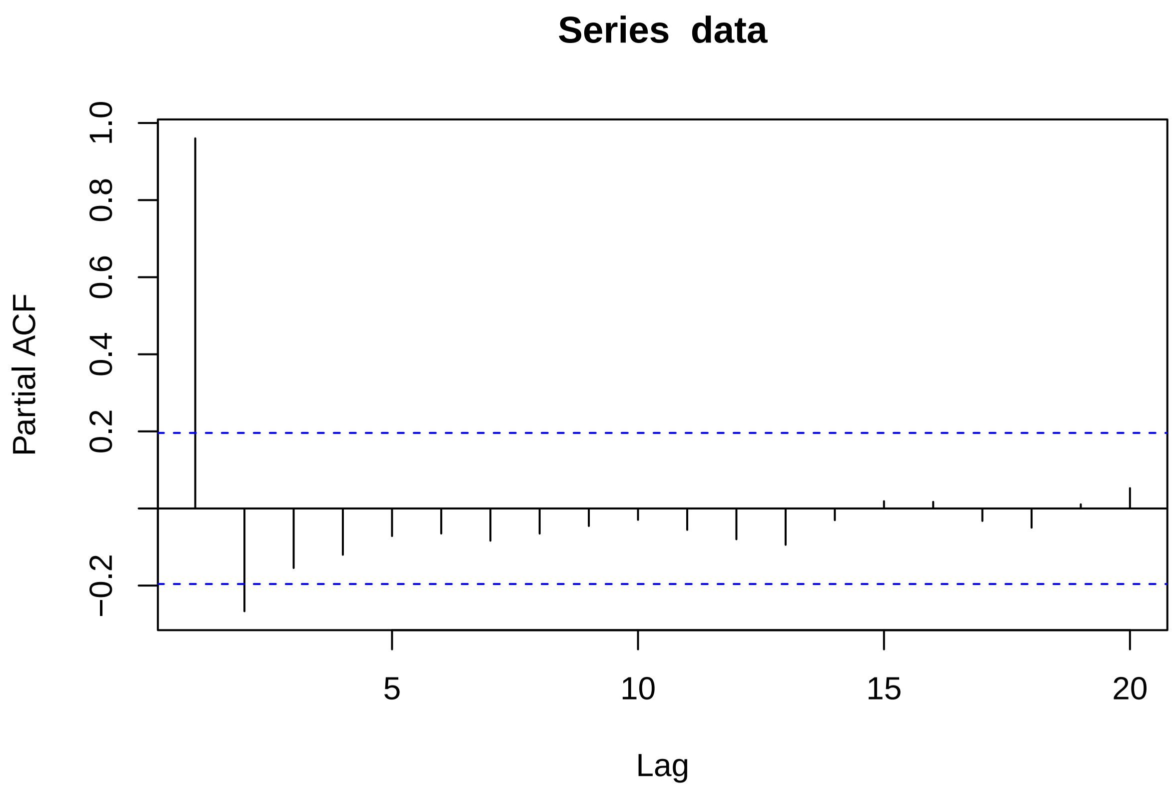
```
plot(data,xlab="Time",ylab="Number of users")
lines(data)
```




```r
# cut off after 31
acf(data,lag.max = 50)
```




```
# cut off after 2
pacf(data)
```

