final project

Loading the libraries

```
library(dplyr)
library(tidyr)
library(readr)
library(lubridate)
library(readxl)
library(ggplot2)
library(knitr)
library(car)
library(corrplot)
library(caret)
library(e1071)
library(prophet)
```

Reading in the data and doing general data wrangling

```
url1 <- "MTA_Daily_Ridership_Data__Beginning_2020.csv"
df <- read_csv(url1)
```

```
url2 <- "weather_nyc_2021_2022.xlsx"
weather_df <- read_excel(url2)
```

```
weather_df <-
  weather_df %>%
  select('datetime', 'tempmax', 'tempmin', 'temp', 'precip', 'snow',
         'snowdepth', 'windspeed', 'conditions', 'icon')
```

```r
# subways data frame
sub_df <-
  df %>%
  # selecting relevant variables
  select('Date', 'Subways: Total Estimated Ridership',
         'Subways: % of Comparable Pre-Pandemic Day') %>%
  na.omit %>%
  # filtering out any dates in the years 2020 and 2023
  filter(!grepl("2023$", Date),
         !grepl("2020$", Date)) %>%
  # mutating date to convert is from a "char" data type
  # creating a new variable that assigns each date their proper day of the week
  mutate("Date" = mdy(Date),
         "Day of Week" = weekdays(Date)) %>%
  select('Day of Week', 'Date', 'Subways: Total Estimated Ridership',
         'Subways: % of Comparable Pre-Pandemic Day')


# joining the weather and subway data frames
sub_df <-
  sub_df %>%
  # joining by the 'date' and 'datetime' variables
  full_join(weather_df, by = c("Date" = "datetime")) %>%
    mutate(Date = as.Date(Date))
```

```r
# buses data frame
bus_df <-
  df %>%
  # selecting relevant variables
  select('Date', 'Buses: Total Estimated Ridership',
         'Buses: % of Comparable Pre-Pandemic Day') %>%
  na.omit() %>%
  # filtering out any dates in the years 2020 and 2023
  filter(!grepl("2023$", Date),
         !grepl("2020$", Date)) %>%
  # mutating date to convert is from a "char" data type
  # creating a new variable that assigns each date their proper day of the week
  mutate("Date" = mdy(Date),
         "Day of Week" = weekdays(Date)) %>%
  select('Day of Week', 'Date', 'Buses: Total Estimated Ridership',
         'Buses: % of Comparable Pre-Pandemic Day')
```

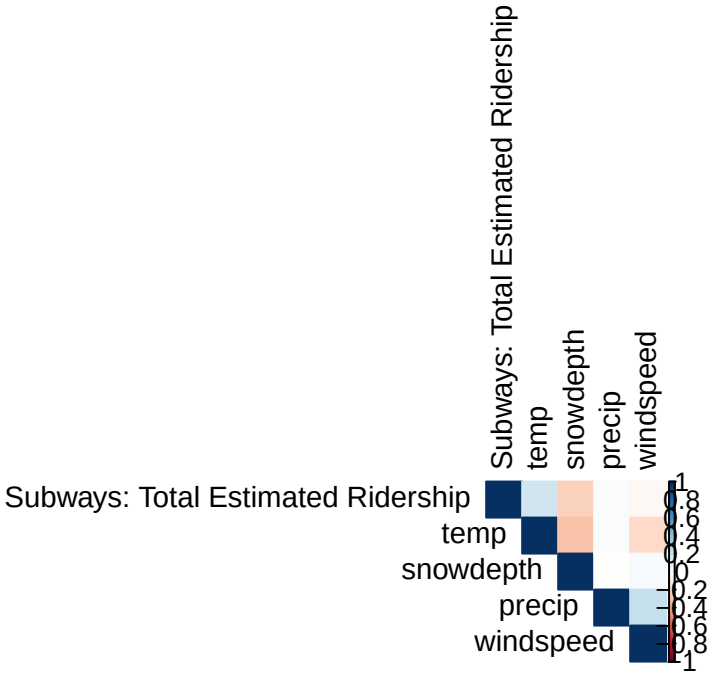
```r
# joining the weather and bus data frames
bus_df <-
  bus_df %>%
  # joining by the 'date' and 'datetime' variables
  full_join(weather_df, by = c("Date" = "datetime")) %>%
  mutate(Date = as.Date(Date))
```

```r
#Creating temperature variable (cat.)
bus_df$temperature <- cut(sub_df$temp,
                          breaks = c(-Inf, 40, 55, 70, 80, Inf),
                          labels = c("Cold", "Cool", "Mild", "Warm", "Hot"))

day_order <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
               "Saturday", "Sunday")
bus_df$`Day of Week` <- factor(bus_df$`Day of Week`, levels = day_order)


sub_df$temperature <- cut(sub_df$temp,
                          breaks = c(-Inf, 40, 55, 70, 80, Inf),
                          labels = c("Cold", "Cool", "Mild", "Warm", "Hot"))

sub_df$`Day of Week` <- factor(sub_df$`Day of Week`, levels = day_order)
```

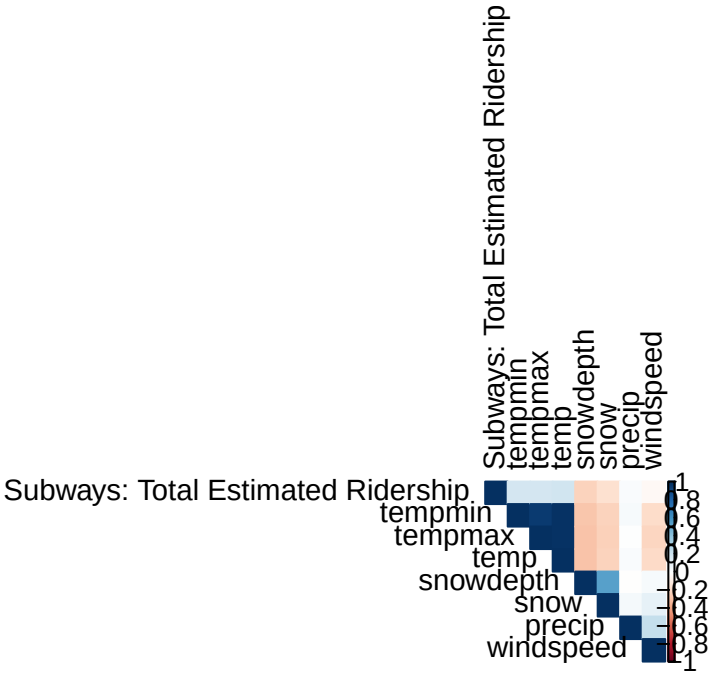
======= ### Creation of corrplots

Correlation plots with full model

```r
#Corrplot with variables we think we want to use
df_subCorr <- sub_df %>%
  select(`Subways: Total Estimated Ridership`, temp, precip, snowdepth,
         windspeed) %>%
  cor()
# creating corrplot
corrplot(df_subCorr, method = 'color', tl.cex = 0.9, tl.col = 'black',
order = 'hclust', type = 'upper')
```

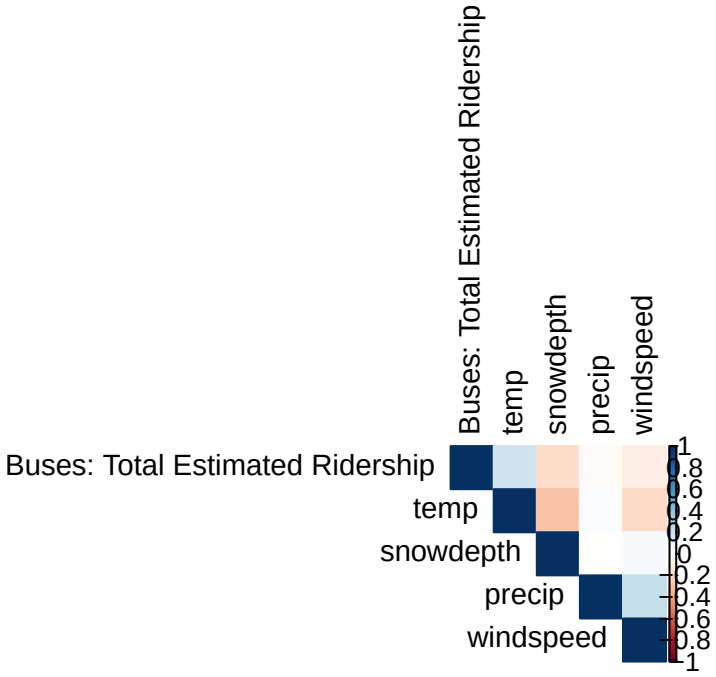



```
#Corrplot with all numeric variables
df_subCorr2 <- sub_df %>%
  select(`Subways: Total Estimated Ridership`, tempmin, tempmax, temp, precip,
         snowdepth, snow, windspeed) %>%
  cor()
# creating corrplot
corrplot(df_subCorr2, method = 'color', tl.cex = 0.9, tl.col = 'black',
order = 'hclust', type = 'upper')
```

