Assignment 4: Unemployment and GDP

Joshua Goldberg

May, 02 2019

Data/Objective

The daily data is from Illinois Dept of Transporation (IDOT) for I80E 1EXIT (the 2nd data column) - note each data point is an hourly count of the number of vehicles at a specific location on I80E.

Use the daily data for last 2 weeks of June 2013 to develop an ARIMA forecasting model.

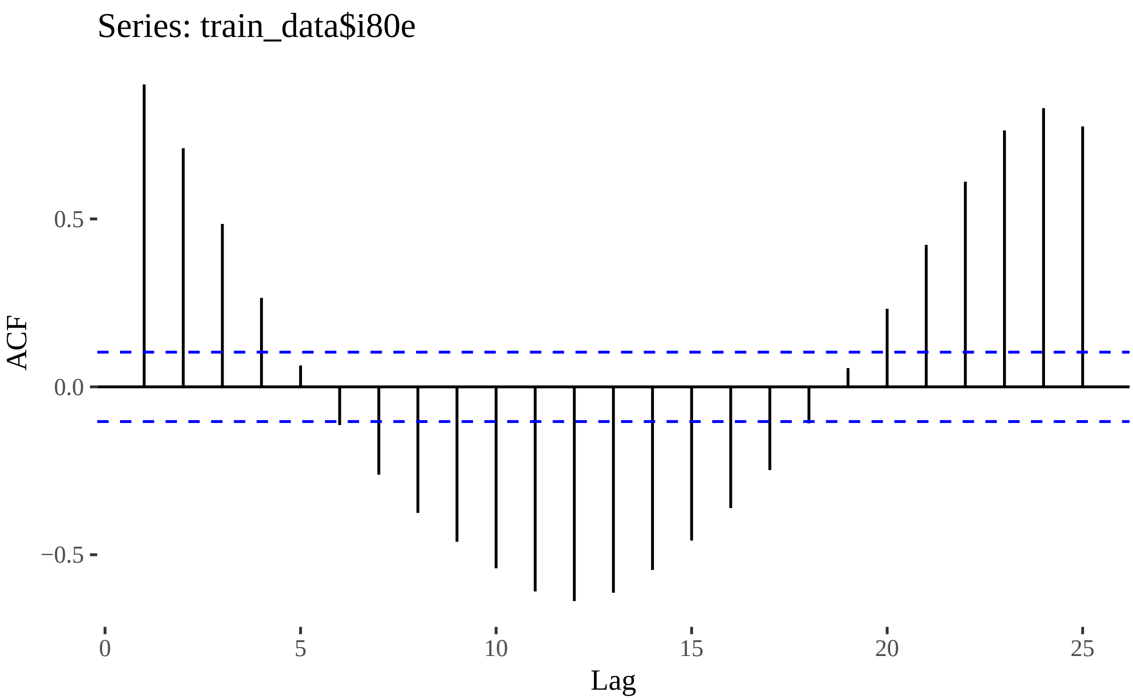
Objective is to forecast the hourly counts for July 1.

The actual data file for July 1 is included for you to test your estimate.

