sARIMA_Model

Yun Huang (Amily)

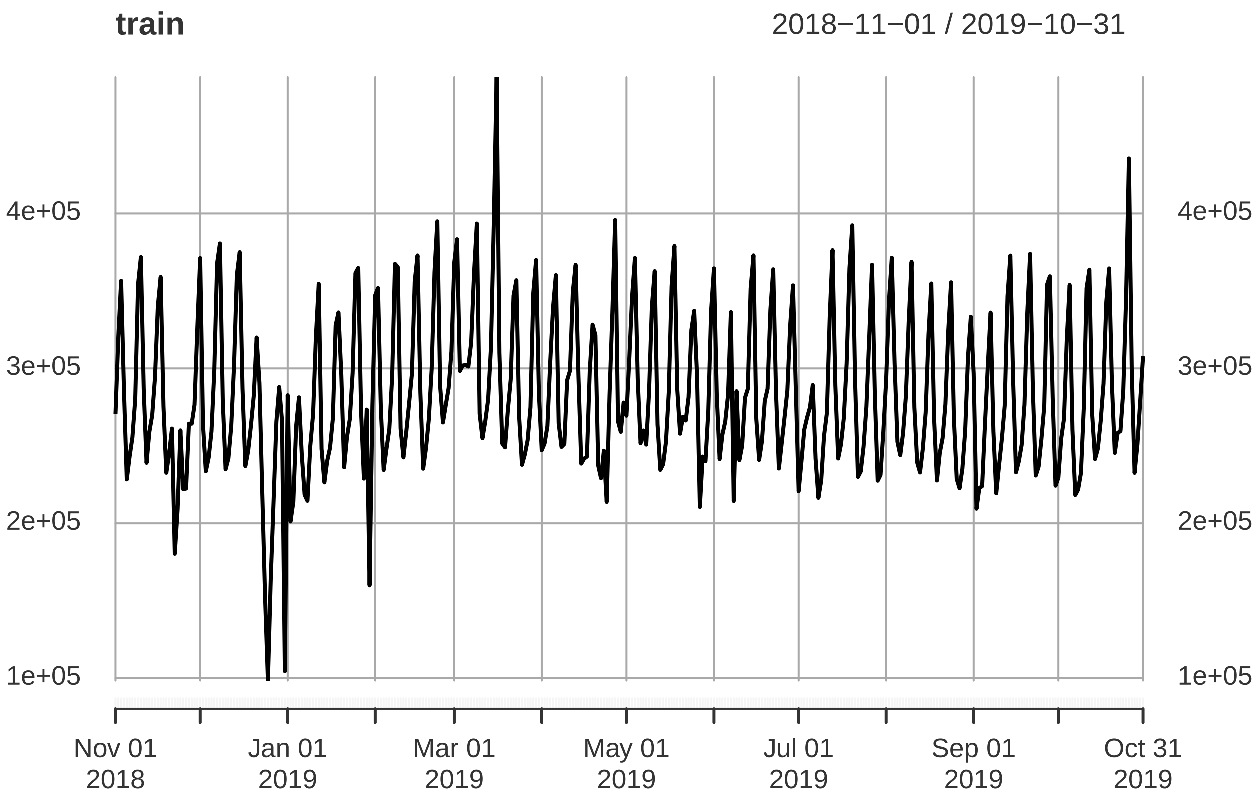
Import Data

```r
dataPath<- "/Users/amilyhuang/Google Drive (yunh@uchicago.edu)/04-Uchicago/03-Fall_20/02-Time_Series/03-
rs<- read.csv(file = paste(dataPath,"RidershipTS.csv", sep = "/"), header = TRUE )
```