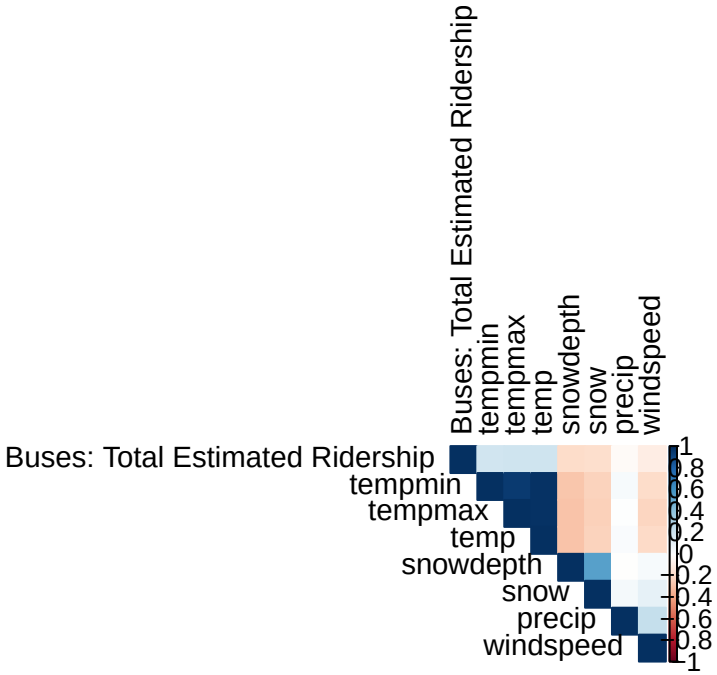


Correlation plots with reduced amount of variables

```
#Corrplot with variables we think we want to use
df_busCorr <- bus_df %>%
  select(`Buses: Total Estimated Ridership`, temp, precip, snowdepth,
         windspeed) %>%
  cor()
# creating corrplot
corrplot(df_busCorr, method = 'color', tl.cex = 0.9, tl.col = 'black',
order = 'hclust', type = 'upper')
```




```
#Corrplot with all numeric variables
df_busCorr2 <- bus_df %>%
  select(`Buses: Total Estimated Ridership`, tempmin, tempmax, temp, precip,
         snowdepth, snow, windspeed) %>%
  cor()
# creating corrplot
corrplot(df_busCorr2, method = 'color', tl.cex = 0.9, tl.col = 'black',
order = 'hclust', type = 'upper')
```



Variance Inflation Factor (iteration 1)

Subways

```
#VIF
library(car)

# Removing date (wrong type: character)
# Removing conditions (too correlated with icon and would rather use icon)
vif_sub_df <-
  sub_df %>%
  select(-Date,
         -conditions)


full_model_sub2 <- lm(`Subways: Total Estimated Ridership` ~ ., data = vif_sub_df)


vif(full_model_sub2) %>% knitr::kable(caption = "Subways")
```

Table 1: Subways

	GVIF	Df	GVIF^(1/(2*Df))
Day of Week	1.234096	6	1.017683
Subways: % of Comparable Pre-Pandemic Day	1.274632	1	1.128996
tempmax	98.108933	1	9.904995
tempmin	90.337361	1	9.504597
temp	326.632763	1	18.072984
precip	1.220347	1	1.104693
snow	2.175639	1	1.475005
snowdepth	1.610135	1	1.268911
windspeed	1.203030	1	1.096827
icon	3.255796	4	1.158996
temperature	18.673508	4	1.441794

Buses

```
vif_bus_df <-
  bus_df %>%
  select(-Date,
         -conditions)

full_model_bus2 <- lm(`Buses: Total Estimated Ridership` ~ ., data = vif_bus_df)

vif(full_model_bus2) %>% knitr::kable(caption = "Buses")
```

Table 2: Buses

	GVIF	Df	GVIF^(1/(2*Df))
Day of Week	1.121389	6	1.009593
Buses: % of Comparable Pre-Pandemic Day	1.248714	1	1.117459
tempmax	96.969904	1	9.847330
tempmin	90.313514	1	9.503342
temp	325.206197	1	18.033474
precip	1.224341	1	1.106499
snow	2.184826	1	1.478116
snowdepth	1.607460	1	1.267856
windspeed	1.201331	1	1.096053
icon	3.239809	4	1.158284
temperature	18.849109	4	1.443482

We can see here that the differetn temp variables are heavily influenced by each other. They have vif values above 5 so there is multicollinearity. We will pick temp(cont) or temperature(cat). We will look at temp(cont) for now

VARIABLE SELECTION

Subway

```
# Removing temperature (split into breaks / categorical)
# Removing conditions (similar to icon, icon is better) and date (character)
sub_df2 <-
  sub_df %>%
  select(-conditions,
         -temperature,
         -Date)

#building full model without the variables temperature (categorical) and conditions
full_model_sub <- lm(`Subways: Total Estimated Ridership` ~ ., data = sub_df2)
summary(full_model_sub)
```

```
Call:
lm(formula = `Subways: Total Estimated Ridership` ~ ., data = sub_df2)

Residuals:
     Min       1Q   Median       3Q      Max
-2168947  -143839    30421   224450   583381

Coefficients:
                                            Estimate Std. Error t value
(Intercept)                                   651664     103277   6.310
`Day of Week`Tuesday                          266275      51463   5.174
`Day of Week`Wednesday                        293446      51638   5.683
`Day of Week`Thursday                         227482      51588   4.410
`Day of Week`Friday                           141690      51416   2.756
`Day of Week`Saturday                        -993131      52369 -18.964
`Day of Week`Sunday                         -1406594      52662 -26.710
`Subways: % of Comparable Pre-Pandemic Day`  3505615     118132  29.675
tempmax                                       -12037       7555  -1.593
tempmin                                        -6068       7844  -0.774
```

```
temp                                          20268      14450   1.403
precip                                       -38081      74075  -0.514
snow                                          36163      54420   0.665
snowdepth                                    -47973      17139  -2.799
windspeed                                     3664       3550   1.032
iconcloudy                                   -87102      96670  -0.901
iconpartly-cloudy-day                         -3349      40821  -0.082
iconrain                                     -20871      38168  -0.547
iconsnow                                     -17533     101872  -0.172
                                            Pr(>|t|)
(Intercept)                                 4.91e-10 ***
`Day of Week`Tuesday                        2.98e-07 ***
`Day of Week`Wednesday                      1.93e-08 ***
`Day of Week`Thursday                       1.20e-05 ***
`Day of Week`Friday                          0.00601 **
`Day of Week`Saturday                        < 2e-16 ***
`Day of Week`Sunday                          < 2e-16 ***
`Subways: % of Comparable Pre-Pandemic Day`  < 2e-16 ***
tempmax                                      0.11156
tempmin                                      0.43945
temp                                         0.16116
precip                                       0.60734
snow                                         0.50658
snowdepth                                    0.00526 **
windspeed                                    0.30228
iconcloudy                                   0.36788
iconpartly-cloudy-day                        0.93463
iconrain                                     0.58468
iconsnow                                     0.86340
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 369300 on 711 degrees of freedom
Multiple R-squared:  0.7852,    Adjusted R-squared:  0.7797
F-statistic: 144.4 on 18 and 711 DF,  p-value: < 2.2e-16
```

```r
#perform backwards variable selection
backward_model_sub <- step(full_model_sub, direction = "backward", scope=formula(full_mode
```

```
Start:  AIC=18734.83
`Subways: Total Estimated Ridership` ~ `Day of Week` + `Subways: % of Comparable Pre-Pandemi
```

```
    tempmax + tempmin + temp + precip + snow + snowdepth + windspeed +
    icon

                                          Df  Sum of Sq        RSS   AIC
- icon                                     4 1.4320e+11 9.7086e+13 18728
- precip                                   1 3.6036e+10 9.6979e+13 18733
- snow                                     1 6.0207e+10 9.7003e+13 18733
- tempmin                                  1 8.1590e+10 9.7025e+13 18733
- windspeed                                1 1.4530e+11 9.7088e+13 18734
<none>                                                 9.6943e+13 18735
- temp                                     1 2.6825e+11 9.7211e+13 18735
- tempmax                                  1 3.4608e+11 9.7289e+13 18735
- snowdepth                                1 1.0682e+12 9.8011e+13 18741
- `Subways: % of Comparable Pre-Pandemic Day`  1 1.2007e+14 2.1701e+14 19321
- `Day of Week`                            6 2.7097e+14 3.6791e+14 19697

Step:  AIC=18727.91
`Subways: Total Estimated Ridership` ~ `Day of Week` + `Subways: % of Comparable Pre-Pandemic
    tempmax + tempmin + temp + precip + snow + snowdepth + windspeed

                                          Df  Sum of Sq        RSS   AIC
- precip                                   1 6.6933e+10 9.7153e+13 18726
- snow                                     1 8.5240e+10 9.7171e+13 18727
- tempmin                                  1 9.7814e+10 9.7184e+13 18727
- windspeed                                1 1.4825e+11 9.7234e+13 18727
- temp                                     1 2.5749e+11 9.7344e+13 18728
<none>                                                 9.7086e+13 18728
- tempmax                                  1 3.0652e+11 9.7393e+13 18728
- snowdepth                                1 1.1097e+12 9.8196e+13 18734
- `Subways: % of Comparable Pre-Pandemic Day`  1 1.2355e+14 2.2063e+14 19325
- `Day of Week`                            6 2.7274e+14 3.6983e+14 19692

Step:  AIC=18726.41
`Subways: Total Estimated Ridership` ~ `Day of Week` + `Subways: % of Comparable Pre-Pandemic
    tempmax + tempmin + temp + snow + snowdepth + windspeed

                                          Df  Sum of Sq        RSS   AIC
- snow                                     1 7.9350e+10 9.7232e+13 18725
- tempmin                                  1 9.2649e+10 9.7246e+13 18725
- windspeed                                1 1.1166e+11 9.7265e+13 18725
- temp                                     1 2.4064e+11 9.7394e+13 18726
<none>                                                 9.7153e+13 18726
- tempmax                                  1 2.8407e+11 9.7437e+13 18727
```

```
- snowdepth                                        1 1.0999e+12 9.8253e+13 18733
- `Subways: % of Comparable Pre-Pandemic Day`      1 1.2367e+14 2.2082e+14 19324
- `Day of Week`                                    6 2.7295e+14 3.7010e+14 19691

Step:  AIC=18725.01
`Subways: Total Estimated Ridership` ~ `Day of Week` + `Subways: % of Comparable Pre-Pandemic
    tempmax + tempmin + temp + snowdepth + windspeed

                                                  Df  Sum of Sq        RSS   AIC
- tempmin                                          1 9.5946e+10 9.7328e+13 18724
- windspeed                                        1 1.2916e+11 9.7362e+13 18724
- temp                                             1 2.4731e+11 9.7480e+13 18725
<none>                                                         9.7232e+13 18725
- tempmax                                          1 2.9389e+11 9.7526e+13 18725
- snowdepth                                        1 1.0995e+12 9.8332e+13 18731
- `Subways: % of Comparable Pre-Pandemic Day`      1 1.2382e+14 2.2105e+14 19323
- `Day of Week`                                    6 2.7328e+14 3.7051e+14 19690

Step:  AIC=18723.73
`Subways: Total Estimated Ridership` ~ `Day of Week` + `Subways: % of Comparable Pre-Pandemic
    tempmax + temp + snowdepth + windspeed

                                                  Df  Sum of Sq        RSS   AIC
- windspeed                                        1 1.5434e+11 9.7483e+13 18723
- tempmax                                          1 2.1504e+11 9.7543e+13 18723
<none>                                                         9.7328e+13 18724
- temp                                             1 3.1572e+11 9.7644e+13 18724
- snowdepth                                        1 1.1010e+12 9.8429e+13 18730
- `Subways: % of Comparable Pre-Pandemic Day`      1 1.2472e+14 2.2205e+14 19324
- `Day of Week`                                    6 2.7509e+14 3.7242e+14 19691

Step:  AIC=18722.89
`Subways: Total Estimated Ridership` ~ `Day of Week` + `Subways: % of Comparable Pre-Pandemic
    tempmax + temp + snowdepth

                                                  Df  Sum of Sq        RSS   AIC
<none>                                                         9.7483e+13 18723
- tempmax                                          1 2.7101e+11 9.7754e+13 18723
- temp                                             1 3.6476e+11 9.7847e+13 18724
- snowdepth                                        1 1.1204e+12 9.8603e+13 18729
- `Subways: % of Comparable Pre-Pandemic Day`      1 1.2512e+14 2.2260e+14 19324
- `Day of Week`                                    6 2.7527e+14 3.7275e+14 19690
```

```r
summary(backward_model_sub)
```

```
Call:
lm(formula = `Subways: Total Estimated Ridership` ~ `Day of Week` +
    `Subways: % of Comparable Pre-Pandemic Day` + tempmax + temp +
    snowdepth, data = sub_df2)

Residuals:
     Min       1Q   Median       3Q      Max
-2157778  -144067    29167   222772   573503

Coefficients:
                                          Estimate Std. Error t value
(Intercept)                                 702603      83206   8.444
`Day of Week`Tuesday                        262790      51087   5.144
`Day of Week`Wednesday                      289471      51134   5.661
`Day of Week`Thursday                       226466      51137   4.429
`Day of Week`Friday                         142447      51042   2.791
`Day of Week`Saturday                      -995143      52087 -19.106
`Day of Week`Sunday                       -1406863      52250 -26.925
`Subways: % of Comparable Pre-Pandemic Day`  3525430     116051  30.378
tempmax                                       -7057       4991  -1.414
temp                                           8750       5334   1.640
snowdepth                                    -41715      14511  -2.875
                                          Pr(>|t|)
(Intercept)                                < 2e-16 ***
`Day of Week`Tuesday                      3.47e-07 ***
`Day of Week`Wednesday                    2.17e-08 ***
`Day of Week`Thursday                     1.10e-05 ***
`Day of Week`Friday                        0.00540 **
`Day of Week`Saturday                      < 2e-16 ***
`Day of Week`Sunday                        < 2e-16 ***
`Subways: % of Comparable Pre-Pandemic Day`  < 2e-16 ***
tempmax                                    0.15785
temp                                       0.10140
snowdepth                                  0.00416 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 368200 on 719 degrees of freedom
Multiple R-squared:  0.784, Adjusted R-squared:  0.781
```

```
F-statistic: 260.9 on 10 and 719 DF,  p-value: < 2.2e-16
```

