MTA Ridership Final Project

J.Kimbrough, Kennedy Rodriguez

2025-05-01

```
url <- "https://data.ny.gov/api/views/vxuj-8kew/rows.csv?accessType=DOWNLOAD"
mta_data_raw <- read_csv(url)
```

```
## Rows: 1776 Columns: 15
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (1): Date
## dbl (14): Subways: Total Estimated Ridership, Subways: % of Comparable Pre-P...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
view(mta_data_raw)
glimpse(mta_data_raw)
```

```
## Rows: 1,776
## Columns: 15
## $ Date                                                  <chr> "01/01/2021"…
## $ `Subways: Total Estimated Ridership`                  <dbl> 613692, 1027…
## $ `Subways: % of Comparable Pre-Pandemic Day`           <dbl> 0.29, 0.38, …
## $ `Buses: Total Estimated Ridership`                    <dbl> 378288, 3508…
## $ `Buses: % of Comparable Pre-Pandemic Day`             <dbl> 0.41, 0.29, …
## $ `LIRR: Total Estimated Ridership`                     <dbl> 28977, 33980…
## $ `LIRR: % of Comparable Pre-Pandemic Day`              <dbl> 0.35, 0.35, …
## $ `Metro-North: Total Estimated Ridership`              <dbl> 14988, 30341…
## $ `Metro-North: % of Comparable Pre-Pandemic Day`       <dbl> 0.17, 0.23, …
## $ `Access-A-Ride: Total Scheduled Trips`                <dbl> 5960, 4904, …
## $ `Access-A-Ride: % of Comparable Pre-Pandemic Day`     <dbl> 0.44, 0.34, …
## $ `Bridges and Tunnels: Total Traffic`                  <dbl> 445950, 4985…
## $ `Bridges and Tunnels: % of Comparable Pre-Pandemic Day` <dbl> 0.65, 0.65, …
## $ `Staten Island Railway: Total Estimated Ridership`    <dbl> 805, 1262, 1…
## $ `Staten Island Railway: % of Comparable Pre-Pandemic Day` <dbl> 0.29, 0.31, …
```

—- Cleaning Data —–

```
anyNA(mta_data_raw)
```

```
## [1] FALSE
```

```r
any(sapply(mta_data_raw, function(x) any(x=="")))
```

```
## [1] FALSE
```

```r
any(sapply(mta_data_raw, function(x) any(x == -999)))
```

```
## [1] FALSE
```

```r
df <- mta_data_raw %>%
  mutate(Date = mdy(Date)) %>%
  mutate(across(where(is.character), ~ na_if(., "missing"))) %>%
  mutate(across(where(is.numeric), ~ na_if(., -999))) %>%
  mutate(across(where(is.character), ~ na_if(., ""))) %>%
  na.omit()

head(df)
```

```