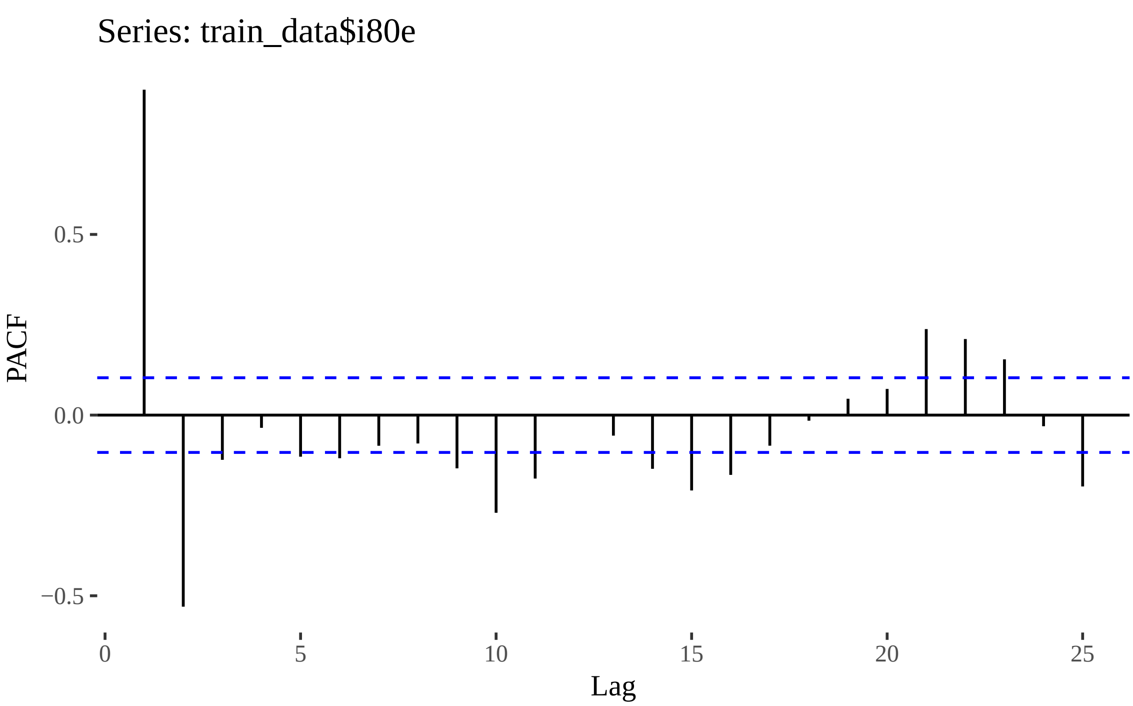
```r
data_files <- data.frame(
  file_name = dir("traffic-flow/"),
  date = ymd(dir("traffic-flow/") %>% str_remove_all("^I-57-|.xls$"))
) %>% slice(-1)

test_files <- data.frame(
  file_name = dir("traffic-flow/"),
  date = ymd(dir("traffic-flow/") %>% str_remove_all("^I-57-|.xls$"))
) %>% slice(1)

extract_excel <- function(file_name, date) {
  readxl::read_xls(
    paste("traffic-flow/", file_name, sep = "/"),
    skip = 2,
    range = cell_cols("C:E")
  ) %>%
    slice(3:(nrow(.) - 2)) %>%
    select(Time, I80E) %>%
    janitor::clean_names() %>%
    mutate(date = date,
           date_time = as.POSIXct(paste(date, time), format = "%Y-%m-%d %H:%M",
                                  tz = Sys.timezone(location = TRUE)))
}

train_data <- pmap_df(data_files, extract_excel) %>%
  mutate_at(vars(i80e), as.numeric) %>%
  as_tsibble(index = "date_time")

test_data <- pmap_df(test_files, extract_excel) %>%
  mutate_at(vars(i80e), as.numeric) %>%
  as_tsibble(index = "date_time")
```

Explore

```r
ggAcf(train_data$i80e)
```




```r
ggPacf(train_data$i80e)
```



Augmented Dickey-Fuller Test

```r
adf.test(train_data$i80e)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  train_data$i80e
## Dickey-Fuller = -8.2071, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

Modeling

Part 1

Use `ARIMA(p, d, q)` model to forecast. Find the model returned by R `auto.arima()`. Change the values of p and q and determine the best model using AICc and BIC. Do AICc and BIC select the same model as the best model?

```
train_auto <- auto.arima(train_data$i80e, seasonal = FALSE)
train_auto
```

```
## Series: train_data$i80e
## ARIMA(3,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3      ma1      mean
##       2.1202  -1.4478  0.2753  -0.9402  746.3552
## s.e.  0.0550   0.1029  0.0543   0.0181    7.7055
##
## sigma^2 estimated as 15246:  log likelihood=-2243.79
## AIC=4499.58   AICc=4499.82   BIC=4522.9
```

Exploring more models.