Bus

```
#Doing the same thing with buses
bus_df2 <-
  bus_df %>%
  select(-conditions,
         -temperature,
         -Date)

full_model_bus <- lm(`Buses: Total Estimated Ridership` ~ ., data = bus_df2)
summary(full_model_bus)
```

```
Call:
lm(formula = `Buses: Total Estimated Ridership` ~ ., data = bus_df2)

Residuals:
    Min      1Q  Median      3Q     Max
-714049  -45112    3428   77574  269922

Coefficients:
                                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                               477510.4    49561.5   9.635  < 2e-16
`Day of Week`Tuesday                       98823.6    19609.3   5.040 5.92e-07
`Day of Week`Wednesday                    110190.5    19668.2   5.602 3.02e-08
`Day of Week`Thursday                      77318.4    19643.3   3.936 9.09e-05
`Day of Week`Friday                        35032.0    19570.9   1.790  0.07388
`Day of Week`Saturday                    -396551.8    19572.1 -20.261  < 2e-16
`Day of Week`Sunday                      -585294.1    19599.2 -29.863  < 2e-16
`Buses: % of Comparable Pre-Pandemic Day` 1031196.8    64665.1  15.947  < 2e-16
tempmax                                    -2742.8     2863.0  -0.958  0.33838
tempmin                                    -2371.9     2988.2  -0.794  0.42762
temp                                        6971.3     5489.9   1.270  0.20456
precip                                    -42347.2    28266.3  -1.498  0.13454
snow                                       -8674.9    20774.2  -0.418  0.67638
snowdepth                                 -17258.7     6528.4  -2.644  0.00838
windspeed                                   1081.4     1351.4   0.800  0.42386
iconcloudy                                -33752.4    36736.2  -0.919  0.35852
iconpartly-cloudy-day                       -167.4    15553.0  -0.011  0.99141
iconrain                                  -14538.0    14532.7  -1.000  0.31747
```

```
iconsnow                                    3031.4    38827.9   0.078  0.93779

(Intercept)                                ***
`Day of Week`Tuesday                       ***
`Day of Week`Wednesday                     ***
`Day of Week`Thursday                      ***
`Day of Week`Friday                        .
`Day of Week`Saturday                      ***
`Day of Week`Sunday                        ***
`Buses: % of Comparable Pre-Pandemic Day` ***
tempmax
tempmin
temp
precip
snow
snowdepth                                  **
windspeed
iconcloudy
iconpartly-cloudy-day
iconrain
iconsnow
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 140700 on 711 degrees of freedom
Multiple R-squared:  0.8102,    Adjusted R-squared:  0.8054
F-statistic: 168.6 on 18 and 711 DF,  p-value: < 2.2e-16
```

```r
backward_model_bus <- step(full_model_bus, direction = "backward", scope=formula(full_mode
```

```
Start:  AIC=17326.02
`Buses: Total Estimated Ridership` ~ `Day of Week` + `Buses: % of Comparable Pre-Pandemic Da
    tempmax + tempmin + temp + precip + snow + snowdepth + windspeed +
    icon

                                           Df  Sum of Sq        RSS   AIC
- icon                                      4 4.0992e+10 1.4114e+13 17320
- snow                                      1 3.4513e+09 1.4076e+13 17324
- tempmin                                   1 1.2470e+10 1.4085e+13 17325
- windspeed                                 1 1.2674e+10 1.4085e+13 17325
- tempmax                                   1 1.8166e+10 1.4091e+13 17325
```

```
- temp                                          1 3.1916e+10 1.4105e+13 17326
<none>                                                      1.4073e+13 17326
- precip                                        1 4.4424e+10 1.4117e+13 17326
- snowdepth                                     1 1.3833e+11 1.4211e+13 17331
- `Buses: % of Comparable Pre-Pandemic Day`     1 5.0333e+12 1.9106e+13 17547
- `Day of Week`                                 6 4.7866e+13 6.1939e+13 18396

Step:  AIC=17320.14
`Buses: Total Estimated Ridership` ~ `Day of Week` + `Buses: % of Comparable Pre-Pandemic Day
    tempmax + tempmin + temp + precip + snow + snowdepth + windspeed

                                               Df  Sum of Sq        RSS   AIC
- snow                                          1 1.6313e+09 1.4115e+13 17318
- windspeed                                     1 1.1071e+10 1.4125e+13 17319
- tempmax                                       1 1.4327e+10 1.4128e+13 17319
- tempmin                                       1 1.7199e+10 1.4131e+13 17319
- temp                                          1 3.1359e+10 1.4145e+13 17320
<none>                                                      1.4114e+13 17320
- precip                                        1 7.0496e+10 1.4184e+13 17322
- snowdepth                                     1 1.3727e+11 1.4251e+13 17325
- `Buses: % of Comparable Pre-Pandemic Day`     1 5.2084e+12 1.9322e+13 17547
- `Day of Week`                                 6 4.7961e+13 6.2075e+13 18389

Step:  AIC=17318.23
`Buses: Total Estimated Ridership` ~ `Day of Week` + `Buses: % of Comparable Pre-Pandemic Day
    tempmax + tempmin + temp + precip + snowdepth + windspeed

                                               Df  Sum of Sq        RSS   AIC
- windspeed                                     1 1.0538e+10 1.4126e+13 17317
- tempmax                                       1 1.4067e+10 1.4129e+13 17317
- tempmin                                       1 1.6990e+10 1.4132e+13 17317
- temp                                          1 3.1046e+10 1.4146e+13 17318
<none>                                                      1.4115e+13 17318
- precip                                        1 7.1171e+10 1.4187e+13 17320
- snowdepth                                     1 1.9949e+11 1.4315e+13 17327
- `Buses: % of Comparable Pre-Pandemic Day`     1 5.3169e+12 1.9432e+13 17550
- `Day of Week`                                 6 4.8036e+13 6.2151e+13 18388

Step:  AIC=17316.77
`Buses: Total Estimated Ridership` ~ `Day of Week` + `Buses: % of Comparable Pre-Pandemic Day
    tempmax + tempmin + temp + precip + snowdepth

                                               Df  Sum of Sq        RSS   AIC
```

```
- tempmax                                    1 1.8133e+10 1.4144e+13 17316
- tempmin                                    1 1.9754e+10 1.4146e+13 17316
- temp                                       1 3.5817e+10 1.4162e+13 17317
<none>                                                   1.4126e+13 17317
- precip                                     1 6.2383e+10 1.4188e+13 17318
- snowdepth                                  1 2.0274e+11 1.4329e+13 17325
- `Buses: % of Comparable Pre-Pandemic Day`  1 5.3105e+12 1.9436e+13 17548
- `Day of Week`                              6 4.8254e+13 6.2380e+13 18389

Step:  AIC=17315.71
`Buses: Total Estimated Ridership` ~ `Day of Week` + `Buses: % of Comparable Pre-Pandemic Day
    tempmin + temp + precip + snowdepth

                                            Df  Sum of Sq        RSS   AIC
- tempmin                                    1 3.7978e+09 1.4148e+13 17314
- temp                                       1 3.0532e+10 1.4175e+13 17315
<none>                                                   1.4144e+13 17316
- precip                                     1 5.4802e+10 1.4199e+13 17317
- snowdepth                                  1 2.0024e+11 1.4344e+13 17324
- `Buses: % of Comparable Pre-Pandemic Day`  1 5.3667e+12 1.9511e+13 17549
- `Day of Week`                              6 4.8283e+13 6.2427e+13 18388

Step:  AIC=17313.9
`Buses: Total Estimated Ridership` ~ `Day of Week` + `Buses: % of Comparable Pre-Pandemic Day
    temp + precip + snowdepth

                                            Df  Sum of Sq        RSS   AIC
<none>                                                   1.4148e+13 17314
- precip                                     1 5.6424e+10 1.4204e+13 17315
- snowdepth                                  1 2.0205e+11 1.4350e+13 17322
- temp                                       1 4.6941e+11 1.4617e+13 17336
- `Buses: % of Comparable Pre-Pandemic Day`  1 5.3727e+12 1.9521e+13 17547
- `Day of Week`                              6 4.8382e+13 6.2530e+13 18387
```

```r
summary(backward_model_bus)
```

```
Call:
lm(formula = `Buses: Total Estimated Ridership` ~ `Day of Week` +
    `Buses: % of Comparable Pre-Pandemic Day` + temp + precip +
    snowdepth, data = bus_df2)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-710071  -43347    4643   74982  257244

Coefficients:
                                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                               482471.2    41892.9  11.517  < 2e-16
`Day of Week`Tuesday                       99506.1    19457.5   5.114 4.05e-07
`Day of Week`Wednesday                    110519.1    19478.7   5.674 2.02e-08
`Day of Week`Thursday                      78729.3    19464.4   4.045 5.80e-05
`Day of Week`Friday                        36049.4    19420.1   1.856  0.06382
`Day of Week`Saturday                    -395034.2    19422.1 -20.339  < 2e-16
`Day of Week`Sunday                      -584938.4    19457.1 -30.063  < 2e-16
`Buses: % of Comparable Pre-Pandemic Day` 1046885.2    63355.6  16.524  < 2e-16
temp                                        1613.6      330.4   4.884 1.28e-06
precip                                    -43598.4    25746.6  -1.693  0.09082
snowdepth                                 -17884.6     5581.2  -3.204  0.00141

(Intercept)                               ***
`Day of Week`Tuesday                      ***
`Day of Week`Wednesday                    ***
`Day of Week`Thursday                     ***
`Day of Week`Friday                       .
`Day of Week`Saturday                     ***
`Day of Week`Sunday                       ***
`Buses: % of Comparable Pre-Pandemic Day` ***
temp                                      ***
precip                                    .
snowdepth                                 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 140300 on 719 degrees of freedom
Multiple R-squared:  0.8092,    Adjusted R-squared:  0.8066
F-statistic:   305 on 10 and 719 DF,  p-value: < 2.2e-16
```