## # A tibble: 6 × 15
##   Date       `Subways: Total Estimated Ridership` Subways: % of Comparable Pre…¹
##   <date>                                    <dbl>                          <dbl>
## 1 2021-01-01                               613692                           0.29
## 2 2022-01-01                              1027918                           0.38
## 3 2023-01-01                              1675507                           0.8
## 4 2024-01-01                              1648734                           0.79
## 5 2025-01-01                              1779352                           0.85
## 6 2021-01-02                               988418                           0.37
## # ℹ abbreviated name: ¹`Subways: % of Comparable Pre-Pandemic Day`
## # ℹ 12 more variables: `Buses: Total Estimated Ridership` <dbl>,
## #   `Buses: % of Comparable Pre-Pandemic Day` <dbl>,
## #   `LIRR: Total Estimated Ridership` <dbl>,
## #   `LIRR: % of Comparable Pre-Pandemic Day` <dbl>,
## #   `Metro-North: Total Estimated Ridership` <dbl>,
## #   `Metro-North: % of Comparable Pre-Pandemic Day` <dbl>, …
```

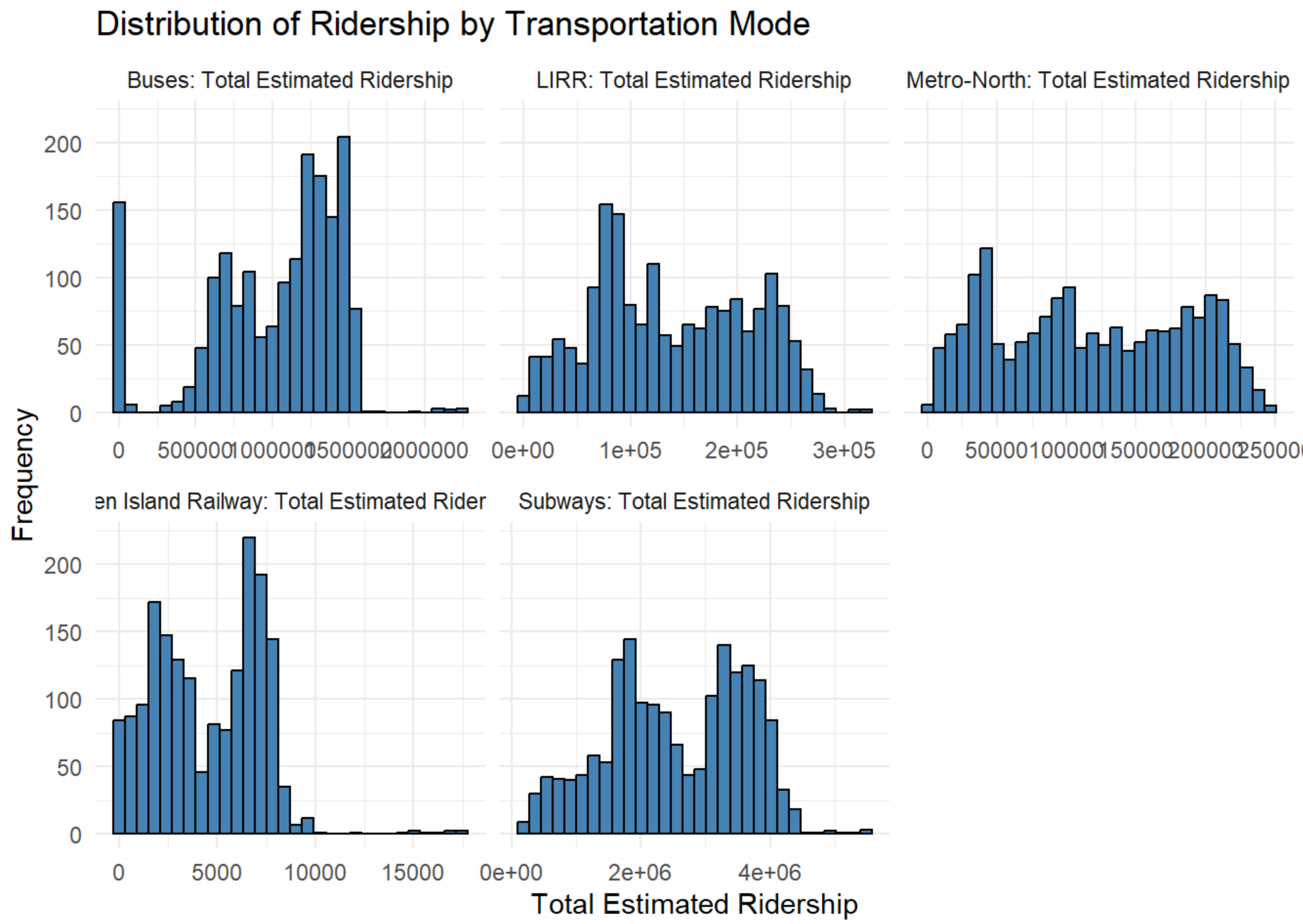
— Exploratory Data Analysis —

```r
colnames(df)
```

```
##  [1] "Date"
##  [2] "Subways: Total Estimated Ridership"
##  [3] "Subways: % of Comparable Pre-Pandemic Day"
##  [4] "Buses: Total Estimated Ridership"
##  [5] "Buses: % of Comparable Pre-Pandemic Day"
##  [6] "LIRR: Total Estimated Ridership"
##  [7] "LIRR: % of Comparable Pre-Pandemic Day"
##  [8] "Metro-North: Total Estimated Ridership"
##  [9] "Metro-North: % of Comparable Pre-Pandemic Day"
## [10] "Access-A-Ride: Total Scheduled Trips"
## [11] "Access-A-Ride: % of Comparable Pre-Pandemic Day"
## [12] "Bridges and Tunnels: Total Traffic"
## [13] "Bridges and Tunnels: % of Comparable Pre-Pandemic Day"
## [14] "Staten Island Railway: Total Estimated Ridership"
## [15] "Staten Island Railway: % of Comparable Pre-Pandemic Day"
```

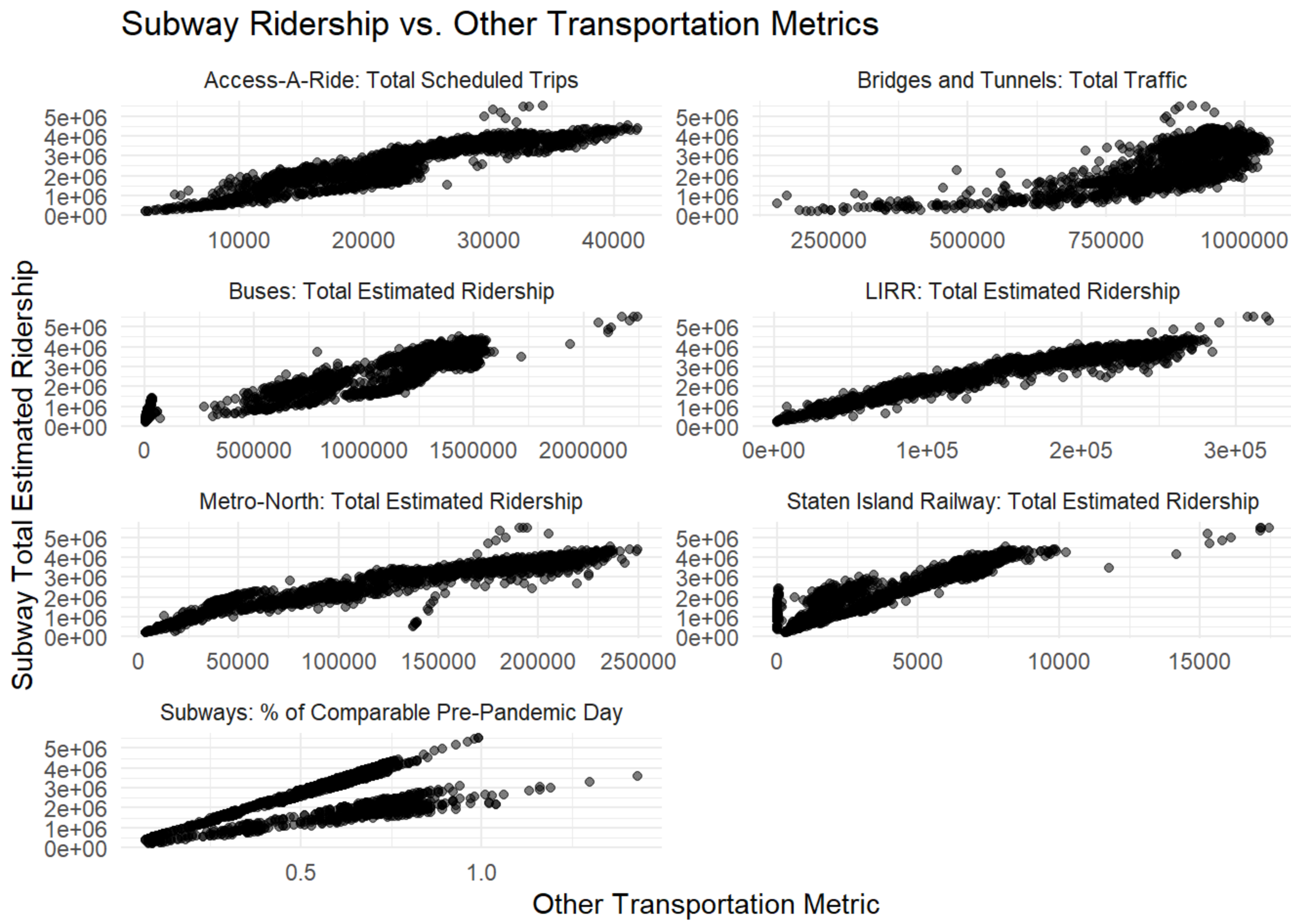
Histograms

```
df_long <- df %>%
   dplyr::select(`Subways: Total Estimated Ridership`,
         `Buses: Total Estimated Ridership`,
         `LIRR: Total Estimated Ridership`,
         `Metro-North: Total Estimated Ridership`,
         `Staten Island Railway: Total Estimated Ridership`) %>%
  pivot_longer(cols = everything(),
               names_to = "Transportation_Mode",
               values_to = "Ridership")

ggplot(df_long, aes(x = Ridership)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "black") +