```
parameters <- list(mod2 = c(2, 0, 3), mod3 = c(3, 0, 3), mod4 =c(3, 0, 2), mod5 = c(3, 0, 1))
models <- map(parameters, ~ Arima(train_data$i80e, order = .x))
models
```

```
## $mod2
## Series: train_data$i80e
## ARIMA(2,0,3) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1      ma2      ma3      mean
##       1.8073  -0.8823  -0.6039  -0.2005  -0.1101  746.3445
## s.e.  0.0295   0.0295   0.0611   0.0599   0.0630    7.6818
##
## sigma^2 estimated as 15187:  log likelihood=-2242.6
## AIC=4499.21   AICc=4499.53   BIC=4526.41
##
## $mod3
## Series: train_data$i80e
## ARIMA(3,0,3) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ar3      ma1      ma2      ma3      mean
##       1.5241  -0.3625  -0.2582  -0.3263  -0.3826  -0.1806  746.3374
```

```
## s.e.  0.3656   0.6719   0.3350   0.3589   0.2518   0.0946    7.6847
##
## sigma^2 estimated as 15202:  log likelihood=-2242.28
## AIC=4500.56   AICc=4500.97   BIC=4531.65
##
## $mod4
## Series: train_data$i80e
## ARIMA(3,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3      ma1      ma2      mean
##       2.0061  -1.2361  0.1690  -0.8155  -0.1148  746.3411
## s.e.  0.1462   0.2722  0.1382   0.1425   0.1295    7.6959
##
## sigma^2 estimated as 15254:  log likelihood=-2243.4
## AIC=4500.79   AICc=4501.11   BIC=4528
##
## $mod5
## Series: train_data$i80e
## ARIMA(3,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3      ma1      mean
##       2.1202  -1.4478  0.2753  -0.9402  746.3552
## s.e.  0.0550   0.1029  0.0543   0.0181    7.7055
##
## sigma^2 estimated as 15246:  log likelihood=-2243.79
## AIC=4499.58   AICc=4499.82   BIC=4522.9
```

AIC and BIC select the same model.

```r
aiccs <- map_dbl(models, "aicc") %>% sort(decreasing = TRUE)
bics <- map_dbl(models, "bic") %>% sort(decreasing = TRUE)

glue::glue("AIC{2:5}: {aiccs}")
```

```
## AIC2: 4501.11219849175
## AIC3: 4500.96850102118
## AIC4: 4499.82166287119
## AIC5: 4499.52630864176
```

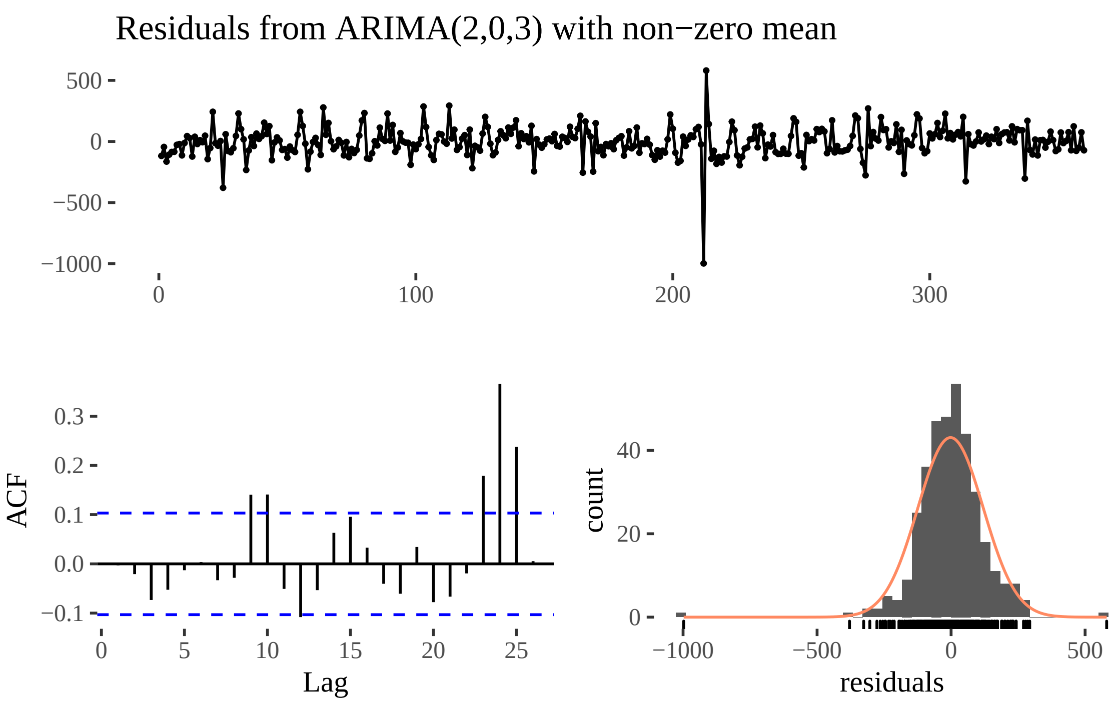
```r
cat("\n")
```