VIF (iteration 2)

Going through VIF again this time with less covariates as determined by the variable selection

Subways

```
#Removing date, conditions, and only leaving temp(cont.)
vif_sub_df2 <-
  sub_df %>%
  select(-Date,
         -conditions,
         -tempmin,
         -tempmax,
         -temperature,
         -icon,
         -snow,
         -precip,
         -windspeed)

full_model_sub3 <- lm(`Subways: Total Estimated Ridership`~.,data = vif_sub_df2)
summary(full_model_sub3)
```

```
Call:
lm(formula = `Subways: Total Estimated Ridership` ~ ., data = vif_sub_df2)

Residuals:
     Min       1Q   Median       3Q      Max
-2154079  -146272    28382   219399   577573

Coefficients:
                                            Estimate Std. Error t value
(Intercept)                                 667368.7    79441.2   8.401
`Day of Week`Tuesday                        260925.4    51105.6   5.106
`Day of Week`Wednesday                      287431.2    51148.9   5.620
`Day of Week`Thursday                       226417.3    51172.7   4.425
`Day of Week`Friday                         139597.9    51037.6   2.735
`Day of Week`Saturday                      -998835.4    52057.0 -19.187
`Day of Week`Sunday                       -1408177.9    52278.2 -26.936
`Subways: % of Comparable Pre-Pandemic Day` 3542211.6   115522.3  30.663
temp                                          1306.6      861.6   1.516
snowdepth                                   -40641.6    14501.4  -2.803
                                           Pr(>|t|)
(Intercept)                                2.36e-16 ***
`Day of Week`Tuesday                       4.22e-07 ***
`Day of Week`Wednesday                     2.74e-08 ***
```

```
`Day of Week`Thursday                          1.12e-05 ***
`Day of Week`Friday                             0.00639 **
`Day of Week`Saturday                           < 2e-16 ***
`Day of Week`Sunday                             < 2e-16 ***
`Subways: % of Comparable Pre-Pandemic Day`     < 2e-16 ***
temp                                            0.12984
snowdepth                                       0.00521 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 368500 on 720 degrees of freedom
Multiple R-squared:  0.7834,    Adjusted R-squared:  0.7807
F-statistic: 289.3 on 9 and 720 DF,  p-value: < 2.2e-16
```

```r
vif(full_model_sub3) %>% knitr::kable(caption = "Subways")
```

Table 3: Subways

	GVIF	Df	GVIF^(1/(2*Df))
Day of Week	1.102527	6	1.008167
Subways: % of Comparable Pre-Pandemic Day	1.208726	1	1.099421
temp	1.118419	1	1.057553
snowdepth	1.147487	1	1.071208

Buses

```r
vif_bus_df2 <-
  bus_df %>%
  select(-Date,
         -conditions,
         -tempmin,
         -tempmax,
         -temperature,
         -icon,
         -snow)

full_model_bus3 <- lm(`Buses: Total Estimated Ridership` ~ .,data = vif_bus_df2)
summary(full_model_bus3)
```

```
Call:
lm(formula = `Buses: Total Estimated Ridership` ~ ., data = vif_bus_df2)

Residuals:
    Min      1Q  Median      3Q     Max
-711193  -42844    4808   77797  266197

Coefficients:
                                        Estimate Std. Error t value Pr(>|t|)
(Intercept)                               464981      46642   9.969  < 2e-16
`Day of Week`Tuesday                      100012      19470   5.137 3.61e-07
`Day of Week`Wednesday                    111147      19496   5.701 1.74e-08
`Day of Week`Thursday                      79777      19507   4.090 4.81e-05
`Day of Week`Friday                        35975      19424   1.852  0.06442
`Day of Week`Saturday                    -394591      19433 -20.305  < 2e-16
`Day of Week`Sunday                      -585757      19484 -30.063  < 2e-16
`Buses: % of Comparable Pre-Pandemic Day` 1047320      63370  16.527  < 2e-16
temp                                        1670        337   4.956 9.00e-07
precip                                    -49282      26598  -1.853  0.06432
snowdepth                                 -17746       5585  -3.178  0.00155
windspeed                                   1117       1308   0.854  0.39360

(Intercept)                              ***
`Day of Week`Tuesday                     ***
`Day of Week`Wednesday                   ***
`Day of Week`Thursday                    ***
`Day of Week`Friday                      .
`Day of Week`Saturday                    ***
`Day of Week`Sunday                      ***
`Buses: % of Comparable Pre-Pandemic Day` ***
temp                                     ***
precip                                   .
snowdepth                                **
windspeed
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 140300 on 718 degrees of freedom
Multiple R-squared:  0.8094,    Adjusted R-squared:  0.8065
F-statistic: 277.2 on 11 and 718 DF,  p-value: < 2.2e-16
```

```r
vif(full_model_bus3) %>% knitr::kable(caption = "Buses")
```

Table 4: Buses

	GVIF	Df	GVIF^(1/(2*Df))
Day of Week	1.026623	6	1.002192
Buses: % of Comparable Pre-Pandemic Day	1.179530	1	1.086062
temp	1.180434	1	1.086478
precip	1.082566	1	1.040464
snowdepth	1.173801	1	1.083421
windspeed	1.125354	1	1.060827

We can see all values are below 5 for both subways and buses. This indicates there is no multicollinearity.