  facet_wrap(~ Transportation_Mode, scales = "free_x") +
  labs(title = "Distribution of Ridership by Transportation Mode",
       x = "Total Estimated Ridership",
       y = "Frequency") +
  theme_minimal()
```



Plotted Relationships

```
df_long <- df %>%
   dplyr::select(
    `Subways: % of Comparable Pre-Pandemic Day`,
    `Buses: Total Estimated Ridership`,
    `LIRR: Total Estimated Ridership`,
    `Metro-North: Total Estimated Ridership`,
    `Access-A-Ride: Total Scheduled Trips`,
    `Bridges and Tunnels: Total Traffic`,
    `Staten Island Railway: Total Estimated Ridership`,
    `Subways: Total Estimated Ridership`
  ) %>%
  pivot_longer(
    cols = -`Subways: Total Estimated Ridership`,
    names_to = "Other_Metric",
    values_to = "Other_Value"
  )

ggplot(df_long, aes(x = Other_Value, y = `Subways: Total Estimated Ridership`)) +
  geom_point(alpha = 0.5) +
  facet_wrap(~ Other_Metric, scales = "free", ncol = 2) + # Removed the extra + before the comme
nt
  labs(
    title = "Subway Ridership vs. Other Transportation Metrics",
    x = "Other Transportation Metric",
    y = "Subway Total Estimated Ridership"
  ) +
  theme_minimal()
```



Correlations

```
numeric_cols <- names(df)[sapply(df, is.numeric) & names(df) != "Subways: Total Estimated Riders
hip"]

correlation_results <- df %>%
  dplyr::select(-Date) %>% # Ensure Date is removed
  pivot_longer(
    cols = all_of(numeric_cols),
    names_to = "Other_Metric",
    values_to = "Other_Value"
  ) %>%
  group_by(Other_Metric) %>%
  summarize(
    correlation = cor(`Subways: Total Estimated Ridership`, Other_Value, use = "pairwise.complet
e.obs")
  )

print(correlation_results)
```

```
## # A tibble: 13 × 2
##    Other_Metric                                          correlation
##    <chr>                                                       <dbl>
##  1 Access-A-Ride: % of Comparable Pre-Pandemic Day             0.706
##  2 Access-A-Ride: Total Scheduled Trips                        0.909
##  3 Bridges and Tunnels: % of Comparable Pre-Pandemic Day       0.659
##  4 Bridges and Tunnels: Total Traffic                          0.738
##  5 Buses: % of Comparable Pre-Pandemic Day                     0.640
##  6 Buses: Total Estimated Ridership                            0.884
##  7 LIRR: % of