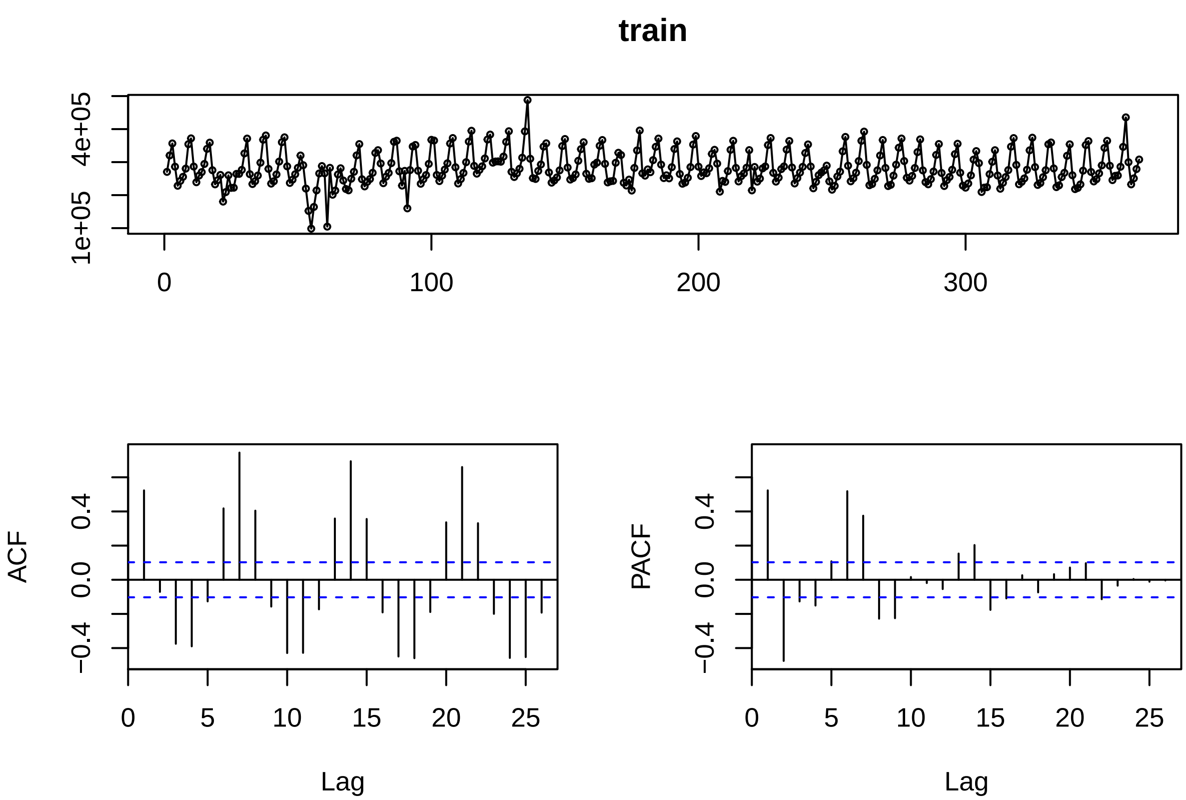
Ridership TS Plot

```r
time_index <- seq(from = as.Date("2018-11-01"),
                  to = as.Date("2019-12-31"), by = "day")

time_index_train <- seq(from = as.Date("2018-11-01"),
                  to = as.Date("2019-10-31"), by = "day")

time_index_test <- seq(from = as.Date("2019-11-01"),
                  to = as.Date("2019-12-31"), by = "day")

rs_ts <- as.xts(rs[,2], order.by = time_index, frequency = 365.25)

train <- as.xts(rs_ts['2018-11-01/2019-10-31'], frequency = 365.25)

test <- as.xts(rs_ts['2019-11-01/2019-12-31'], frequency = 365.25)
```

```r
plot(train)
```




```r
tsdisplay(train)
```



```r
adf.test(train, k = 25)  #Nonstationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  train
## Dickey-Fuller = -3.1652, Lag order = 25, p-value = 0.09404
## alternative hypothesis: stationary
```