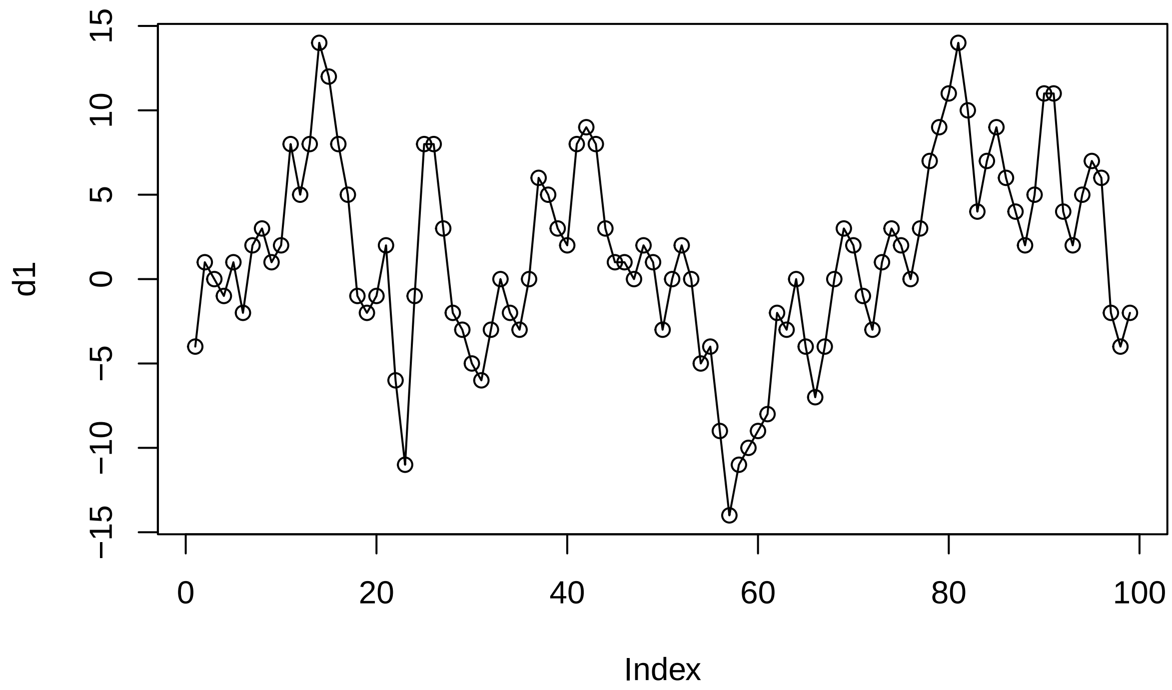


We use `acf` and `pacf` function to plot out the Autocorrelation Function value up to max lag of 50 by specifying `lag.max = 50`. We notice that ACF value cut off after lag 31. For Partial Autocorrelation Function value, we notice that it cut off after lag 2.

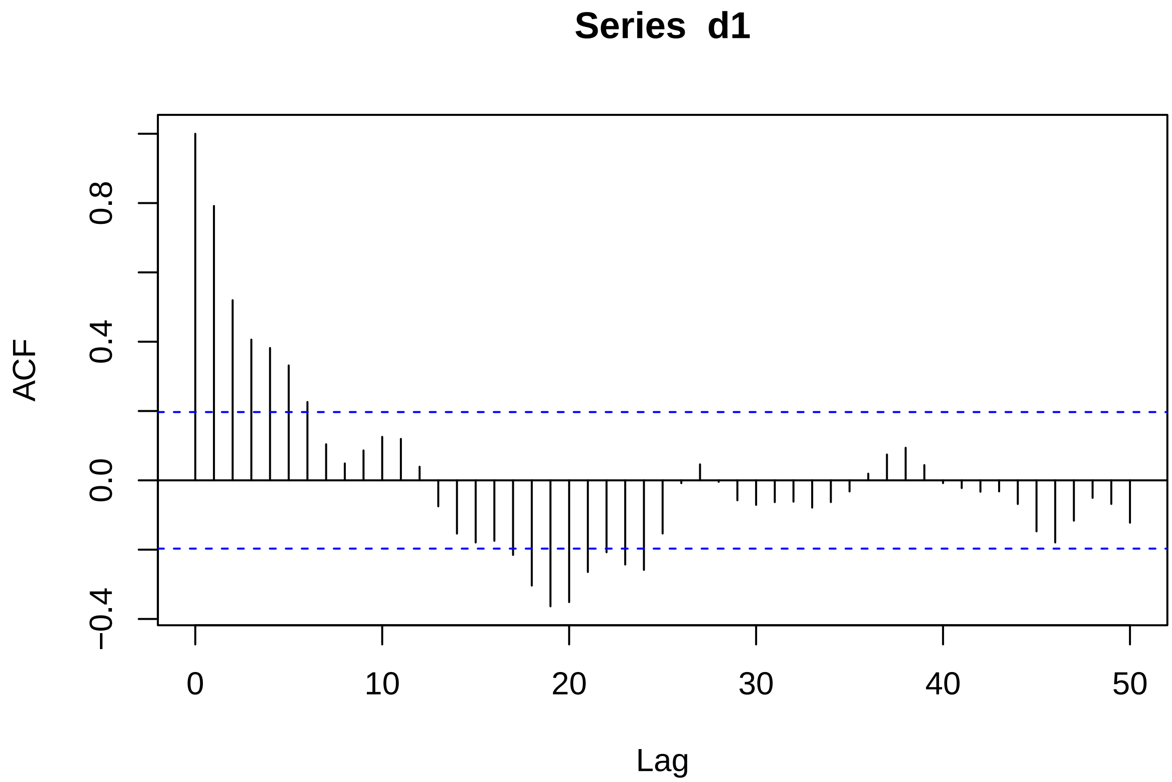
First Order Differencing

We do a first order differencing on the original data, then plot it out. We notice the plot is more stationary than the original plot. Then we plot the ACF and PACF as before. We notice ACF cut off after lag 24, PACF cut off after lag 3. We try out AR(3) model. Then we use Yule-Walker Estimation method to make an estimation on the parameters. YW Estimation suggests it is a AR(3) model with parameters 1.106, -0.596 and 0.303. We fit the original data into a AR(3) model and run diagnosis.