Creation of datasets with desired covariates

```r
sub_new <- sub_df %>%
  select(`Subways: Total Estimated Ridership`, `Day of Week`,
         `Subways: % of Comparable Pre-Pandemic Day`,temp, snowdepth)
lm_sub <- lm(`Subways: Total Estimated Ridership` ~ ., data = sub_new)

bus_new <- bus_df %>%
  select(`Buses: Total Estimated Ridership`,`Day of Week`,
         `Buses: % of Comparable Pre-Pandemic Day`, temp, precip, snowdepth)
lm_bus <- lm(`Buses: Total Estimated Ridership` ~ ., data = bus_new)
```

Using K-Fold Cross Validation to split the data

```r
k <- 5

# making folds for subways
make_folds_sub <- function(sub_new, k){
  folds <- sample(1:k, nrow(sub_new), replace = T)
  df_folds <- list()
  for (i in 1:k){
    df_folds[[i]] <- list()
    df_folds[[i]]$train <- sub_new[which(folds != i), ]
```

```
    df_folds[[i]]$test <- sub_new[which(folds == i), ]
  }
  return(df_folds)
}


cv_mspe <- function(formula, df_folds){
  kfold_mspe <- c()
  for (i in 1:length(df_folds)){
    # change model to fit our project
    model <- lm(formula, df_folds[[i]]$train)
    # change model again
    y_hat <- predict(model, df_folds[[i]]$test)
    # change the response from 'medv' to our response
    kfold_mspe[i] <- sqrt(
      mean((y_hat - df_folds[[i]]$test$`Subways: Total Estimated Ridership`)^2)
      )
  }
  return(mean(kfold_mspe))
}

cv_mspe(`Subways: Total Estimated Ridership` ~ ., make_folds_sub(sub_new, k))
```

```
[1] 370036.9
```

```
# making folds for buses
make_folds_bus <- function(bus_new, k){
  folds <- sample(1:k, nrow(bus_new), replace = T)
  df_folds <- list()
  for (i in 1:k){
    df_folds[[i]] <- list()
    df_folds[[i]]$train = bus_new[which(folds != i), ]
    df_folds[[i]]$test = bus_new[which(folds == i), ]
  }
  return(df_folds)
}


cv_mspe <- function(formula, df_folds){
  kfold_mspe <- c()
  for (i in 1:length(df_folds)){
```

```
    # change model to fit our project
    model <- lm(formula, df_folds[[i]]$train)
    # change model again
    y_hat <- predict(model, df_folds[[i]]$test)
    # change the response from 'medv' to our response
    kfold_mspe[i] <- sqrt(
      mean((y_hat - df_folds[[i]]$test$`Subways: Total Estimated Ridership`)^2)
      )
  }
  return(mean(kfold_mspe))
}

rmse <- function(y, yhat) {
  sqrt(mean((y - yhat)^2))
}
```

Time Series K-fold

```
# Set the number of folds for cross-validation

folds <- make_folds_sub(sub_new, k = 10)

temp_list <- list(folds[[1]]$train, folds[[2]]$train, folds[[3]]$train,
                  folds[[4]]$train, folds[[5]]$train)
train_data <- Reduce(function(x, y) merge(x, y, all = TRUE), temp_list)



ctrl <- trainControl(method = "cv", number = 10)

# Define the model to use
model <- train(`Subways: Total Estimated Ridership` ~., data = train_data,
               method = "svmRadial", trControl = ctrl)

# Print the cross-validation results
print(model$results)
```

```
      sigma    C     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
1 0.1233898 0.25 267652.6 0.8728940 130150.7 97960.61 0.08811815 29379.84
2 0.1233898 0.50 261836.1 0.8783395 124782.3 97774.88 0.08469956 28696.04
3 0.1233898 1.00 255801.0 0.8835975 121345.5 96376.07 0.08187832 28103.48
```

Full Bus SVM

```
train1 <- bus_df[1:233, ]
test1 <- bus_df[234:292, ]
model1 <- train(`Buses: Total Estimated Ridership` ~ ., data = train1,
                method = "svmRadial")
m1RMSE <- model1$results$RMSE

train2 <- bus_df[1:320, ]
test2 <- bus_df[320:401, ]
model2 <- train(`Buses: Total Estimated Ridership` ~ ., data = train2,
                method = "svmRadial")
m2RMSE <- model2$results$RMSE


train3 <- bus_df[1:408, ]
test3 <- bus_df[409:511, ]
model3 <- train(`Buses: Total Estimated Ridership` ~ ., data = train3,
                method = "svmRadial")
m3RMSE <- model3$results$RMSE


train4 <- bus_df[1:496, ]
test4 <- bus_df[497:620, ]
model4 <- train(`Buses: Total Estimated Ridership` ~ ., data = train4,
                method = "svmRadial")
m4RMSE <- model4$results$RMSE


train5 <- bus_df[1:584, ]
test5 <- bus_df[585:730, ]
model5 <- train(`Buses: Total Estimated Ridership` ~ ., data = train5,
                method = "svmRadial")
m5RMSE <- model5$results$RMSE

fullBusSVM_RMSE <- (sum(m1RMSE)+sum(m2RMSE)+sum(m3RMSE)+sum(m4RMSE)+sum(m5RMSE))/15
```

Full Sub SVM

```
train1 <- sub_df[1:233, ]
test1 <- sub_df[234:292, ]
model1 <- train(`Subways: Total Estimated Ridership` ~ ., data = train1,
```

```
                method = "svmRadial")
m1RMSE <- model1$results$RMSE

train2 <- sub_df[1:320, ]
test2 <- sub_df[320:401, ]
model2 <- train(`Subways: Total Estimated Ridership` ~ ., data = train2,
                method = "svmRadial")
m2RMSE <- model2$results$RMSE


train3 <- sub_df[1:408, ]
test3 <- sub_df[409:511, ]
model3 <- train(`Subways: Total Estimated Ridership` ~ ., data = train3,
                method = "svmRadial")
m3RMSE <- model3$results$RMSE


train4 <- sub_df[1:496, ]
test4 <- sub_df[497:620, ]
model4 <- train(`Subways: Total Estimated Ridership` ~ ., data = train4,
                method = "svmRadial")
m4RMSE <- model4$results$RMSE


train5 <- sub_df[1:584, ]
test5 <- sub_df[585:730, ]
model5 <- train(`Subways: Total Estimated Ridership` ~ ., data = train5,
                method = "svmRadial")
m5RMSE <- model5$results$RMSE

fullSubSVM_RMSE <- (sum(m1RMSE)+sum(m2RMSE)+sum(m3RMSE)+sum(m4RMSE)+sum(m5RMSE))/15
```

Full Bus LM

```
train1 <- bus_df[1:233, ]
test1 <- bus_df[234:292, ]
model1 <- train(`Buses: Total Estimated Ridership` ~ ., data = train1,
                method = "lm")
m1RMSE <- model1$results$RMSE

train2 <- bus_df[1:320, ]
```

```
test2 <- bus_df[320:401, ]
model2 <- train(`Buses: Total Estimated Ridership` ~ ., data = train2,
                method = "lm")
m2RMSE <- model2$results$RMSE


train3 <- bus_df[1:408, ]
test3 <- bus_df[409:511, ]
model3 <- train(`Buses: Total Estimated Ridership` ~ ., data = train3,
                method = "lm")
m3RMSE <- model3$results$RMSE


train4 <- bus_df[1:496, ]
test4 <- bus_df[497:620, ]
model4 <- train(`Buses: Total Estimated Ridership` ~ ., data = train4,
                method = "lm")
m4RMSE <- model4$results$RMSE


train5 <- bus_df[1:584, ]
test5 <- bus_df[585:730, ]
model5 <- train(`Buses: Total Estimated Ridership` ~ ., data = train5,
                method = "lm")
m5RMSE <- model5$results$RMSE

fullBusLM_RMSE <- (sum(m1RMSE)+sum(m2RMSE)+sum(m3RMSE)+sum(m4RMSE)+sum(m5RMSE))/15
```

Full Sub LM

```
train1 <- sub_df[1:233, ]
test1 <- sub_df[234:292, ]
model1 <- train(`Subways: Total Estimated Ridership` ~ ., data = train1,
                method = "lm")
m1RMSE <- model1$results$RMSE

train2 <- sub_df[1:320, ]
test2 <- sub_df[320:401, ]
model2 <- train(`Subways: Total Estimated Ridership` ~ ., data = train2,
                method = "lm")
m2RMSE <- model2$results$RMSE
```

```
train3 <- sub_df[1:408, ]
test3 <- sub_df[409:511, ]
model3 <- train(`Subways: Total Estimated Ridership` ~ ., data = train3,
                method = "lm")
m3RMSE <- model3$results$RMSE


train4 <- sub_df[1:496, ]
test4 <- sub_df[497:620, ]
model4 <- train(`Subways: Total Estimated Ridership` ~ ., data = train4,
                method = "lm")
m4RMSE <- model4$results$RMSE


train5 <- sub_df[1:584, ]
test5 <- sub_df[585:730, ]
model5 <- train(`Subways: Total Estimated Ridership` ~ ., data = train5,
                method = "lm")
m5RMSE <- model5$results$RMSE

fullSubLM_RMSE <- (sum(m1RMSE)+sum(m2RMSE)+sum(m3RMSE)+sum(m4RMSE)+sum(m5RMSE))/15
```

Bus new SVM

```
train1 <- bus_new[1:233, ]
test1 <- bus_new[234:292, ]
model1 <- train(`Buses: Total Estimated Ridership` ~ ., data = train1,
                method = "svmRadial")
m1RMSE <- model1$results$RMSE

train2 <- bus_new[1:320, ]
test2 <- bus_new[320:401, ]
model2 <- train(`Buses: Total Estimated Ridership` ~ ., data = train2,
                method = "svmRadial")
m2RMSE <- model2$results$RMSE


train3 <- bus_new[1:408, ]
test3 <- bus_new[409:511, ]
model3 <- train(`Buses: Total Estimated Ridership` ~ ., data = train3,
                method = "svmRadial")
```

```
m3RMSE <- model3$results$RMSE


train4 <- bus_new[1:496, ]
test4 <- bus_new[497:620, ]
model4 <- train(`Buses: Total Estimated Ridership` ~ ., data = train4,
                method = "svmRadial")
m4RMSE <- model4$results$RMSE


train5 <- bus_new[1:584, ]
test5 <- bus_new[585:730, ]
model5 <- train(`Buses: Total Estimated Ridership` ~ ., data = train5,
                method = "svmRadial")
m5RMSE <- model5$results$RMSE

newBusSVM_RMSE <- (sum(m1RMSE)+sum(m2RMSE)+sum(m3RMSE)+sum(m4RMSE)+sum(m5RMSE))/15
```