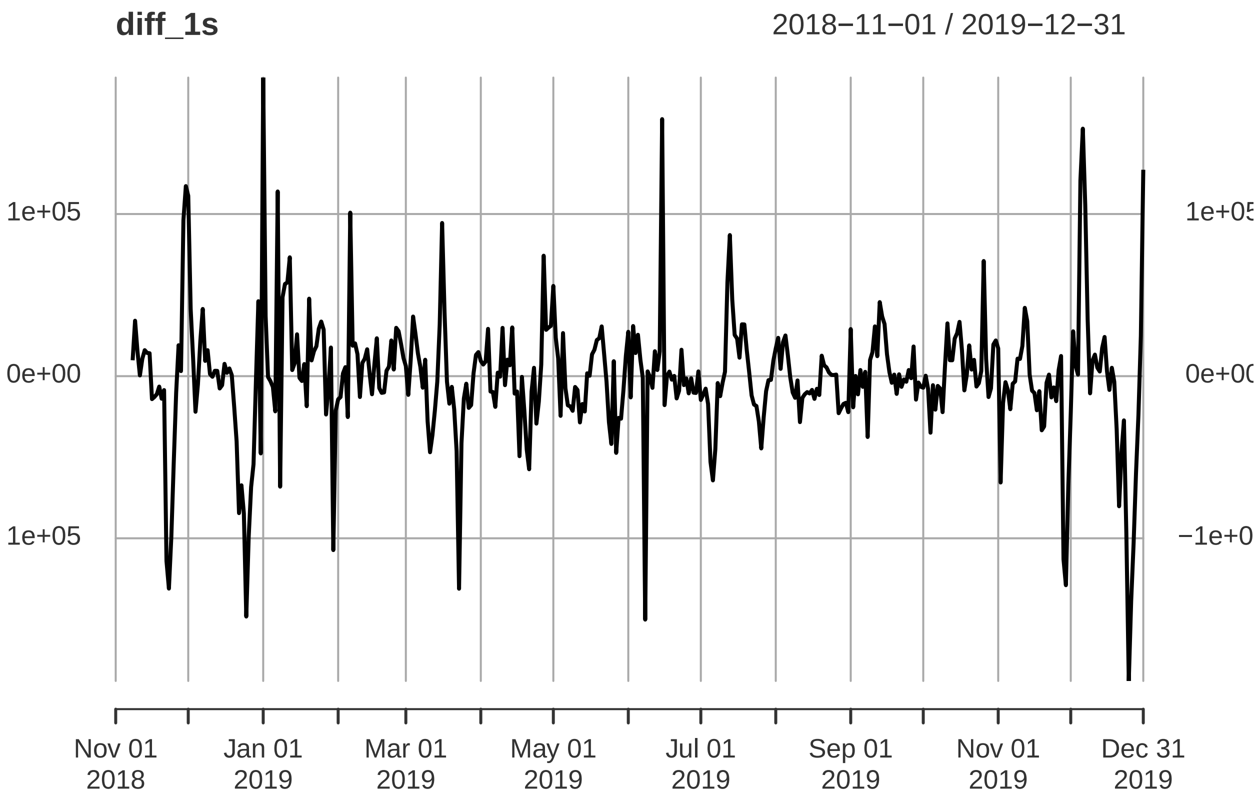
```r
#Nonstationary + Seasonality at lag = 7
```

Model Identification and Model Selection

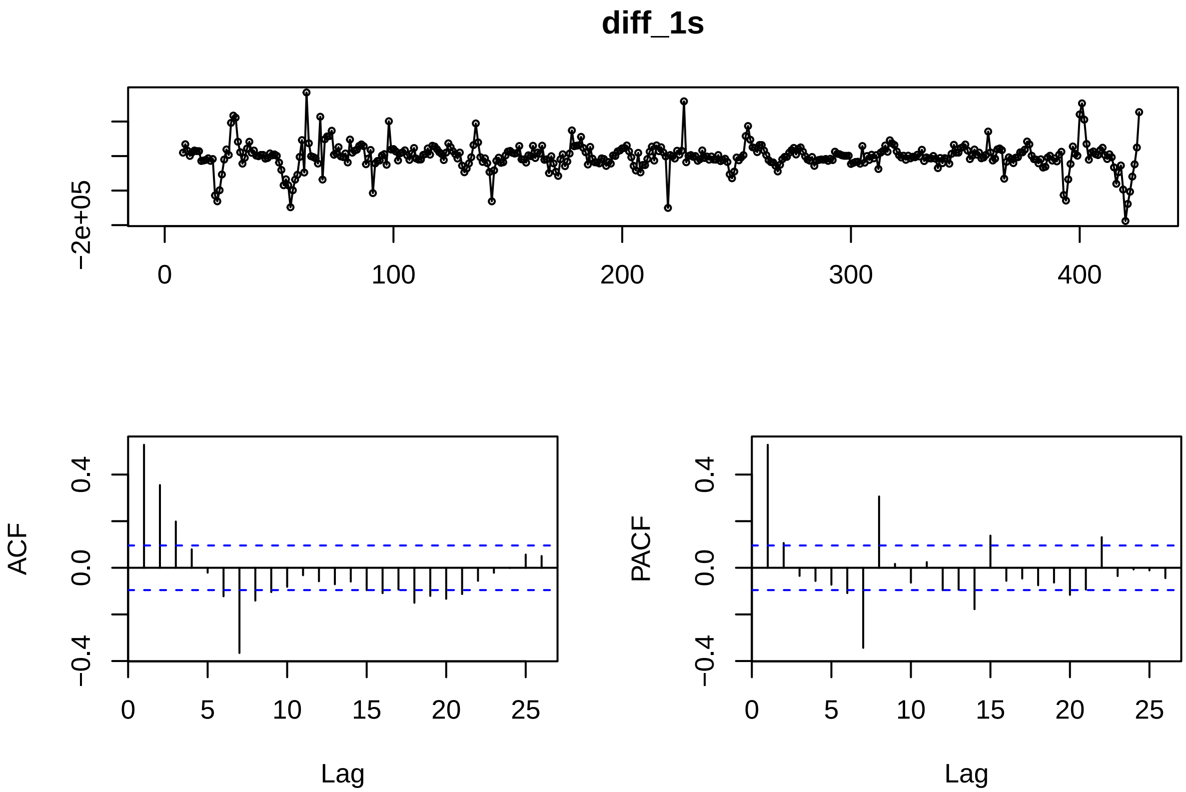
We don't need boxcox transformation.