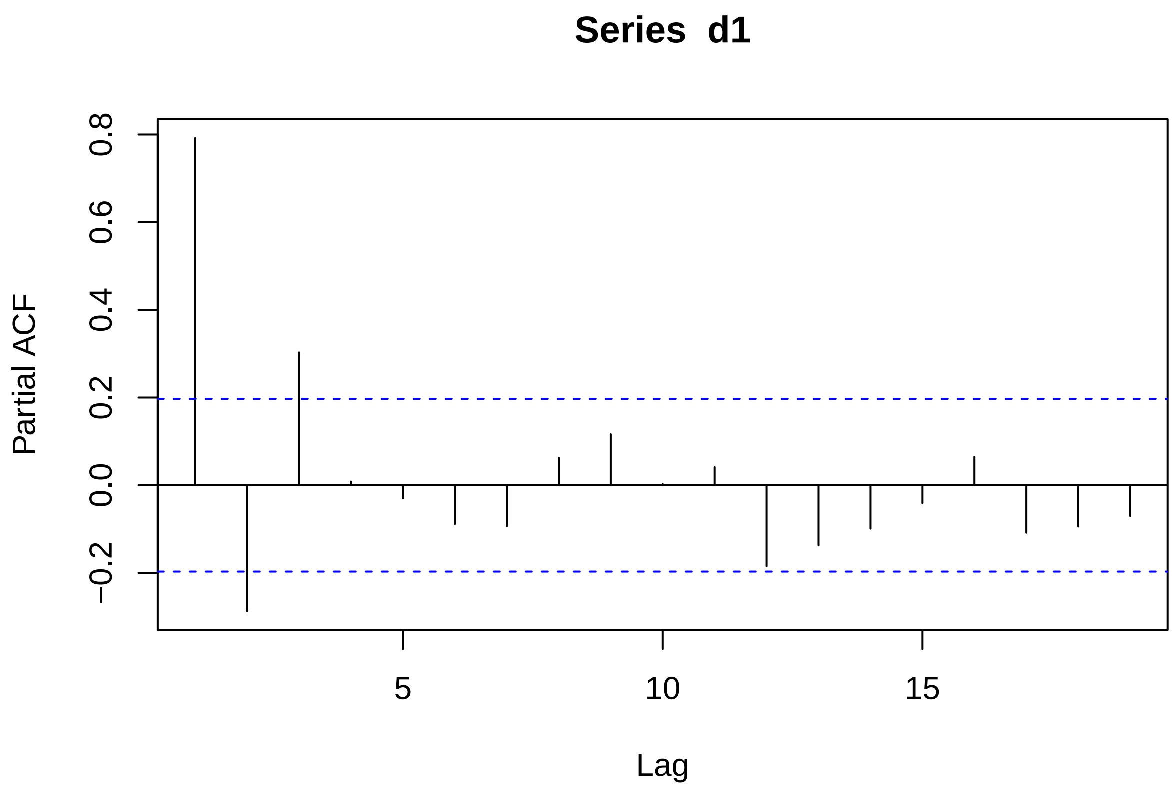
```
d1 = diff(data)
plot(d1)
lines(d1)
```




```
# cut off after 24
acf(d1,lag.max = 50)
```




```
# cut off after 3
pacf(d1)
```




```
# Yule-Walker Estimation
yw1 = ar.yw(d1)
summary(yw1);yw1 # AR(3): 1.106, -0.596, 0.303
```

```
##              Length Class  Mode
## order         1     -none- numeric
## ar            3     -none- numeric
## var.pred      1     -none- numeric
## x.mean        1     -none- numeric
## aic          20     -none- numeric
## n.used        1     -none- numeric
## n.obs         1     -none- numeric
## order.max     1     -none- numeric
## partialacf   19     -none- numeric
## resid        99     -none- numeric
## method        1     -none- character
## series        1     -none- character
## frequency     1     -none- numeric
## call          2     -none- call
## asy.var.coef  9     -none- numeric
```