```r
glue::glue("BIC{2:5}: {bics}")
```

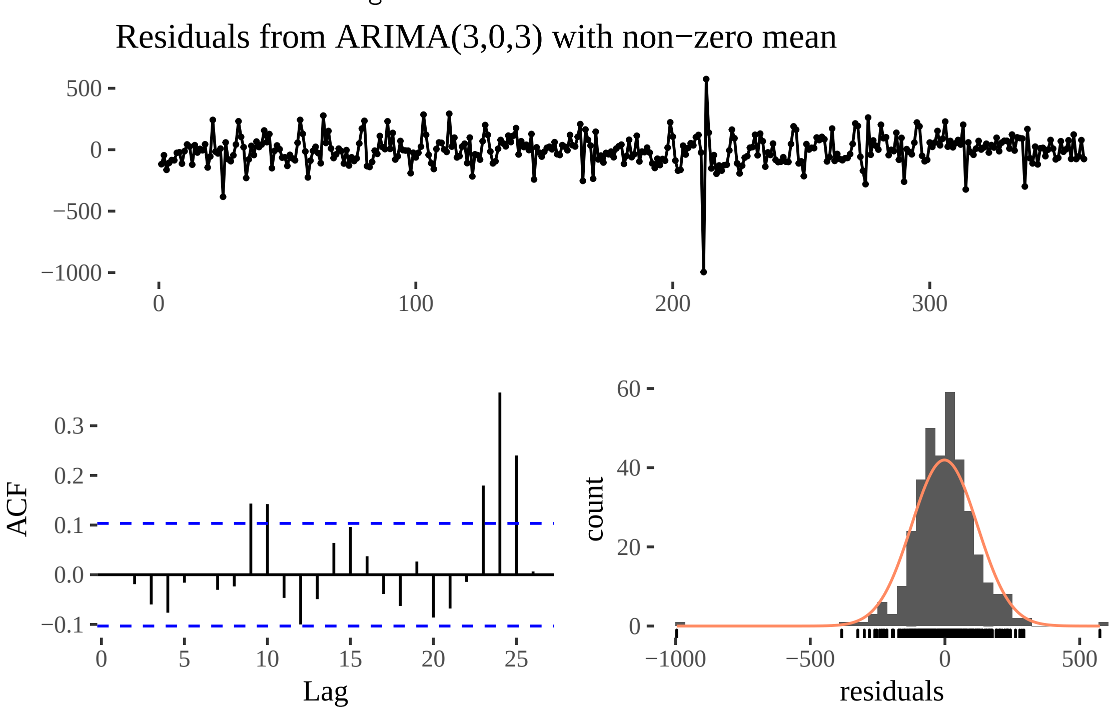
```
## BIC2: 4531.64707686252
## BIC3: 4527.99674489371
## BIC4: 4526.41085504373
## BIC5: 4522.90032671995
```

Examining residuals and normality for each models.