Comparable Pre-Pandemic Day                      0.409
##  8 LIRR: Total Estimated Ridership                             0.961
##  9 Metro-North: % of Comparable Pre-Pandemic Day               0.601
## 10 Metro-North: Total Estimated Ridership                      0.943
## 11 Staten Island Railway: % of Comparable Pre-Pandemic Day     0.469
## 12 Staten Island Railway: Total Estimated Ridership            0.923
## 13 Subways: % of Comparable Pre-Pandemic Day                   0.646
```

```
"Buses: Total Estimated Ridership
The correlation coefficient between bus ridership and subway ridership is 0.88, indicating a str
ong positive correlation. This suggests that when more people ride the bus, subway ridership ten
ds to be higher as well. The two systems may serve as complements or be influenced by similar de
mand patterns.

Buses: % of Comparable Pre-Pandemic Day
The correlation here is 0.64, showing a moderate positive relationship. As bus ridership returns
to its pre-pandemic baseline, subway ridership also tends to increase, though not as strongly as
with total bus numbers.

LIRR: Total Estimated Ridership
With a very high correlation of 0.96, this represents a very strong positive relationship. It in
dicates that Long Island Rail Road ridership is highly synchronized with subway usage, likely du
e to intermodal transfers and similar commuter patterns.

LIRR: % of Comparable Pre-Pandemic Day
The correlation is 0.41, which is weakly positive. This suggests that while total LIRR ridership
is strongly associated with subway use, the pace of recovery relative to pre-pandemic levels is
not as tightly linked.

Metro-North: Total Estimated Ridership
The coefficient is 0.94, indicating another very strong positive correlation. Like the LIRR, Met
ro-North ridership appears to rise and fall in tandem with subway ridership.

Metro-North: % of Comparable Pre-Pandemic Day
This yields a moderate positive correlation of 0.60. Similar to LIRR trends, total ridership ali
gns more closely with subway usage than the relative recovery percentage does.

Access-A-Ride: Total Scheduled Trips
A correlation of 0.91 suggests a strong positive relationship. This implies that when more parat
ransit trips are scheduled, subway ridership also increases, possibly reflecting broader pattern
s of urban mobility and accessibility.