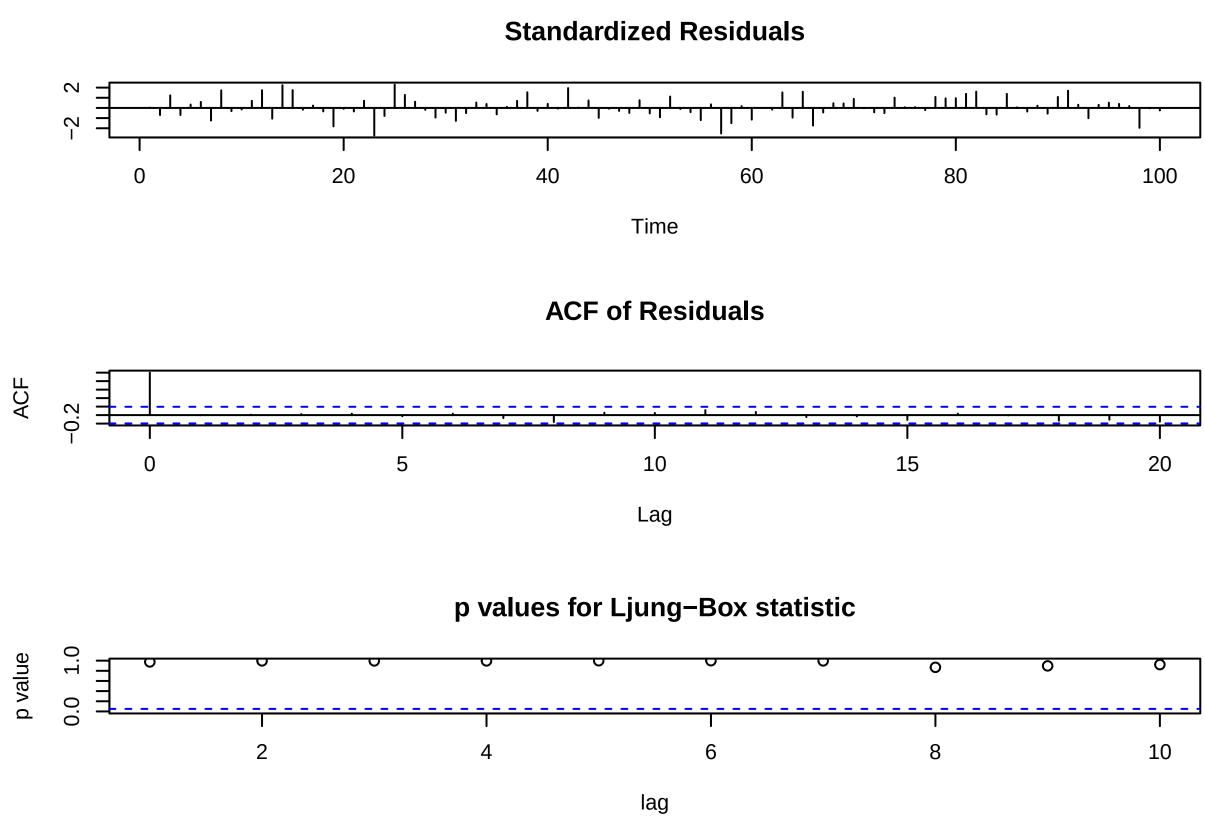
```
##
## Call:
## ar.yw.default(x = d1)
##
```

```
## Coefficients:
##       1        2        3
##  1.1060  -0.5957   0.3029
##
## Order selected 3  sigma^2 estimated as  10.32
```

```
# Fit ARIMA Model (3,1,0)
arima1 = arima(data, order=c(3,1,0))
arima1
```

```
##
## Call:
## arima(x = data, 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
```

```
tsdiag(arima1)
```



```
AIC(arima1)
```

```
## [1] 511.994
```

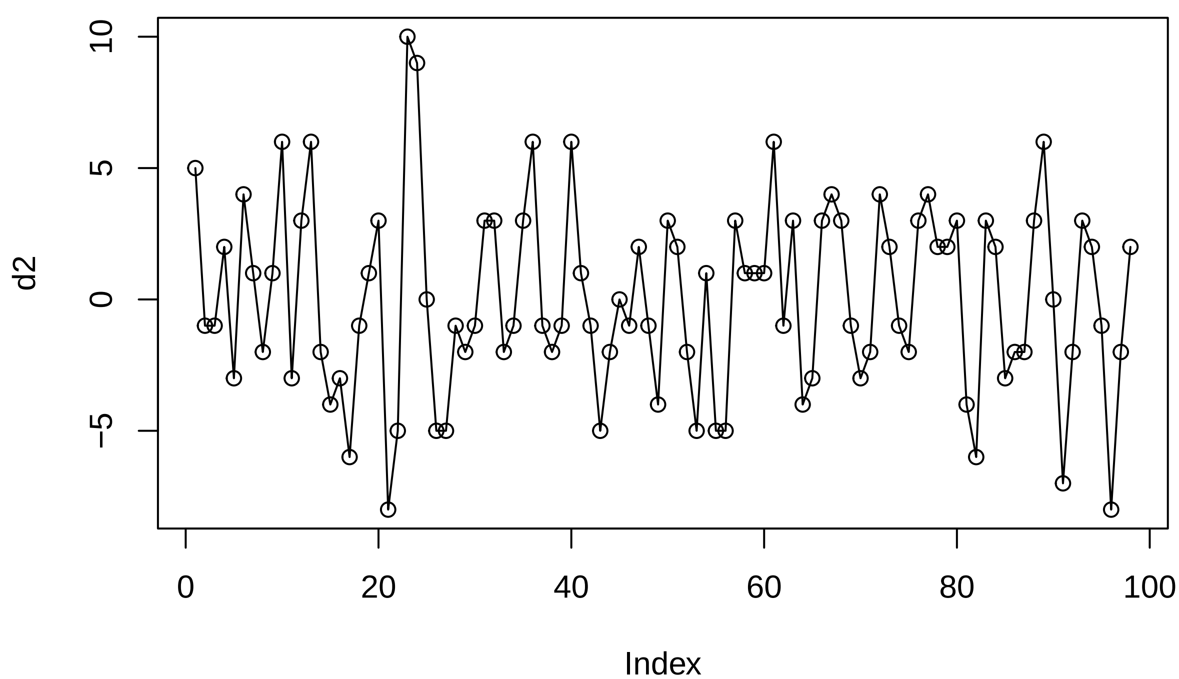
The diagnostic checking is shown above. We can see the fitted ARIMA(3,1,0) model is OK(adequate) because:

- The residual values look random, which makes it white noise term
- The ACF cuts off after lag 0
- The p values indicate they are significance

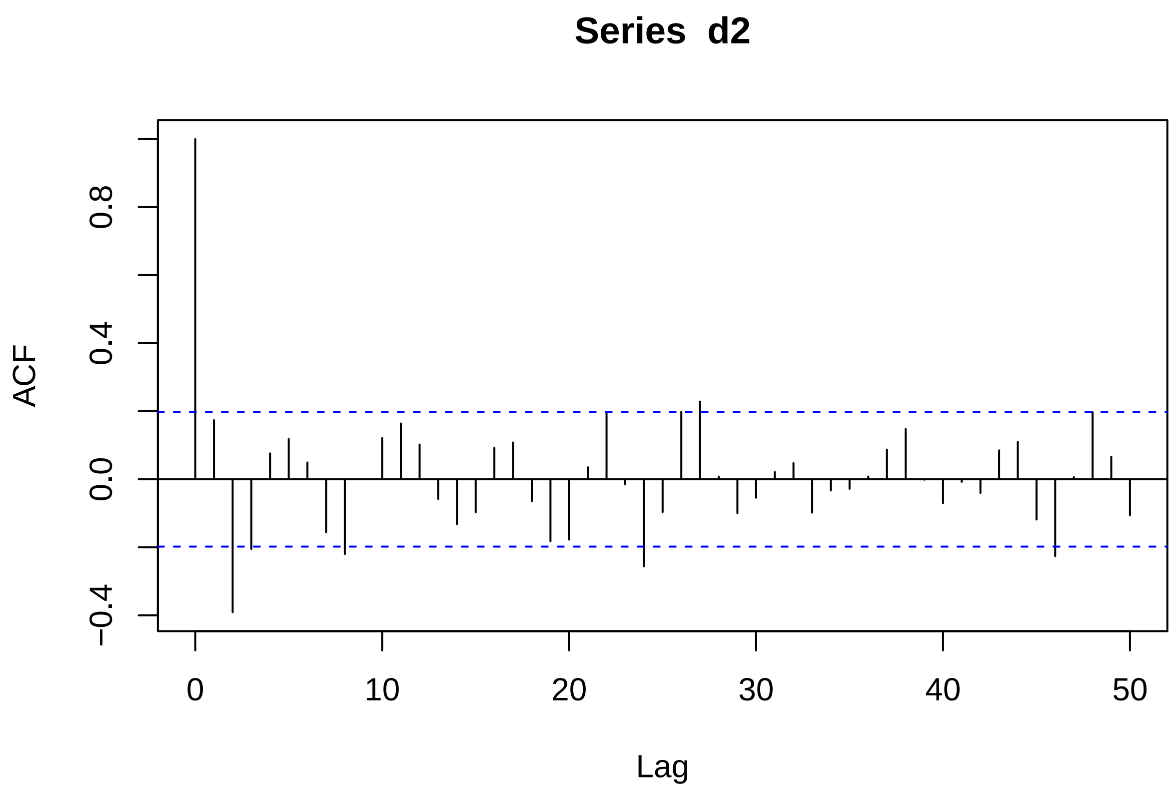
Second Order Differencing

Next, we do a second order differencing by differencing one more time on `d1` data. We plot the graph as before, then print out the ACF and PACF. We notice ACF cut off after lag 27, PACF cut off after lag 2. Therefore we try out AR(2) model. Same as before, we use Yule-Walker Estimation to estimate the parameters and get 0.2489 and -0.4341. We fit data into ARIMA(2,2,0) model and run diagnosis.