```r
walk(models, ~ checkresiduals(.x))
```

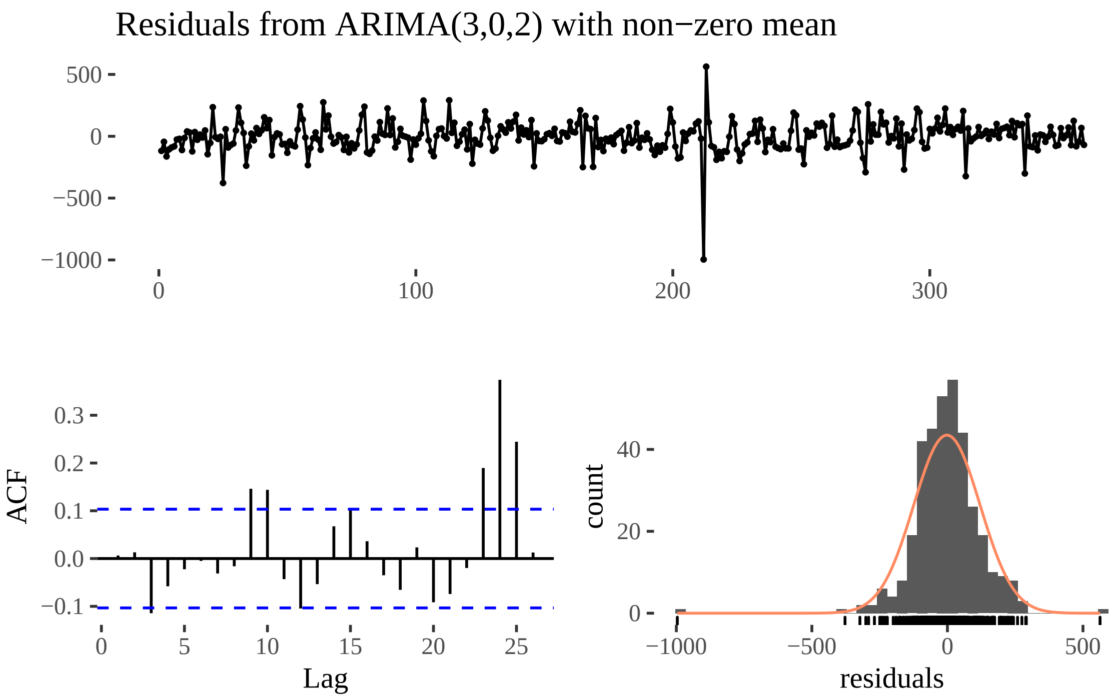



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,3) with non-zero mean
## Q* = 18.659, df = 4, p-value = 0.0009167
##
## Model df: 6.   Total lags used: 10
```

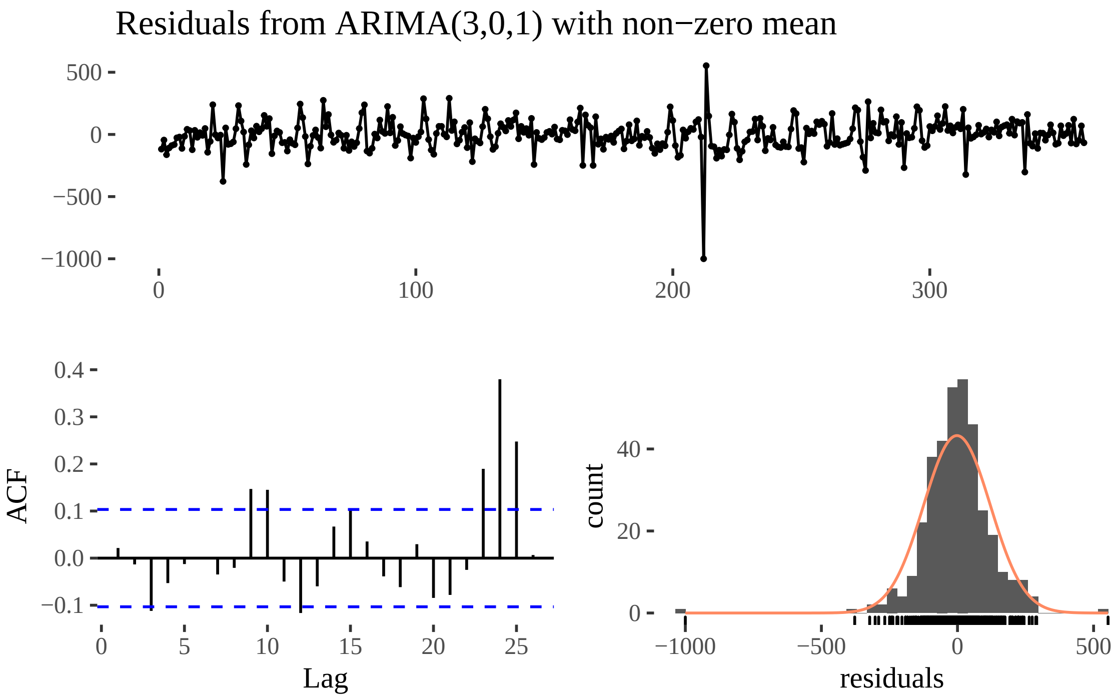



```
##
##  Ljung-Box test
```

```
##
## data:  Residuals from ARIMA(3,0,3) with non-zero mean
## Q* = 19.36, df = 3, p-value = 0.0002303
##
## Model df: 7.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,0,2) with non-zero mean
## Q* = 22.357, df = 4, p-value = 0.0001701
##
## Model df: 6.   Total lags used: 10
```