First Seasonal Differencing

```r
#d=1
diff_1s<- diff(rs_ts, lag = 7)
plot(diff_1s)
```



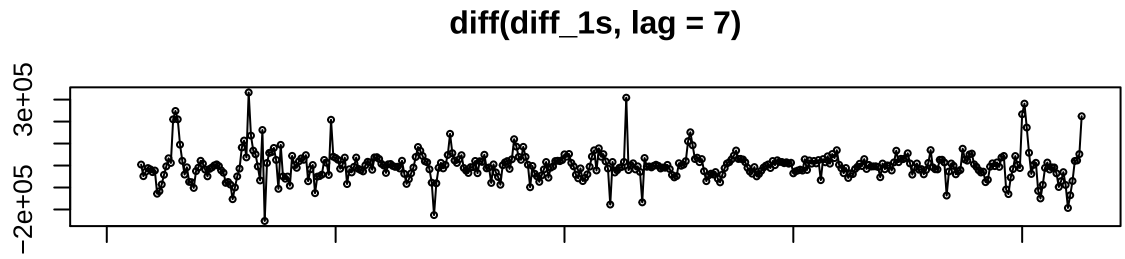

```r
tsdisplay(diff_1s)
```

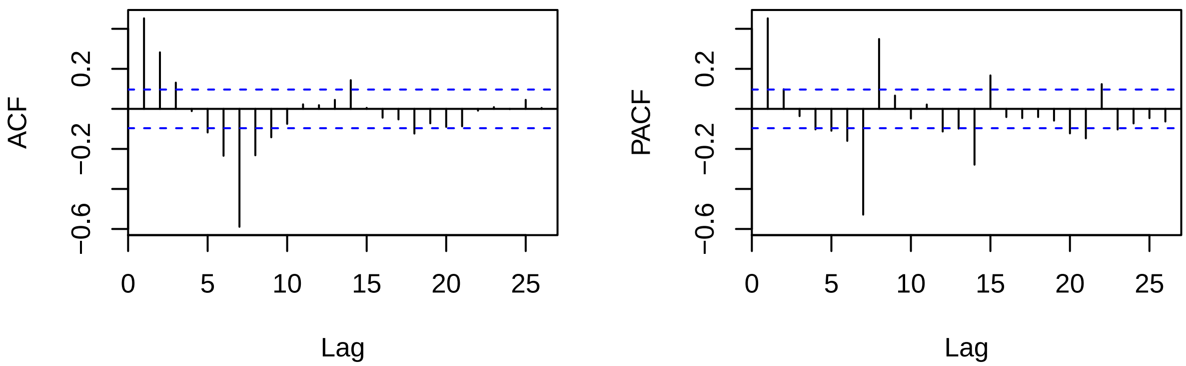



```
#sARIMA(1,0,0)(0,1,1)[7]
#sARIMA(2,0,0)(0,1,1)[7]
```

Second differencing is not better

```
tsdisplay(diff(diff_1s, lag=7))
```





Apply adf and kpss test

```r
#Null hypothesis: The process is nonstationary.
#Alternative hypothesis:process is stationary

adf.test(diff_1s[8:365]) #Is stationary
```

```
## Warning in adf.test(diff_1s[8:365]): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff_1s[8:365]
## Dickey-Fuller = -7.1761, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```r
#Null hypothesis: The process is stationary.
#Alternative hypothesis: The process is nonstationary

kpss.test(diff_1s) #Is stationary
```

```
## Warning in kpss.test(diff_1s): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff_1s
## KPSS Level = 0.052945, Truncation lag parameter = 5, p-value = 0.1
```