Sub new SVM

```
train1 <- sub_new[1:233, ]
test1 <- sub_new[234:292, ]
model1 <- train(`Subways: Total Estimated Ridership` ~ ., data = train1,
                method = "svmRadial")
m1RMSE <- model1$results$RMSE

train2 <- sub_new[1:320, ]
test2 <- sub_new[320:401, ]
model2 <- train(`Subways: Total Estimated Ridership` ~ ., data = train2,
                method = "svmRadial")
m2RMSE <- model2$results$RMSE


train3 <- sub_new[1:408, ]
test3 <- sub_new[409:511, ]
model3 <- train(`Subways: Total Estimated Ridership` ~ ., data = train3,
                method = "svmRadial")
m3RMSE <- model3$results$RMSE


train4 <- sub_new[1:496, ]
```

```
test4 <- sub_new[497:620, ]
model4 <- train(`Subways: Total Estimated Ridership` ~ ., data = train4,
                method = "svmRadial")
m4RMSE <- model4$results$RMSE


train5 <- sub_new[1:584, ]
test5 <- sub_new[585:730, ]
model5 <- train(`Subways: Total Estimated Ridership` ~ ., data = train5,
                method = "svmRadial")
m5RMSE <- model5$results$RMSE

newSubSVM_RMSE <-(sum(m1RMSE)+sum(m2RMSE)+sum(m3RMSE)+sum(m4RMSE)+sum(m5RMSE))/15
```

Bus new LM

```
train1 <- bus_new[1:233, ]
test1 <- bus_new[234:292, ]
model1 <- train(`Buses: Total Estimated Ridership` ~ ., data = train1,
                method = "lm")
m1RMSE <- model1$results$RMSE

train2 <- bus_new[1:320, ]
test2 <- bus_new[320:401, ]
model2 <- train(`Buses: Total Estimated Ridership` ~ ., data = train2,
                method = "lm")
m2RMSE <- model2$results$RMSE


train3 <- bus_new[1:408, ]
test3 <- bus_new[409:511, ]
model3 <- train(`Buses: Total Estimated Ridership` ~ ., data = train3,
                method = "lm")
m3RMSE <- model3$results$RMSE


train4 <- bus_new[1:496, ]
test4 <- bus_new[497:620, ]
model4 <- train(`Buses: Total Estimated Ridership` ~ ., data = train4,
                method = "lm")
m4RMSE <- model4$results$RMSE
```

```
train5 <- bus_new[1:584, ]
test5 <- bus_new[585:730, ]
model5 <- train(`Buses: Total Estimated Ridership` ~ ., data = train5,
                method = "lm")
m5RMSE <- model5$results$RMSE

newBusLM_RMSE <- (sum(m1RMSE)+sum(m2RMSE)+sum(m3RMSE)+sum(m4RMSE)+sum(m5RMSE))/15
```

Sub new LM

```
train1 <- sub_new[1:233, ]
test1 <- sub_new[234:292, ]
model1 <- train(`Subways: Total Estimated Ridership` ~ ., data = train1,
                method = "lm")
m1RMSE <- model1$results$RMSE

train2 <- sub_new[1:320, ]
test2 <- sub_new[320:401, ]
model2 <- train(`Subways: Total Estimated Ridership` ~ ., data = train2,
                method = "lm")
m2RMSE <- model2$results$RMSE


train3 <- sub_new[1:408, ]
test3 <- sub_new[409:511, ]
model3 <- train(`Subways: Total Estimated Ridership` ~ ., data = train3,
                method = "lm")
m3RMSE <- model3$results$RMSE


train4 <- sub_new[1:496, ]
test4 <- sub_new[497:620, ]
model4 <- train(`Subways: Total Estimated Ridership` ~ ., data = train4,
                method = "lm")
m4RMSE <- model4$results$RMSE


train5 <- sub_new[1:584, ]
test5 <- sub_new[585:730, ]
model5 <- train(`Subways: Total Estimated Ridership` ~ ., data = train5,
                method = "lm")
```

```
m5RMSE <- model5$results$RMSE

newSubLM_RMSE <- (sum(m1RMSE)+sum(m2RMSE)+sum(m3RMSE)+sum(m4RMSE)+sum(m5RMSE))/15
```

RMSE Table

```
df_RMSE <-
  data.frame(
    Model = c('Bus SVM',
              'Sub SVM',
              'Bus LM',
              'Sub LM'),
    RMSE = c(newBusSVM_RMSE,
             newSubSVM_RMSE,
             newBusLM_RMSE,
             newSubLM_RMSE
             )
  )
df_RMSE
```

```
    Model      RMSE
1 Bus SVM 126549.24
2 Sub SVM 288559.10
3  Bus LM  49165.62
4  Sub LM 122793.57
```

Plotting/predictions

```
train1 <- bus_new[1:233, ]
test1 <- bus_new[234:292, ]
model1 <- train(`Buses: Total Estimated Ridership` ~ ., data = train1,
                method = "lm")
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
m1RMSE <- model1$results$RMSE

train2 <- bus_new[1:320, ]
test2 <- bus_new[320:401, ]
model2 <- train(`Buses: Total Estimated Ridership` ~ ., data = train2,
                method = "lm")
m2RMSE <- model2$results$RMSE


train3 <- bus_new[1:408, ]
test3 <- bus_new[409:511, ]
model3 <- train(`Buses: Total Estimated Ridership` ~ ., data = train3,
                method = "lm")
m3RMSE <- model3$results$RMSE


train4 <- bus_new[1:496, ]
test4 <- bus_new[497:620, ]
model4 <- train(`Buses: Total Estimated Ridership` ~ ., data = train4,
                method = "lm")
m4RMSE <- model4$results$RMSE


train5 <- bus_new[1:584, ]
test5 <- bus_new[585:730, ]
model5 <- train(`Buses: Total Estimated Ridership` ~ ., data = train5,
                method = "lm")
m5RMSE <- model5$results$RMSE

newBusLM_RMSE <- (sum(m1RMSE)+sum(m2RMSE)+sum(m3RMSE)+sum(m4RMSE)+sum(m5RMSE))/15


nrow(test4$Date)
```

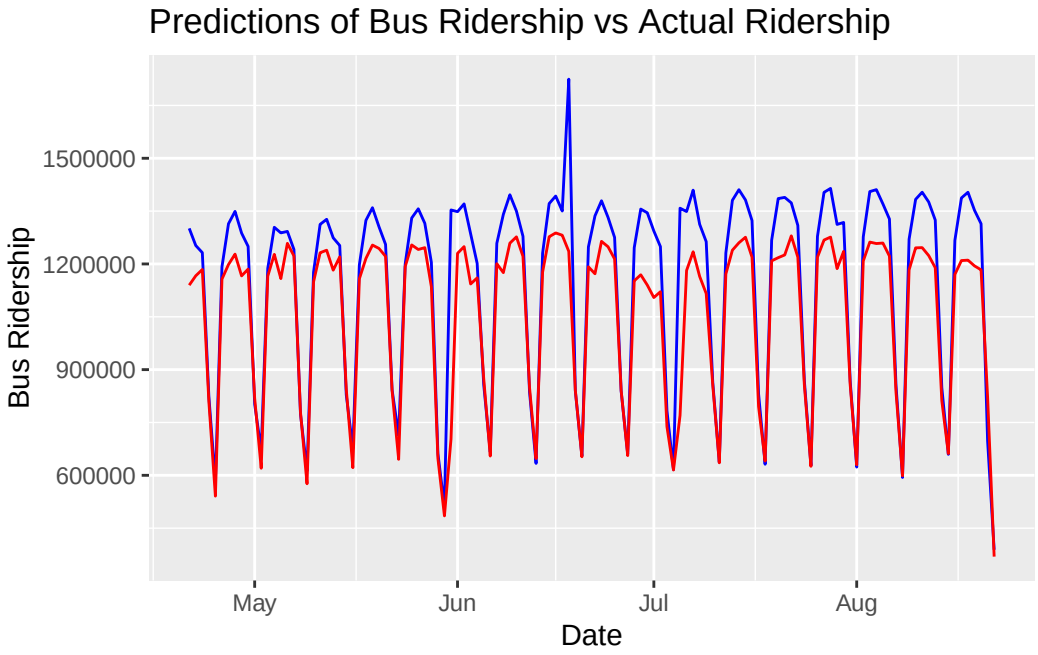
```
Warning: Unknown or uninitialised column: `Date`.
```

```
NULL
```

```
predictions <- predict(model4, test4)
```

```
plotsdf <- bus_df[497:620, ] %>%
  select(Date,`Buses: Total Estimated Ridership`) %>%
  mutate(pred = predictions)
```

```
plotsdf %>%
  ggplot(aes(x = Date)) +
  geom_line(aes(y = pred), color = 'blue') +
  geom_line(aes(y = `Buses: Total Estimated Ridership`), color = 'red') +
  ylab('Bus Ridership') +
  ggtitle('Predictions of Bus Ridership vs Actual Ridership')
```