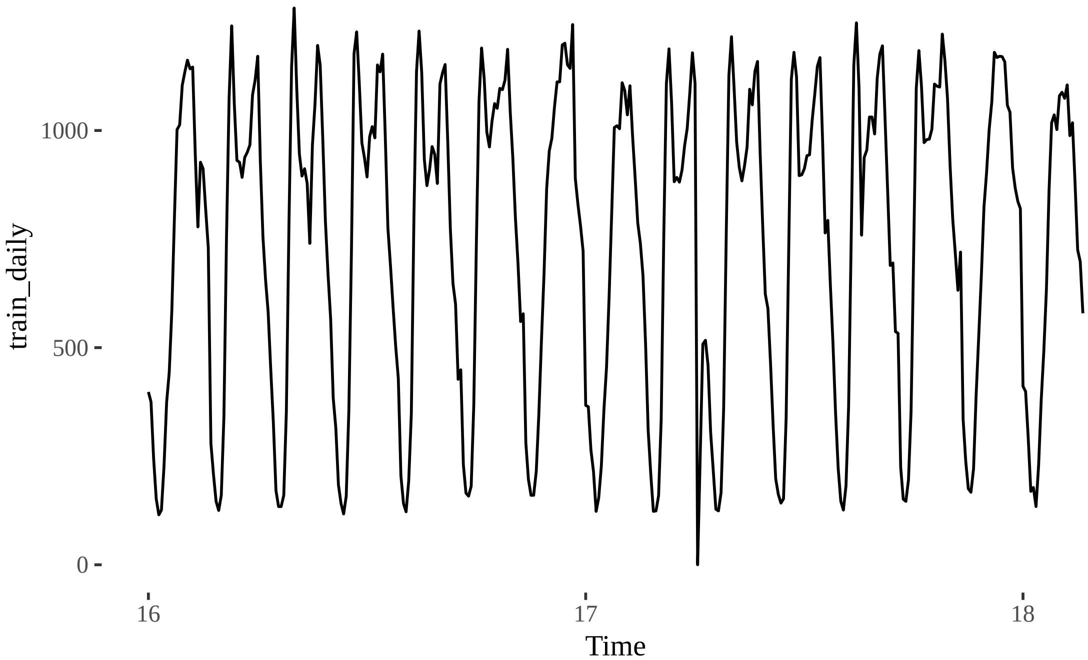



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,0,1) with non-zero mean
## Q* = 22.335, df = 5, p-value = 0.000452
##
## Model df: 5.   Total lags used: 10
```

Part 2

Use day of the week seasonal `ARIMA(p,d,q)(P,Q,D)s` model to forecast for July 1 (which is a Monday) - note use the hourly data.