Access-A-Ride: % of Comparable Pre-Pandemic Day
The coefficient is 0.71, still a moderately strong correlation. As Access-A-Ride approaches pre-
pandemic service levels, subway usage tends to rise, though again with less intensity than total
trip counts.

Bridges and Tunnels: Total Traffic
The correlation of 0.74 shows a moderate positive relationship between road traffic and subway r
idership. This may reflect general increases in overall mobility within the region.

Bridges and Tunnels: % of Comparable Pre-Pandemic Day
At 0.66, this remains a moderate positive correlation, implying that as bridge and tunnel traffi
c returns to normal, subway ridership also tends to rise.

Staten Island Railway: Total Estimated Ridership
This yields a strong positive correlation of 0.92. Subway and Staten Island Railway usage appear
closely linked, likely due to the ferry and rail connections to the main subway system.

Staten Island Railway: % of Comparable Pre-Pandemic Day
```

```
The correlation is 0.47, which is weak but positive. Again, while absolute ridership levels alig
n well, the recovery pace doesn't strongly predict subway usage.

Summary
Overall, total ridership figures for other transportation modes tend to have stronger correlatio
ns with subway usage than their percentages of pre-pandemic levels. This suggests that actual vo
lume of riders is a more reliable indicator of subway demand than recovery benchmarks alone. Com
muter rail lines (LIRR and Metro-North) and Access-A-Ride show particularly high associations wi
th subway usage, pointing to potential interdependencies in how New Yorkers use different MTA se
rvices."
```

```
## [1] "Buses: Total Estimated Ridership\nThe correlation coefficient between bus ridership and
subway ridership is 0.88, indicating a strong positive correlation. This suggests that when more
people ride the bus, subway ridership tends to be higher as well. The two systems may serve as c
omplements or be influenced by similar demand patterns.\n\nBuses: % of Comparable Pre-Pandemic D
ay\nThe correlation here is 0.64, showing a moderate positive relationship. As bus ridership ret
urns to its pre-pandemic baseline, subway ridership also tends to increase, though not as strong
ly as with total bus numbers.\n\nLIRR: Total Estimated Ridership\nWith a very high correlation o
f 0.96, this represents a very strong positive relationship. It indicates that Long Island Rail
Road ridership is highly synchronized with subway usage, likely due to intermodal transfers and
similar commuter patterns.\n\nLIRR: % of Comparable Pre-Pandemic Day\nThe correlation is 0.41, w
hich is weakly positive. This suggests that while total LIRR ridership is strongly associated wi
th subway use, the pace of recovery relative to pre-pandemic levels is not as tightly linked.\n
\nMetro-North: Total Estimated Ridership\nThe coefficient is 0.94, indicating another very stron
g positive correlation. Like the LIRR, Metro-North ridership appears to rise and fall in tandem
with subway ridership.\n\nMetro-North: % of Comparable Pre-Pandemic Day\nThis yields a moderate
positive correlation of 0.60. Similar to LIRR trends, total ridership aligns more closely with s
ubway usage than the relative recovery percentage does.\n\nAccess-A-Ride: Total Scheduled Trips
\nA correlation of 0.91 suggests a strong positive relationship. This implies that when more par
atransit trips are scheduled, subway ridership also increases, possibly reflecting broader patte
rns of urban mobility and accessibility.\n\nAccess-A-Ride: % of Comparable Pre-Pandemic Day\nThe
coefficient is 0.71, still a moderately strong correlation. As Access-A-Ride approaches pre-pand
emic service levels, subway usage tends to rise, though again with less intensity than total tri
p counts.\n\nBridges and Tunnels: Total Traffic\nThe correlation of 0.74 shows a moderate positi
ve relationship between road traffic and subway ridership. This may reflect general increases in
overall mobility within the region.\n\nBridges and Tunnels: % of Comparable Pre-Pandemic Day\nAt
0.66, this remains a moderate positive correlation, implying that as bridge and tunnel traffic r
eturns to normal, subway ridership also tends to rise.\n\nStaten Island Railway: Total Estimated
Ridership\nThis yields a strong positive correlation of 0.92. Subway and Staten Island Railway u
sage appear closely linked, likely due to the ferry and rail connections to the main subway syst
em.\n\nStaten Island Railway: % of Comparable Pre-Pandemic Day\nThe correlation is 0.47, which i
s weak but positive. Again, while absolute ridership levels align well, the recovery pace does
n't strongly predict subway usage.\n\nSummary\nOverall, total ridership figures for other transp
ortation modes tend to have stronger correlations with subway usage than their percentages of pr
e-pandemic levels. This suggests that actual volume of riders is a more reliable indicator of su
bway demand than recovery benchmarks alone. Commuter rail lines (LIRR and Metro-North) and Acces
s-A-Ride show particularly high associations with subway usage, pointing to potential interdepen
dencies in how New Yorkers use different MTA services."