Model Estimation

Use maximum likelihood estimation

```r
#sARIMA(1,0,0)(0,1,1)[7]
(res_1<-Arima(train, order = c(1,0,0), seasonal = list(order = c(0,1,1), period=7), method="ML"))
```

```
## Series: train
## ARIMA(1,0,0)(0,1,1)[7]
##
## Coefficients:
##          ar1     sma1
##       0.5744  -0.9999
## s.e.  0.0433   0.0940
##
## sigma^2 estimated as 539818524:  log likelihood=-4120.09
## AIC=8246.18   AICc=8246.24   BIC=8257.82
```

```r
#sARIMA(2,0,0)(0,1,1)[7]
(res_2<-Arima(train, order = c(2,0,0), seasonal = list(order = c(0,1,1), period=7), method="ML"))
```

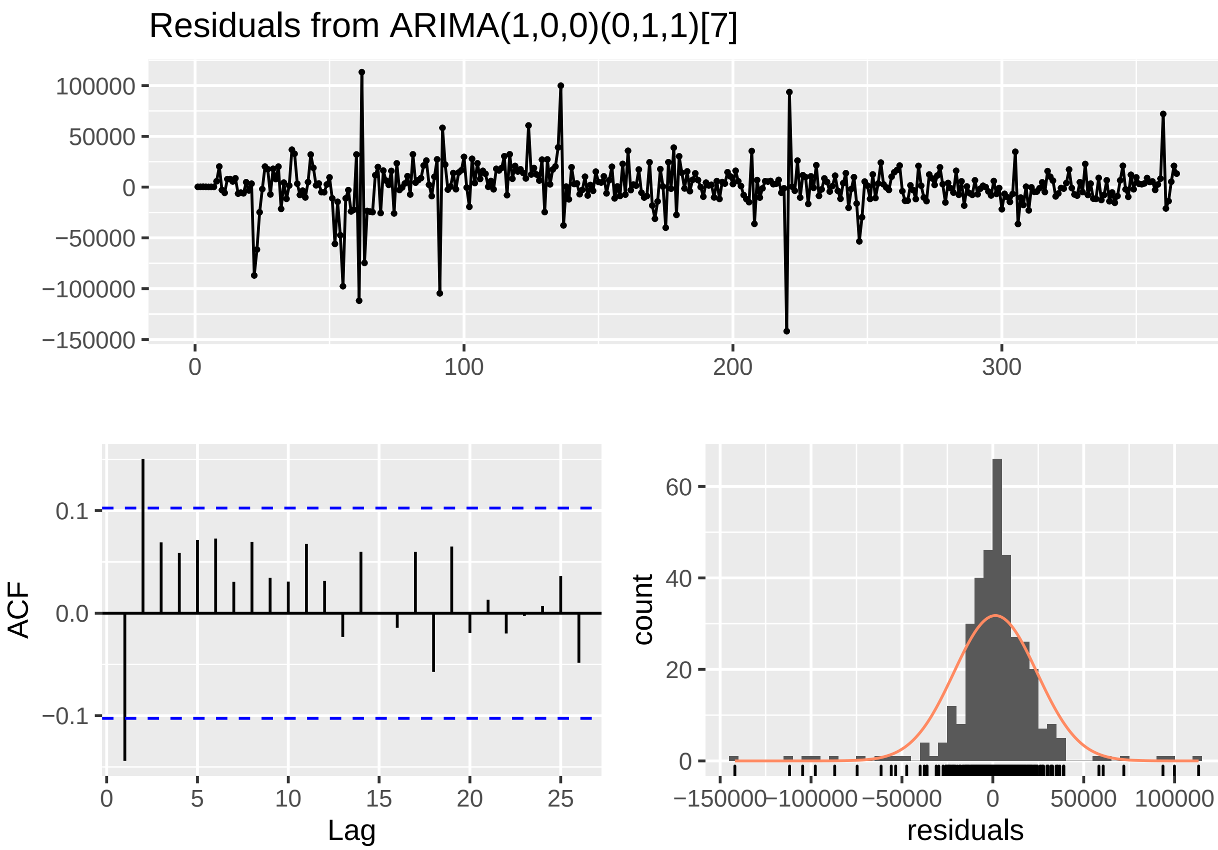
```
## Series: train
## ARIMA(2,0,0)(0,1,1)[7]
##
## Coefficients:
##          ar1     ar2     sma1
##       0.4347  0.2456  -0.9918
## s.e.  0.0513  0.0513   0.1406
##
## sigma^2 estimated as 512850697:  log likelihood=-4109
## AIC=8225.99   AICc=8226.1   BIC=8241.51
```

```r
#ARIMA(2,0,1)
res_3<- auto.arima(train, D=1, seasonal = TRUE, trace = TRUE, method = "ML")
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2) with non-zero mean : Inf
##  ARIMA(0,0,0) with non-zero mean : 8958.549
##  ARIMA(1,0,0) with non-zero mean : 8843.946
##  ARIMA(0,0,1) with non-zero mean : 8799.564
##  ARIMA(0,0,0) with zero mean     : 10214.15
##  ARIMA(1,0,1) with non-zero mean : 8789.544
##  ARIMA(2,0,1) with non-zero mean : 8746.774
##  ARIMA(2,0,0) with non-zero mean : 8752.326
##  ARIMA(3,0,1) with non-zero mean : 8748.536
##  ARIMA(1,0,2) with non-zero mean : 8778.918
##  ARIMA(3,0,0) with non-zero mean : 8748.287
##  ARIMA(3,0,2) with non-zero mean : 8750.71
##  ARIMA(2,0,1) with zero mean     : 8928.314
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(2,0,1) with non-zero mean : 8746.774
##
##  Best model: ARIMA(2,0,1) with non-zero mean
```