```
train_daily <- ts(train_data$i80e, start = c(16, 1), frequency = 24 * 7)
autoplot(train_daily)
```

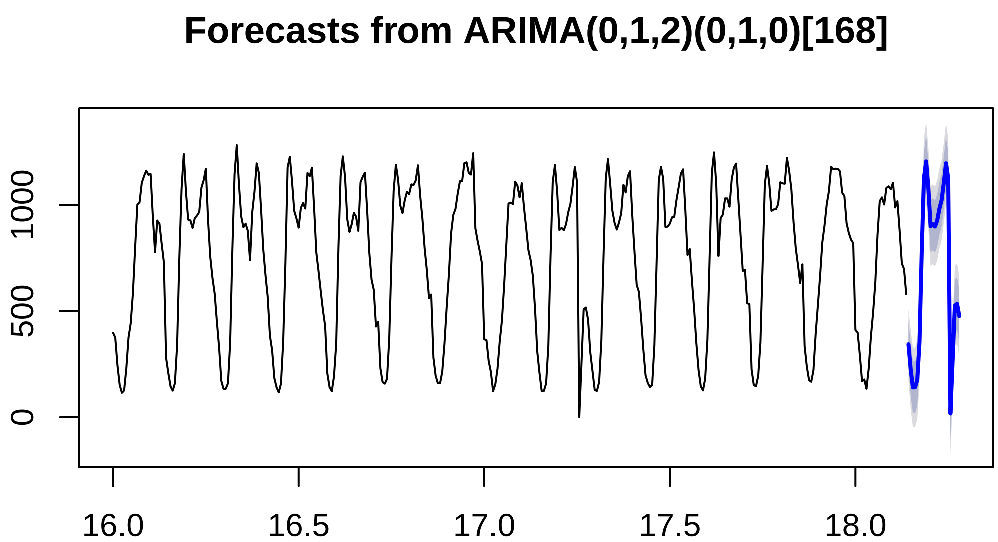



```r
model_day_of_week = auto.arima(train_daily, seasonal = TRUE)
summary(model_day_of_week)
```

```
## Series: train_daily
## ARIMA(0,1,2)(0,1,0)[168]
##
## Coefficients:
##           ma1      ma2
##       -0.4747  -0.4837
## s.e.   0.0609   0.0603
##
## sigma^2 estimated as 7114:  log likelihood=-1122.08
## AIC=2250.16   AICc=2250.29   BIC=2259.92
##
## Training set error measures:
##                    ME     RMSE      MAE  MPE MAPE      MASE       ACF1
## Training set 2.260637 61.11247 24.91202 -Inf  Inf 0.5212061 0.03385938
```

Let's forecast the values for July 1st.

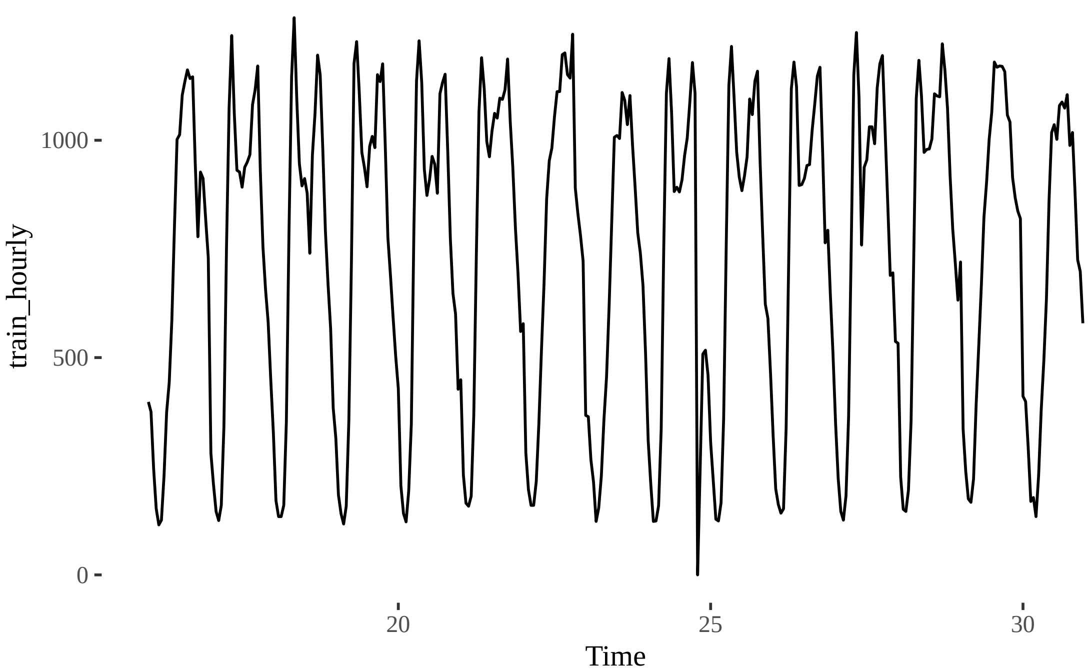
```r
forecast_day_of_week <- forecast(model_day_of_week, h = 24)
plot(forecast_day_of_week)
```



Part 3

Use hour of the day seasonal `ARIMA(p,d,q)(P,D,Q)s` model to forecast for the hours 8:00, 9:00, 17:00 and 18:00 on July 1.