```

Data Summary Table

```
df_with_year <- df %>%
  mutate(Year = year(Date))

ridership_summary <- df_with_year %>%
  group_by(Year) %>%
  summarize(
    Total_Subway_Ridership = sum(`Subways: Total Estimated Ridership`, na.rm = TRUE),
    Total_Bus_Ridership = sum(`Buses: Total Estimated Ridership`, na.rm = TRUE),
    Total_LIRR_Ridership = sum(`LIRR: Total Estimated Ridership`, na.rm = TRUE),
    Total_MNR_Ridership = sum(`Metro-North: Total Estimated Ridership`, na.rm = TRUE),
    Total_SIR_Ridership = sum(`Staten Island Railway: Total Estimated Ridership`, na.rm = TRUE),
    Total_AAR_Ridership =sum(`Access-A-Ride: Total Scheduled Trips`, na.rm = TRUE),
  ) %>%
  filter(Year != 2025)


print(ridership_summary)
```

```
## # A tibble: 5 × 7
##    Year Total_Subway_Ridership Total_Bus_Ridership Total_LIRR_Ridership
##   <dbl>                  <dbl>               <dbl>                <dbl>
## 1  2020              370096769           147387699             17816724
## 2  2021              759810246           381637866             35269817
## 3  2022             1012505879           423946824             51998770
## 4  2023             1150217108           425539799             64882573
## 5  2024             1194201712           408872440             74848660
## # ℹ 3 more variables: Total_MNR_Ridership <dbl>, Total_SIR_Ridership <dbl>,
## #   Total_AAR_Ridership <dbl>
```

Regression Question

Updated - What type of regression model best predicts total daily subway ridership?

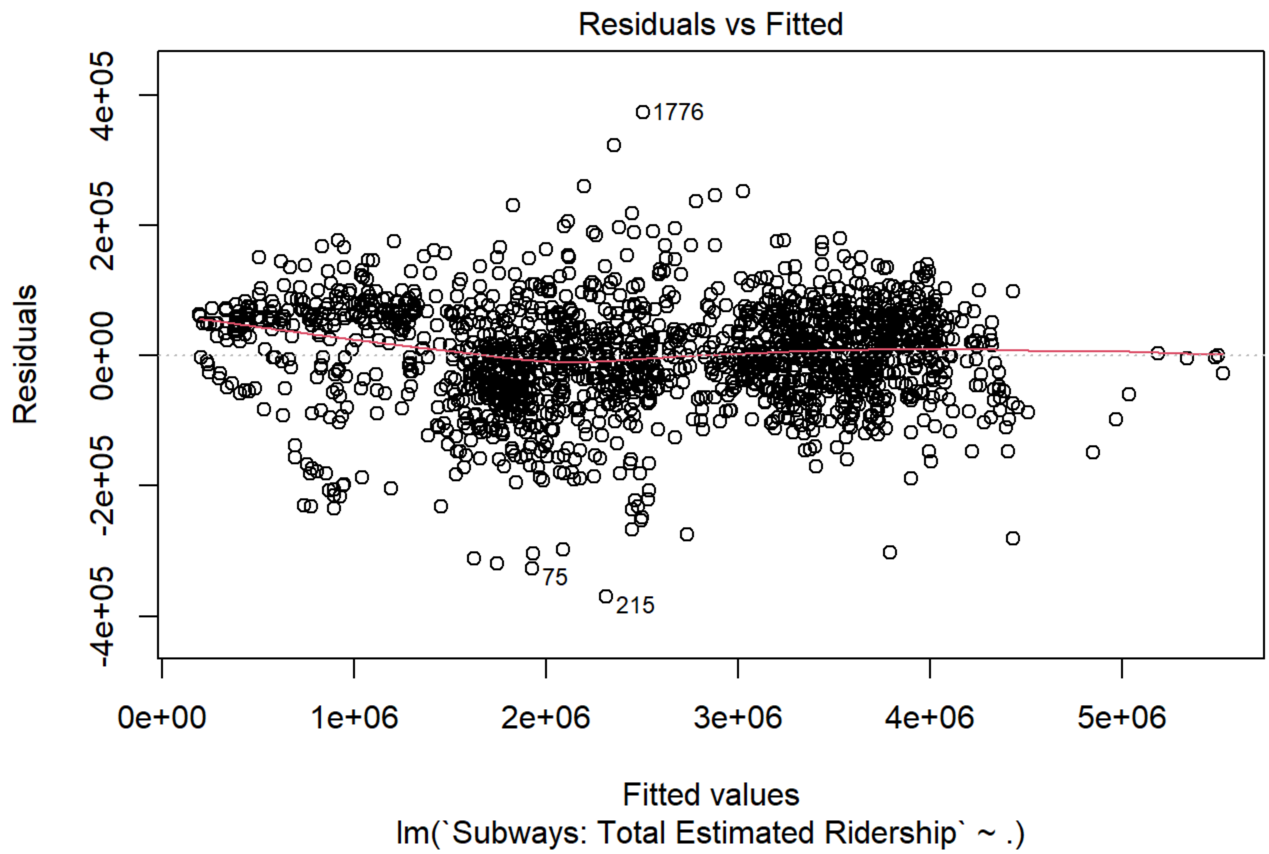
— Multiple Linear Regression —