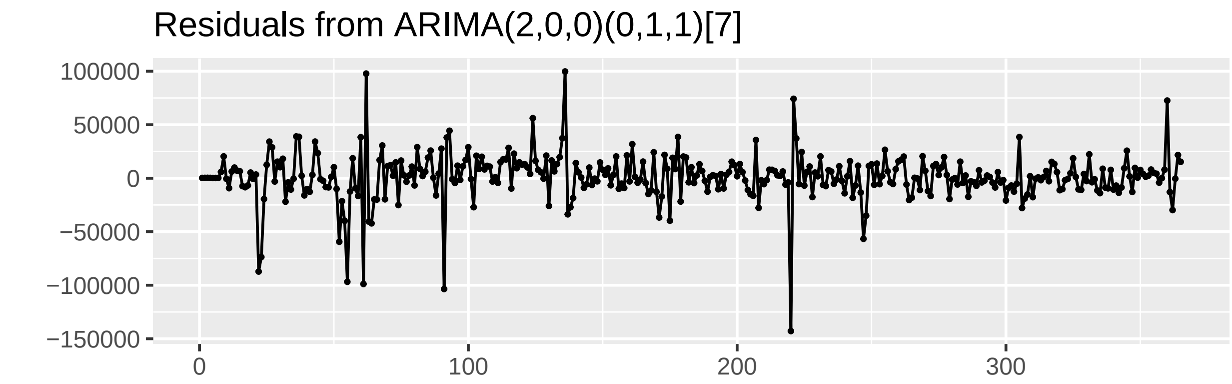
Model Diagnostic

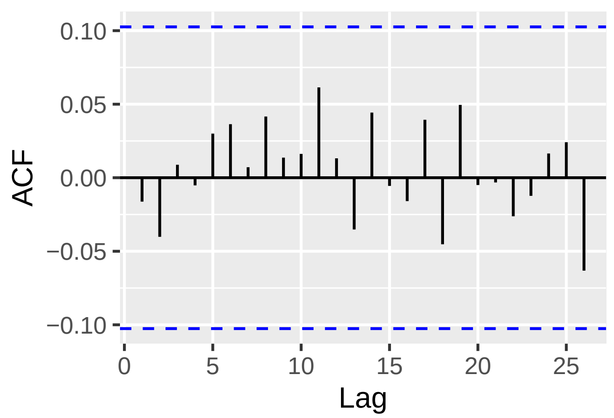
```r
#H0: White Noise
#H1: They exhibit serial correlation
checkresiduals(res_1)
```

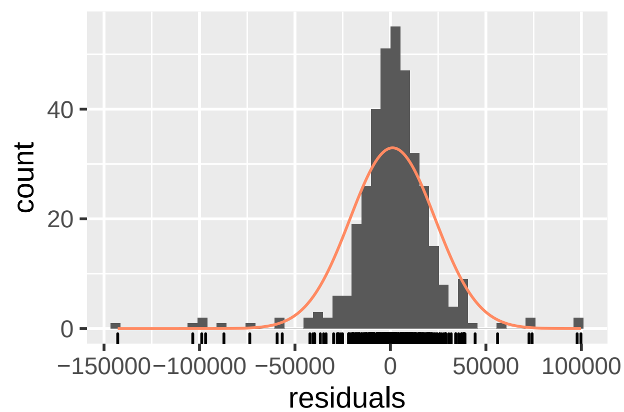



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(0,1,1)[7]
## Q* = 25.873, df = 8, p-value = 0.001104
##
## Model df: 2.   Total lags used: 10
```