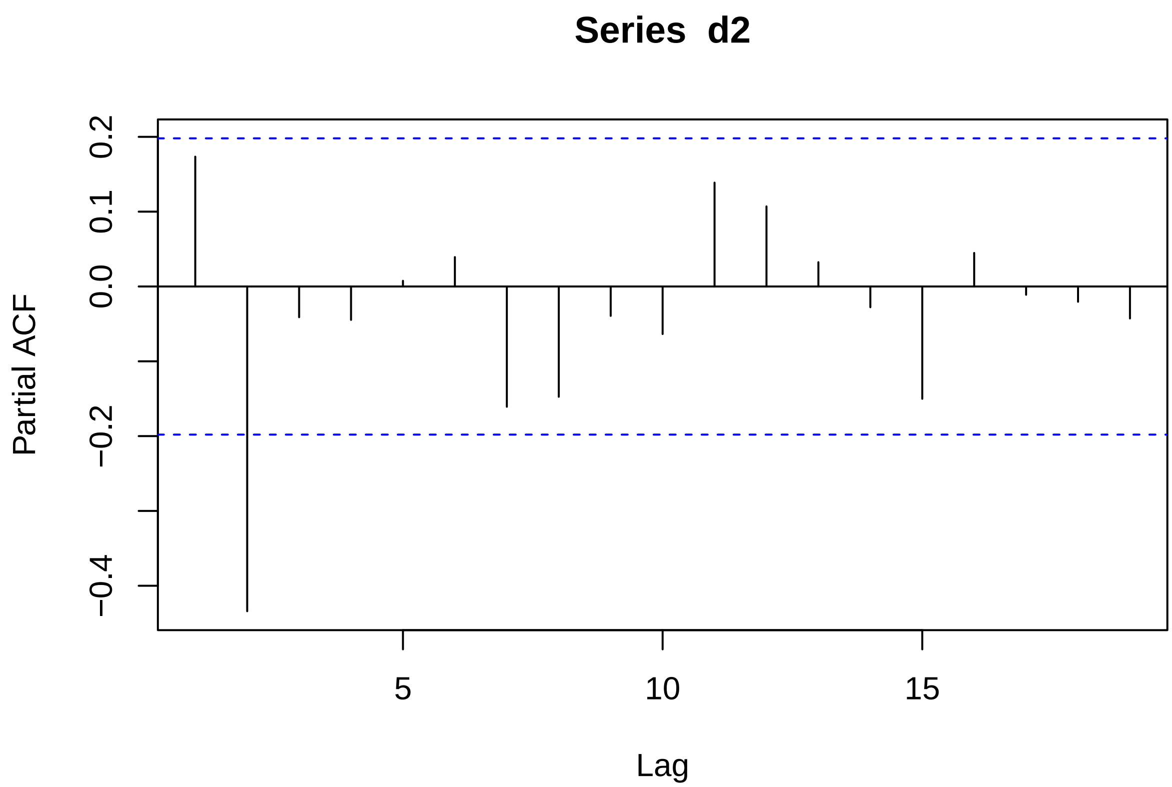
```
d2 = diff(d1)
plot(d2)
lines(d2)
```



```r
# cut off after 27
acf(d2,lag.max = 50)
```




```r
# cut off after 2
pacf(d2)
```




```
# Yule-Walker Estimation
yw2 = ar.yw(d2)
summary(yw2);yw2 # AR(2): 0.2489 and -0.4341
```

```
##               Length Class  Mode
## order          1     -none- numeric
## ar             2     -none- numeric
## var.pred       1     -none- numeric
## x.mean         1     -none- numeric
## aic           20     -none- numeric
## n.used         1     -none- numeric
## n.obs          1     -none- numeric
## order.max      1     -none- numeric
## partialacf    19     -none- numeric
## resid         98     -none- numeric
## method         1     -none- character
## series         1     -none- character
## frequency      1     -none- numeric
## call           2     -none- call
## asy.var.coef   4     -none- numeric
```