```
train_hourly = ts(train_data$i80e, start = c(16, 1), frequency = 24)
autoplot(train_hourly)
```

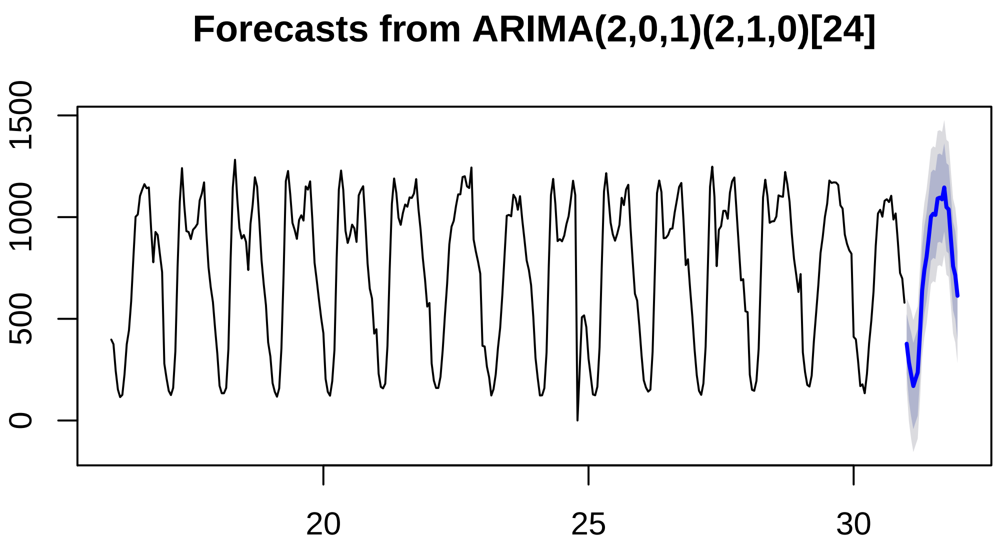



```
model_hourly = auto.arima(train_hourly, seasonal = TRUE)
summary(model_hourly)
```

```
## Series: train_hourly
## ARIMA(2,0,1)(2,1,0)[24]
##
## Coefficients:
```

```
##          ar1      ar2      ma1     sar1     sar2
##       1.4383  -0.6126  -0.6102  -0.3924  -0.3234
## s.e.   0.1627   0.1057   0.1945   0.0586   0.0550
##
## sigma^2 estimated as 12898:  log likelihood=-2068.54
## AIC=4149.09   AICc=4149.34   BIC=4171.99
##
## Training set error measures:
##                    ME     RMSE      MAE  MPE MAPE     MASE        ACF1
## Training set 1.497486 108.8999 68.49439 -Inf  Inf 0.694158 0.008616745
```

```r
forecast_hourly <- forecast(model_hourly, h = 24)
plot(forecast_hourly)
```



Part 4

For the July 1 8:00, 9:00, 17:00 and 18:00 forecasts, which model is better (part 2 or part 3)?

```r
test_data_ts <- ts(test_data[, 3], start = c(16, 1), frequency = 24)
```

```r
indexes <- c(8, 9, 17, 18)
test_data_ts[indexes]
```

```
## [1] 15887 15887 15887 15887
```

```r
forecast_day_of_week$mean[indexes]
```

```
## [1] 1125.378 1205.378 1104.378 1196.378
```

```r
forecast_hourly$mean[indexes]
```

```
## [1]  640.3597  740.1932 1087.4075 1145.7572
```

```r
sse_weekly = sum((forecast_day_of_week$mean[indexes] - test_data_ts[indexes])^2)
sse_hourly = sum((forecast_hourly$mean[indexes] - test_data_ts[indexes])^2)

glue::glue("see_weekly: {sse_weekly}")
```

```
## see_weekly: 867795749.130608
```

```r
glue::glue("sse_hourly: {sse_hourly}")
```

```
## sse_hourly: 898217974.317736
```

Based on SSE, the weekly model performs better.