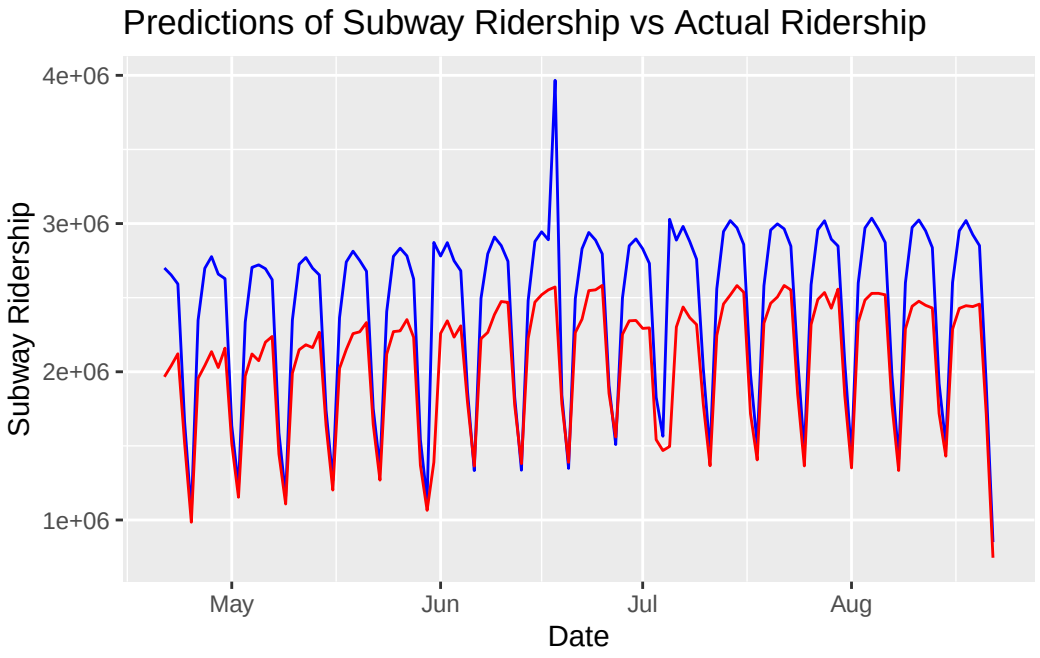



```
train1 <- sub_new[1:233, ]
test1 <- sub_new[234:292, ]
model1 <- train(`Subways: Total Estimated Ridership` ~ ., data = train1,
                method = "lm")
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
m1RMSE <- model1$results$RMSE

train2 <- sub_new[1:320, ]
test2 <- sub_new[320:401, ]
model2 <- train(`Subways: Total Estimated Ridership` ~ ., data = train2,
                method = "lm")
m2RMSE <- model2$results$RMSE


train3 <- sub_new[1:408, ]
test3 <- sub_new[409:511, ]
model3 <- train(`Subways: Total Estimated Ridership` ~ ., data = train3,

                method = "lm")
m3RMSE <- model3$results$RMSE


train4 <- sub_new[1:496, ]
test4 <- sub_new[497:620, ]
model4 <- train(`Subways: Total Estimated Ridership` ~ ., data = train4,
                method = "lm")
m4RMSE <- model4$results$RMSE


train5 <- sub_new[1:584, ]
test5 <- sub_new[585:730, ]
model5 <- train(`Subways: Total Estimated Ridership` ~ ., data = train5,
                method = "lm")
m5RMSE <- model5$results$RMSE
```

```
newSubLM_RMSE <- (sum(m1RMSE)+sum(m2RMSE)+sum(m3RMSE)+sum(m4RMSE)+sum(m5RMSE))/15



predictions <- predict(model4, test4)

plotsdf <- sub_df[497:620, ] %>%
  select(Date,`Subways: Total Estimated Ridership`) %>%
  mutate(pred = predictions)

plotsdf %>%
  ggplot(aes(x = Date)) +
  geom_line(aes(y = pred), color = 'blue' ) +
  geom_line(aes(y = `Subways: Total Estimated Ridership`), color = 'red') +
  ylab('Subway Ridership') +
  ggtitle('Predictions of Subway Ridership vs Actual Ridership')
```




```
k <- 10

data <- sub_df %>% select(-conditions,
                          -Date,
```

```
                          -icon,
                          -temperature)

split_indices <- seq(1, nrow(data), by = floor(nrow(data) / k))

# Perform cross-validation
for (i in 2:length(split_indices)) {
  # Split the data into training and testing sets
  train_indices <- seq(split_indices[i - 1], split_indices[i] - 1)
  test_indices<-split_indices[i]:min(split_indices[i] + floor(nrow(data) / k) - 1,
                                     nrow(data))
  train_data <- data[train_indices, ]
  test_data <- data[test_indices, ]

  # Train the model on the training data
  model <- train(`Subways: Total Estimated Ridership` ~ ., data = train_data,
                 method = "lm")

  # Make predictions on the testing data
  predictions <- predict(model, newdata = test_data)

  # Compute the accuracy metric(s) for this fold
  accuracy <- sqrt(
    mean((test_data$`Subways: Total Estimated Ridership` - predictions) ^ 2)
    )

  # Print the accuracy metric(s) for this fold
  print(accuracy)
}
```

```
[1] 496740.8
[1] 374982.8
[1] 332034.3
[1] 438947.8
[1] 461855.1
[1] 839692
[1] 381064.6
[1] 187870.6
[1] 294260.8
```

Including Holidays

```
holiday_df <- data.frame(
  holiday = c("New Year's Day", "Independence Day", "Thanksgiving", "Christmas"),
  ds = as.Date(c("2021-01-01", "2021-07-04", "2021-11-25", "2021-12-25")),
  lower_window = 0,
  upper_window = 1
)

sub_new <- sub_df %>%
  select (`Subways: Total Estimated Ridership`, `Day of Week`,
           `Subways: % of Comparable Pre-Pandemic Day`,temp, snowdepth, Date)

bus_new <- bus_df %>%
  select(`Buses: Total Estimated Ridership`,`Day of Week`,
         `Buses: % of Comparable Pre-Pandemic Day`, temp, windspeed, snowdepth,
         Date)


holiday_df %>% head
```

```
          holiday         ds lower_window upper_window
1   New Year's Day 2021-01-01            0            1
2 Independence Day 2021-07-04            0            1
3     Thanksgiving 2021-11-25            0            1
4        Christmas 2021-12-25            0            1
```

```
#Merging the dataset
sub_df <- sub_new %>% mutate(isholiday = ifelse(Date %in% holiday_df$ds, 1, 0))
bus_df <- bus_new %>% mutate(isholiday = ifelse(Date %in% holiday_df$ds, 1, 0))
```

Creating the train/test datasets

```
# Split the data into training and testing sets
train_sub_df <- sub_df %>% filter(Date < as.Date("2022-01-01"))
test_sub_df <- sub_df %>% filter(Date >= as.Date("2022-01-01"))

train_bus_df <- bus_df %>% filter(Date < as.Date("2022-01-01"))
test_bus_df <- bus_df %>% filter(Date >= as.Date("2022-01-01"))
```

Preparing the model to use the `prophet` function

```r
# Create a function to prepare the data for the Prophet model
prepare_data_for_prophet <- function(df, y_column) {
  df <- df %>%
    select(Date, y = !!y_column) %>%
    rename(ds = Date)
  return(df)
}

# Prepare the data for the Prophet model
train_sub_prophet <- prepare_data_for_prophet(train_sub_df,
                                              'Subways: Total Estimated Ridership')
test_sub_prophet <- prepare_data_for_prophet(test_sub_df,
                                              'Subways: Total Estimated Ridership')

train_bus_prophet <- prepare_data_for_prophet(train_bus_df,
                                                'Buses: Total Estimated Ridership')
test_bus_prophet <- prepare_data_for_prophet(test_bus_df,
                                              'Buses: Total Estimated Ridership')
```

Using the prophet model to forecast ridership

```r
sub_prophet_model <- prophet(df = train_sub_prophet, holidays = holiday_df,
                             yearly.seasonality = TRUE)
```

```
Disabling daily seasonality. Run prophet with daily.seasonality=TRUE to override this.
```

```r
sub_forecast <- predict(sub_prophet_model, test_sub_prophet)

bus_prophet_model <- prophet(df = train_bus_prophet, holidays = holiday_df, yearly.seasona
```

```
Disabling daily seasonality. Run prophet with daily.seasonality=TRUE to override this.
```

```r
bus_forecast <- predict(bus_prophet_model, test_bus_prophet)
```

Combining actual and forecasted values for comparison

```r
sub_comparison <- test_sub_prophet %>%
  left_join(sub_forecast %>% select(ds, yhat), by = c("ds")) %>%
  rename(actual = y, forecast = yhat)
```

```r
bus_comparison <- test_bus_prophet %>%
  left_join(bus_forecast %>% select(ds, yhat), by = c("ds")) %>%
  rename(actual = y, forecast = yhat)
```

Calculating MAE, MSE and RMSE

```r
calculate_metrics <- function(actual, forecast){
  mae <- mean(abs(actual - forecast))
  mse <- mean((actual - forecast)^2)
  rmse <- sqrt(mse)

  return(list(MAE = mae, MSE = mse, RMSE = rmse))
}

sub_metrics <- calculate_metrics(sub_comparison$actual, sub_comparison$forecast)
sub_metrics
```

```
$MAE
[1] 846832.1

$MSE
[1] 866503246341

$RMSE
[1] 930861.6
```