```r
checkresiduals(res_2) #White Noise
```

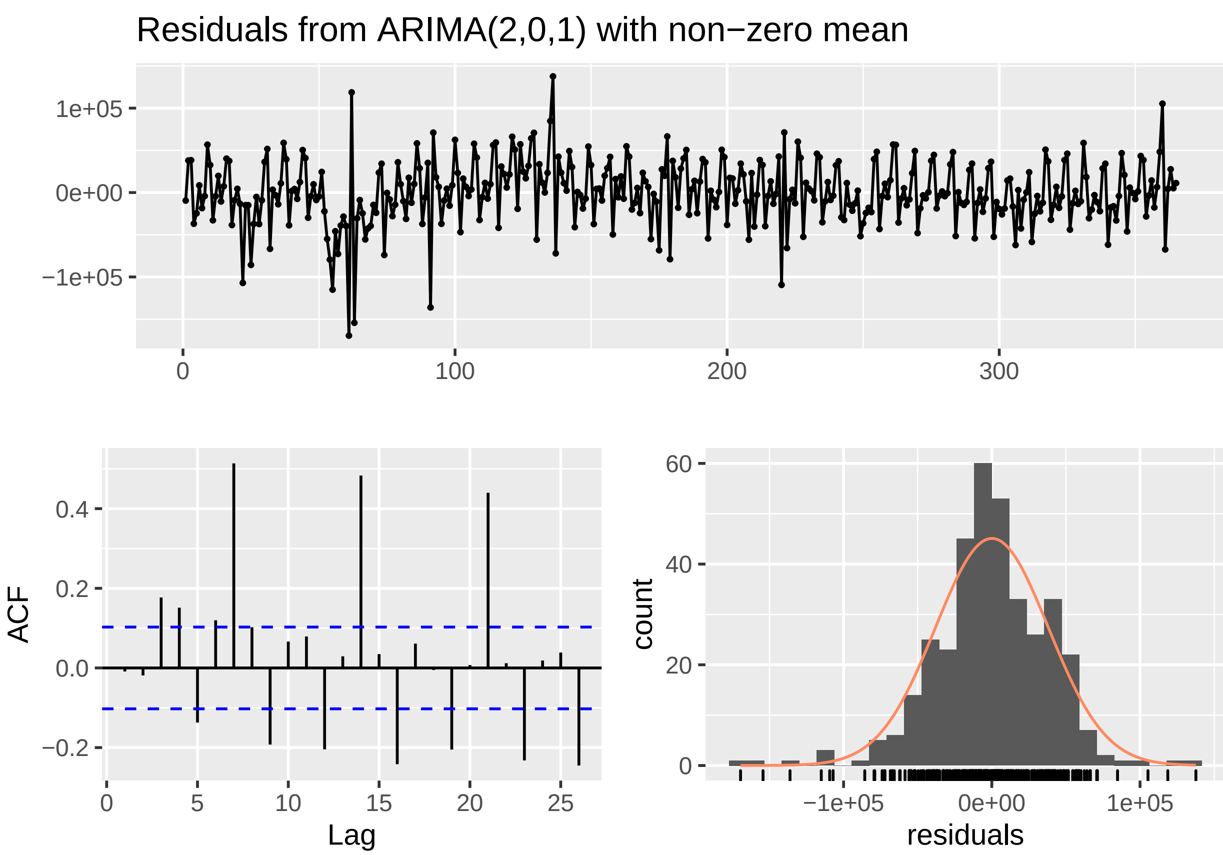







```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,0)(0,1,1)[7]
## Q* = 2.4003, df = 7, p-value = 0.9344
##
## Model df: 3.   Total lags used: 10
```

```
checkresiduals(res_3)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,1) with non-zero mean
## Q* = 150.77, df = 6, p-value < 2.2e-16
##
## Model df: 4.   Total lags used: 10
```

Forecast

Simple Forecasting Methods as a Benchmark

```
train_Mean <- meanf(train, h=61)
train_Naive <- naive(train, h=61)
train_SNaive <- snaive(train, h=61)
train_Drift <- rwf(train, h=61, drift=TRUE)
```

```
(accuracy(train_Mean, test))
```

```
##                          ME     RMSE      MAE       MPE     MAPE     MASE
## Training set -1.483585e-11 51406.57 40426.26 -3.606503 14.96598 1.000000
## Test set     -4.531775e+02 59252.97 45953.46 -5.721919 18.40051 1.136723
```