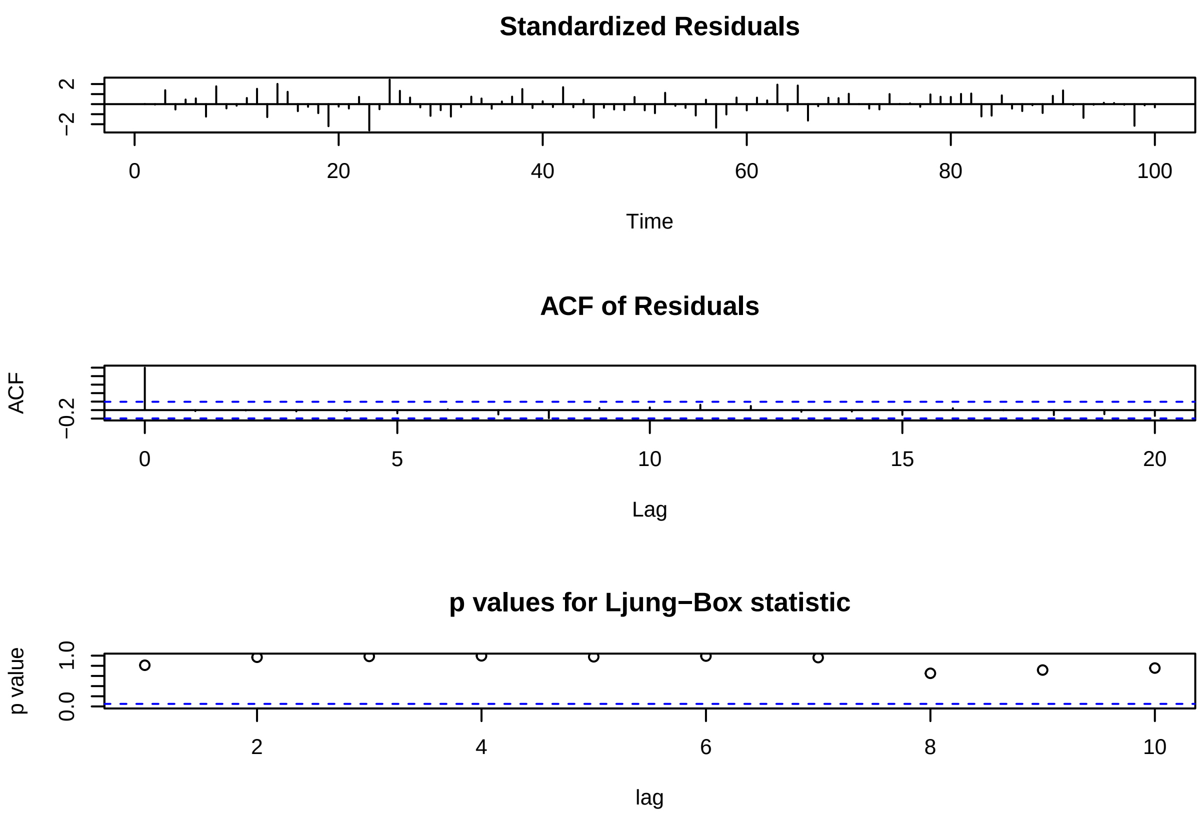
```
##
## Call:
## ar.yw.default(x = d2)
##
```

```
## Coefficients:
##       1        2
##  0.2489  -0.4341
##
## Order selected 2  sigma^2 estimated as  10.56
```

```r
# Fit ARIMA Model (2,2,0)
arima2 = arima(data, order=c(2,2,0))
arima2
```

```
##
## Call:
## arima(x = data, 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
```

```r
tsdiag(arima2)
```



```r
AIC(arima2)
```

```
## [1] 511.4645
```

The diagnostic checking is shown above. We can see the fitted ARIMA(2,2,0) model is OK(adequate) because:

- The residual values look random, which makes it white noise term
- The ACF cuts off after lag 0
- The p values indicate they are significance

Auto ARIMA

In this section, we use the AUTO ARIMA function of the forecast library to compare with the manual statistics. AUTO ARIMA suggests ARIMA(1,1,0) model, we run diagnosis on the arima3.

```r
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

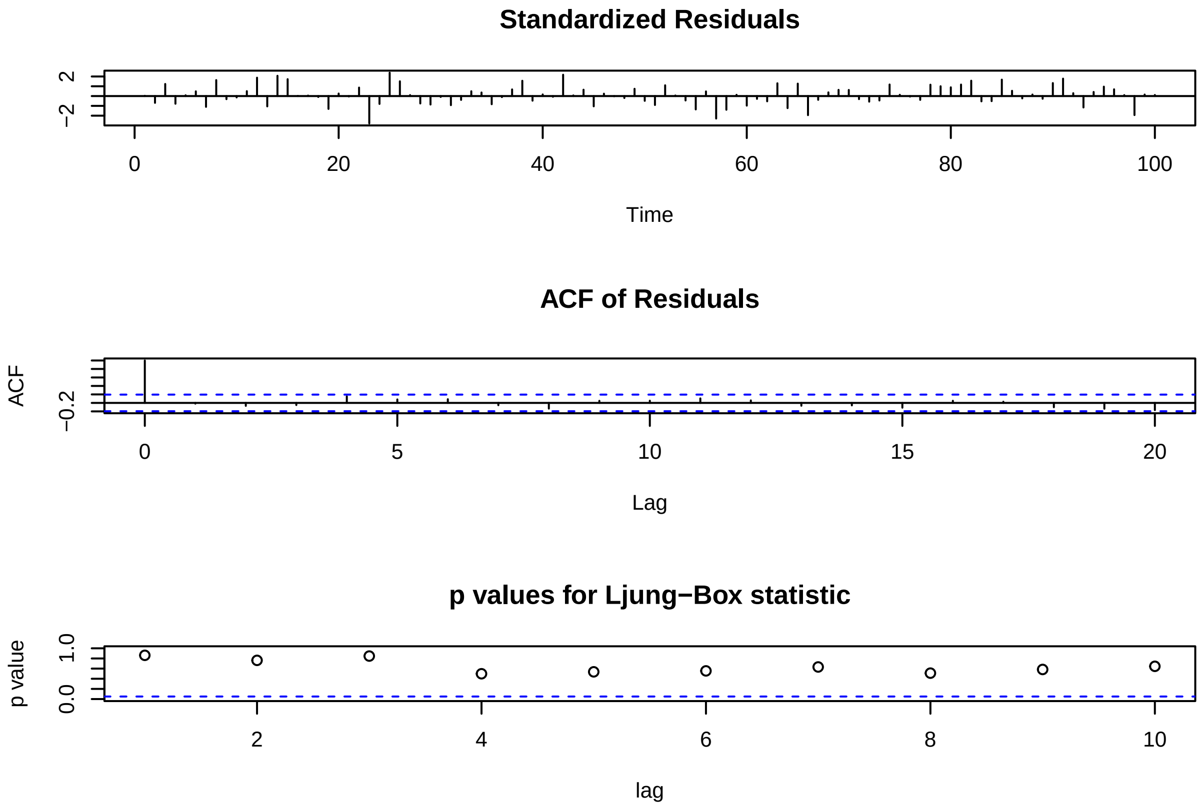
```r
autoarima = auto.arima(data)
autoarima
```