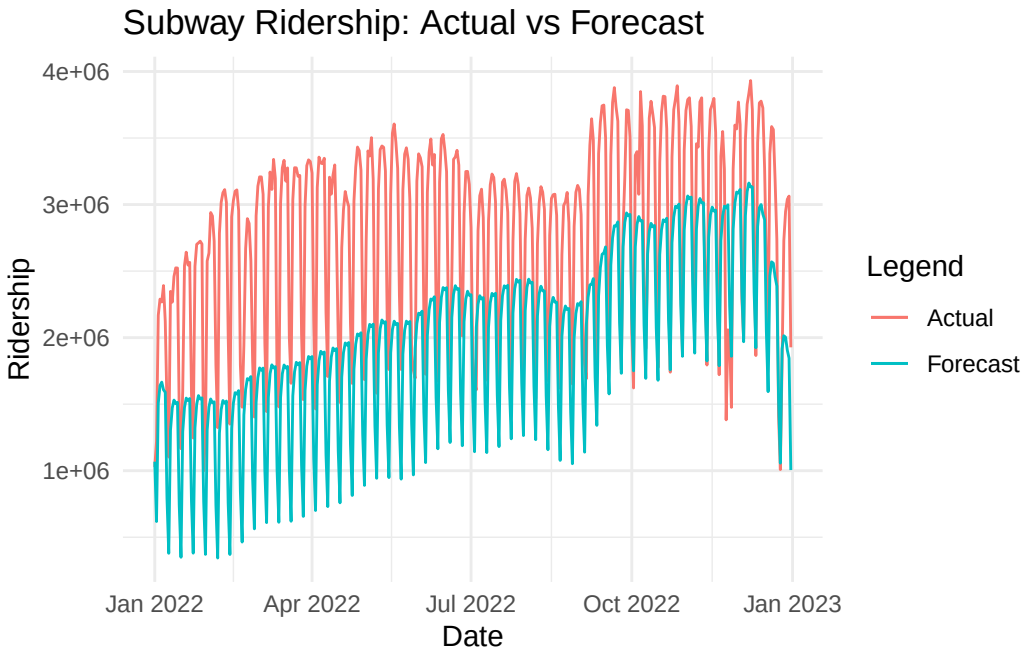
```r
bus_metrics <- calculate_metrics(bus_comparison$actual, bus_comparison$forecast)
bus_metrics
```

```
$MAE
[1] 426422.9

$MSE
[1] 203328063945

$RMSE
[1] 450919.1
```

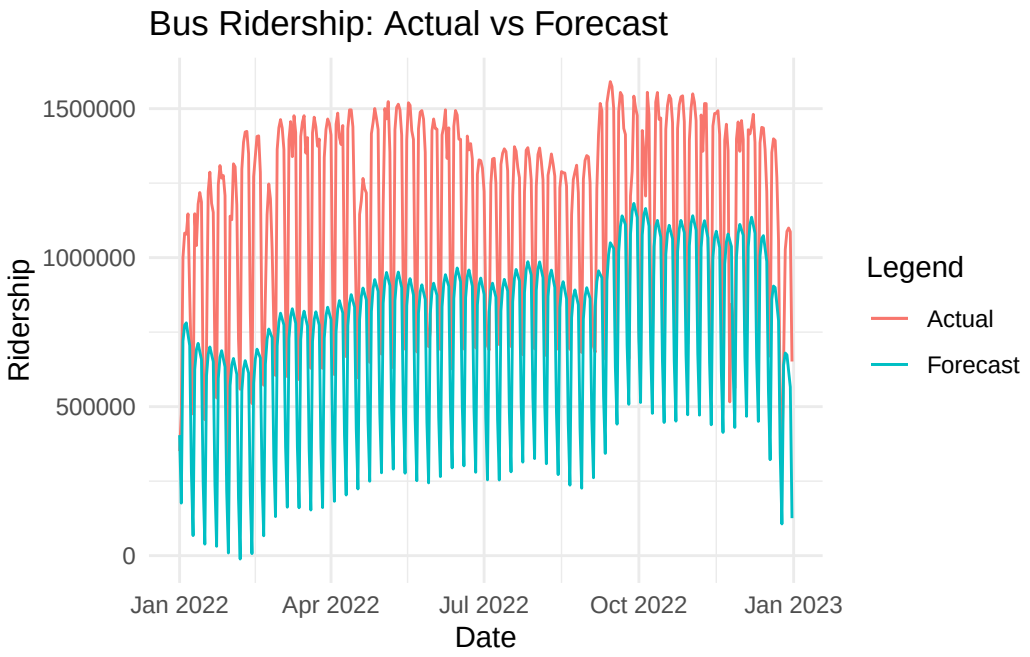
Inferences from ML

Plotting actual vs forecasted values

```
ggplot(sub_comparison, aes(x = ds)) +
  geom_line(aes(y = actual, color = "Actual")) +
  geom_line(aes(y = forecast, color = "Forecast")) +
  labs(title = "Subway Ridership: Actual vs Forecast",
       x = "Date",
       y = "Ridership",
       color = "Legend") +
  theme_minimal()
```



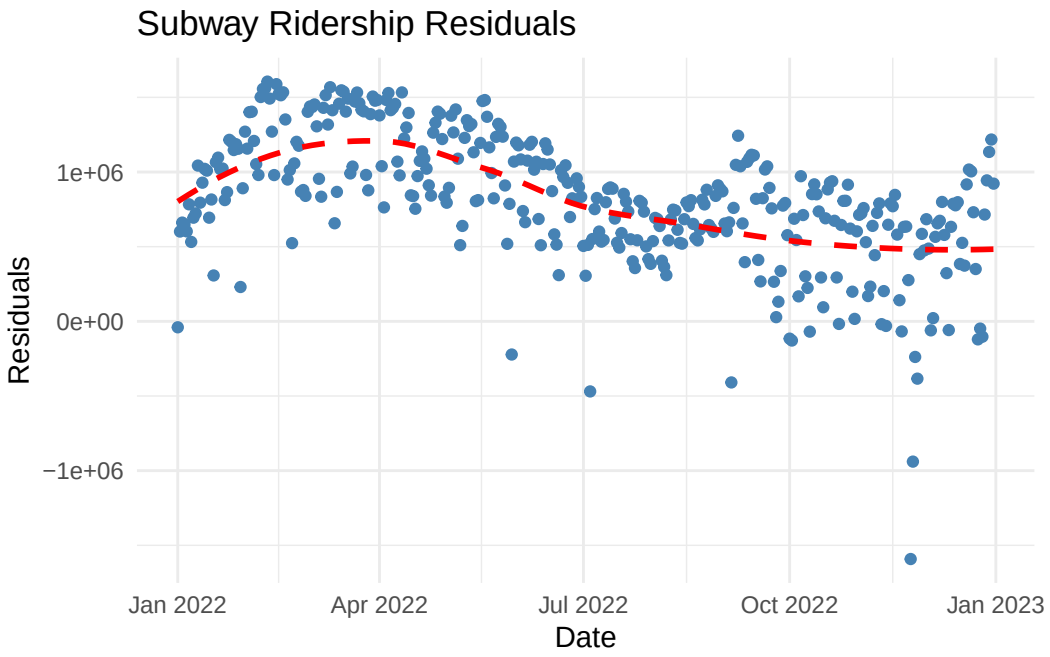

```
ggplot(bus_comparison, aes(x = ds)) +
  geom_line(aes(y = actual, color = "Actual")) +
  geom_line(aes(y = forecast, color = "Forecast")) +
  labs(title = "Bus Ridership: Actual vs Forecast",
       x = "Date",
       y = "Ridership",
       color = "Legend") +
  theme_minimal()
```



Visualizing Residuals

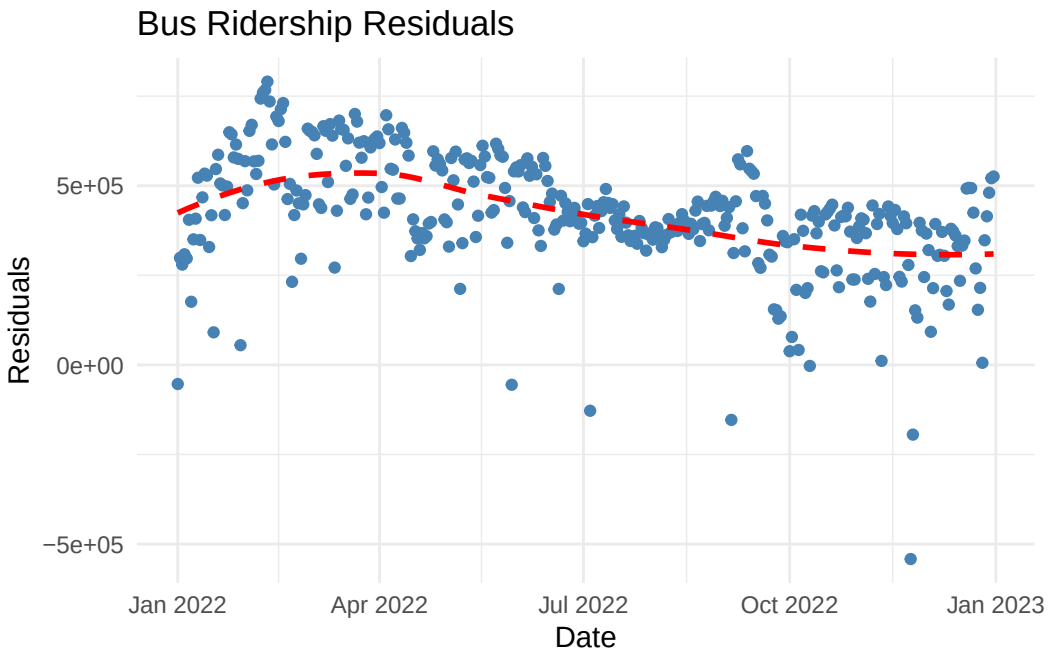
```
sub_comparison <- sub_comparison %>% mutate(residuals = actual - forecast)
bus_comparison <- bus_comparison %>% mutate(residuals = actual - forecast)

ggplot(sub_comparison, aes(x = ds, y = residuals)) +
  geom_point(color = "steelblue") +
  geom_smooth(se = FALSE, color = "red", linetype = "dashed") +
  labs(title = "Subway Ridership Residuals",
       x = "Date",
       y = "Residuals") +
  theme_minimal()
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
ggplot(bus_comparison, aes(x = ds, y = residuals)) +
  geom_point(color = "steelblue") +
  geom_smooth(se = FALSE, color = "red", linetype = "dashed") +
  labs(title = "Bus Ridership Residuals",
       x = "Date",
       y = "Residuals") +
  theme_minimal()
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```


Bus Ridership Residuals
Residuals
5e+05
0e+00
−5e+05
Jan 2022
Apr 2022
Jul 2022
Oct 2022
Jan 2023
Date