```
(accuracy(train_Naive, test))
```

```
##                        ME     RMSE      MAE        MPE     MAPE     MASE
## Training set     102.8077 50237.34 39418.91  -1.707504 14.53370 1.000000
## Test set      -25043.1967 64326.28 52639.10 -14.899948 22.31009 1.335377
##                    ACF1
```

```
## Training set 0.1251373
## Test set              NA
```

```
(accuracy(train_SNaive, test))
```

```
##                        ME     RMSE      MAE        MPE     MAPE     MASE
## Training set    102.8077 50237.34 39418.91  -1.707504 14.53370 1.000000
## Test set     -25043.1967 64326.28 52639.10 -14.899948 22.31009 1.335377
##                   ACF1
## Training set 0.1251373
## Test set            NA
```

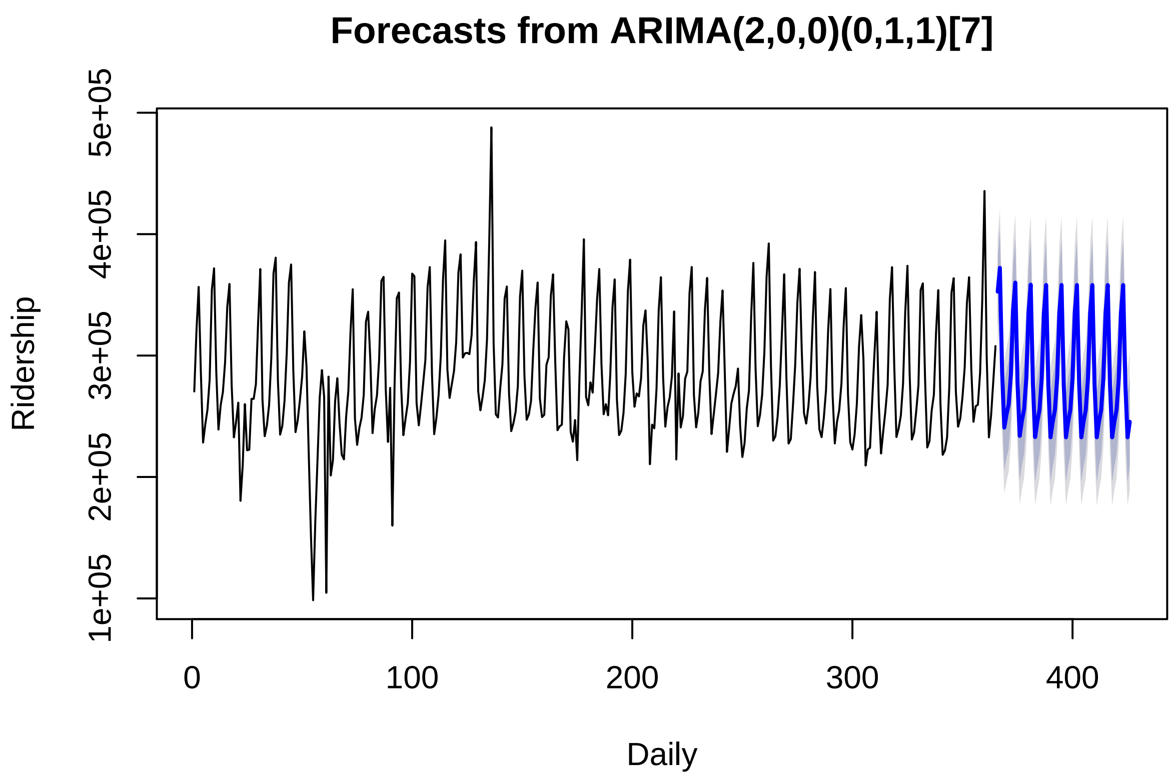
```
(accuracy(train_Drift, test))
```

```
##                          ME     RMSE      MAE        MPE     MAPE      MASE
## Training set  4.478440e-12 50237.24 39384.45  -1.745107 14.52413 0.9991259
## Test set     -2.823024e+04 66098.03 54417.64 -16.146656 23.16252 1.3804956
##                   ACF1
## Training set 0.1251373
## Test set            NA
```

Forecast using Model 2

```
forecast_2 <-forecast(res_2, h=61)
```

```
plot(forecast_2, xlab = "Daily", ylab = "Ridership")
```




```
(accuracy(forecast_2, test))
```

```
##                     ME     RMSE      MAE        MPE      MAPE      MASE
## Training set  1124.973 22333.83 14023.28 -0.5027608  5.536542 0.3557500
## Test set     -2352.365 42979.81 30065.31 -4.1788677 13.118230 0.7627129
##                   ACF1
```

```
## Training set -0.01623781
## Test set               NA
```