```
## Series: data
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1     ma1
##       0.6504  0.5256
## s.e.  0.0842  0.0896
##
## sigma^2 = 9.995:  log likelihood = -254.15
## AIC=514.3   AICc=514.55   BIC=522.08
```

```r
arima3 = arima(data, order=c(1,1,1))
arima3
```

```
##
## Call:
## arima(x = data, order = c(1, 1, 1))
##
## Coefficients:
##          ar1     ma1
##       0.6504  0.5256
## s.e.  0.0842  0.0896
##
## sigma^2 estimated as 9.793:  log likelihood = -254.15,  aic = 514.3
```

```
tsdiag(arima3)
```




```
AIC(arima3)
```

```
## [1] 514.2995
```

The diagnostic checking is shown above. We can see the fitted ARIMA(1,1,1) model is OK(adequate) because:

- The residual values look random, which makes it white noise term
- The ACF cuts off after lag 0
- The p values indicate they are significance

Conclusion

In summary, we load the data, it is not stationary, we do a first order differencing, it becomes more stationary, therefore we try to fit it into a model. We find the best model for one-time differencing models is ARIMA(3,1,0). Then we also tried two-time differencing models, and find the best model is ARIMA(2,2,0). Then we used `forecast` to do a auto arima fitting, which gives us ARIMA(1,1,1). The statistics:

```
-----------------------
Model           AIC
-----------------------
```

```
ARIMA(3,1,0)    511.994
ARIMA(2,2,0)    511.4645
ARIMA(1,1,1)    514.2995
------------------------
```

The Best model we found is therefore: ***ARIMA(2,2,0)***

Appendix: Code

```
knitr::opts_chunk$set(echo = TRUE)
data = scan("./wwwusage.txt",skip=1)
min(data);max(data);mean(data)
plot(data,xlab="Time",ylab="Number of users")
lines(data)
# cut off after 31
acf(data,lag.max = 50)
# cut off after  2
pacf(data)
d1 = diff(data)
plot(d1)
lines(d1)
# cut off after 24
acf(d1,lag.max = 50)
# cut off after 3
pacf(d1)

# Yule-Walker Estimation
yw1 = ar.yw(d1)
summary(yw1);yw1 # AR(3): 1.106, -0.596, 0.303

# Fit ARIMA Model (3,1,0)
arima1 = arima(data, order=c(3,1,0))
arima1
tsdiag(arima1)
AIC(arima1)
d2 = diff(d1)
plot(d2)
lines(d2)
# cut off after 27
acf(d2,lag.max = 50)
# cut off after 2
pacf(d2)

# Yule-Walker Estimation
yw2 = ar.yw(d2)
summary(yw2);yw2 # AR(2): 0.2489 and -0.4341

# Fit ARIMA Model (2,2,0)
arima2 = arima(data, order=c(2,2,0))
arima2
tsdiag(arima2)
```

```
AIC(arima2)
library(forecast)
autoarima = auto.arima(data)
autoarima
arima3 = arima(data, order=c(1,1,1))
arima3
tsdiag(arima3)
AIC(arima3)
-----------------------
Model           AIC
-----------------------
ARIMA(3,1,0)    511.994
ARIMA(2,2,0)    511.4645
ARIMA(1,1,1)    514.2995
------------------------
```