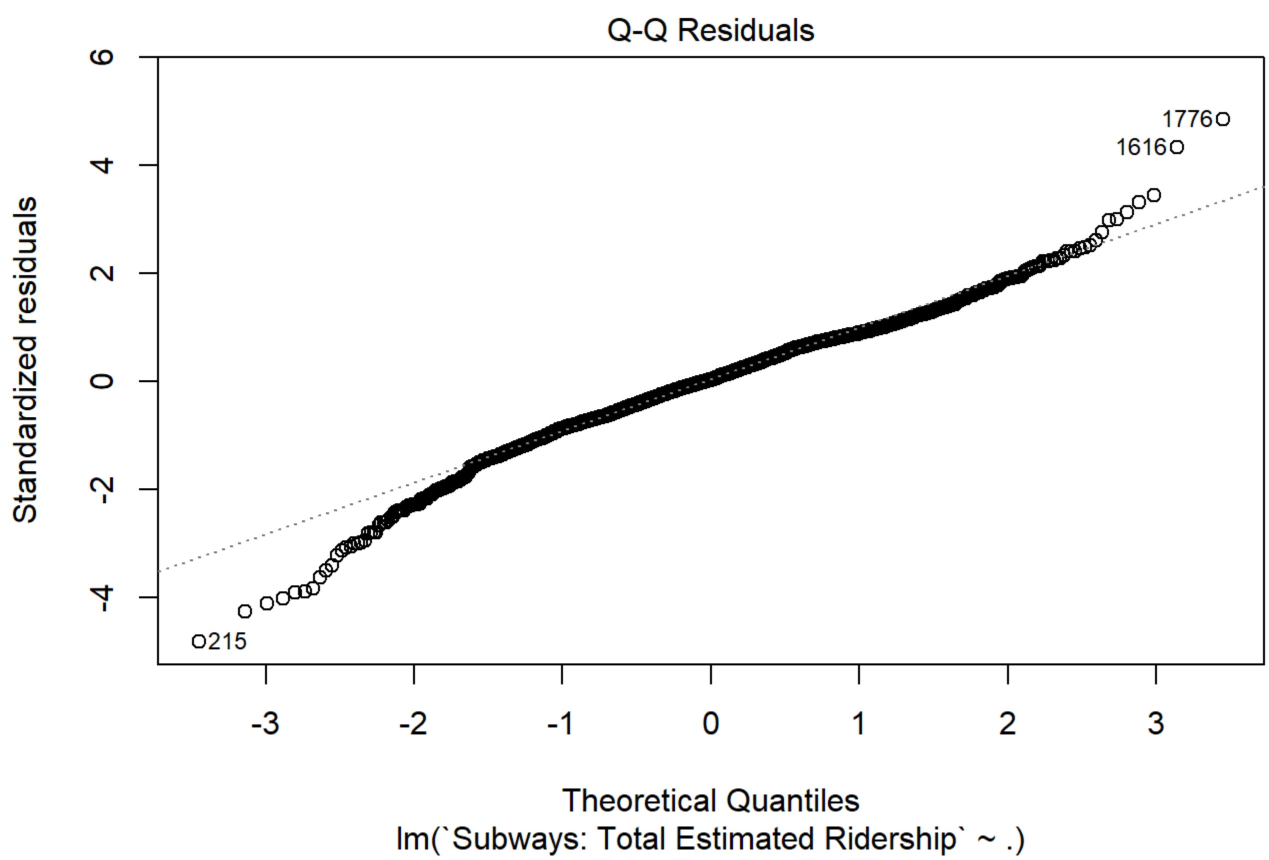
```
# Subways: Total Estimated Ridership

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

summary(full_model)
```

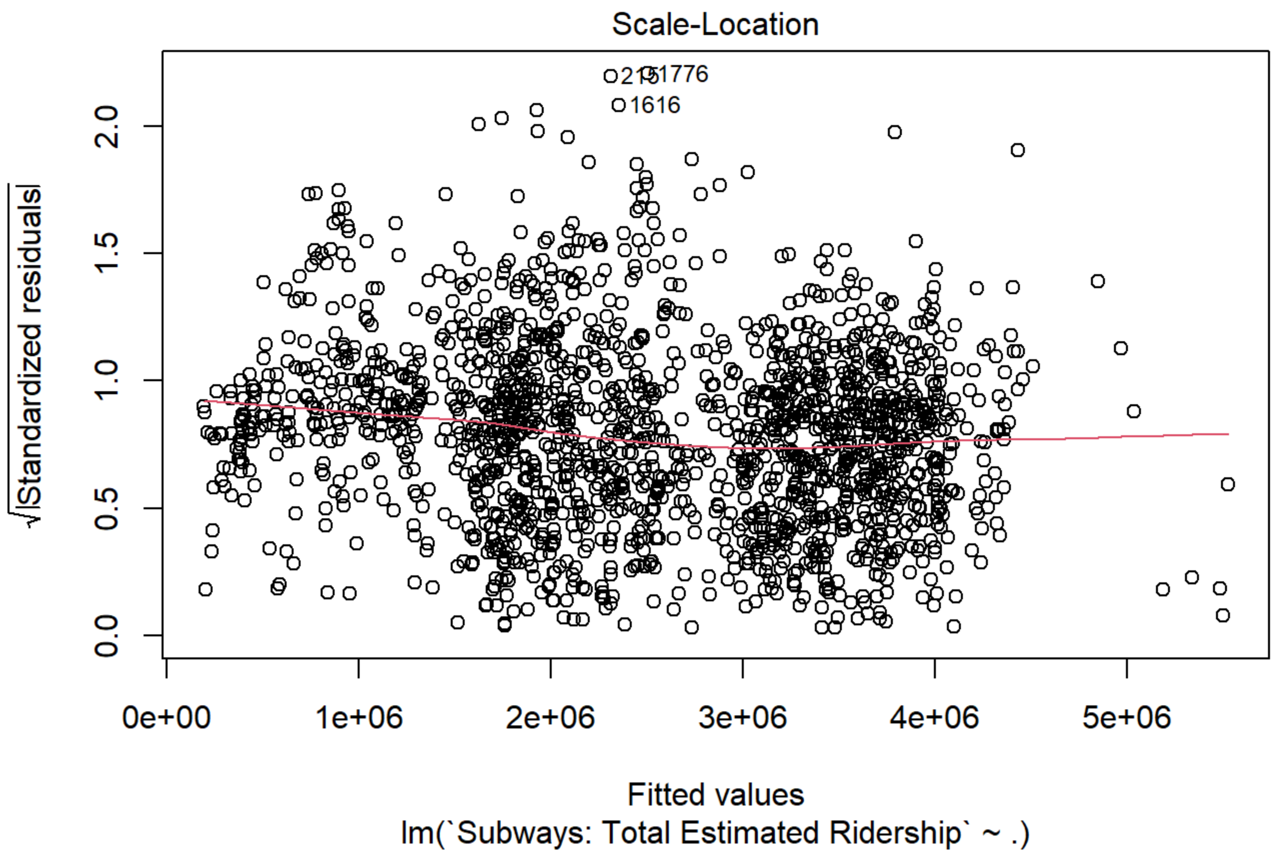
```
##
## Call:
## lm(formula = `Subways: Total Estimated Ridership` ~ ., data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -370472  -47635    2301   54610  373079
##
## Coefficients:
##                                                             Estimate Std. Error
## (Intercept)                                               -6.109e+05  2.492e+05
## Date                                                       3.061e+01  1.363e+01
## `Subways: % of Comparable Pre-Pandemic Day`                3.309e+06  5.087e+04
## `Buses: Total Estimated Ridership`                         7.759e-01  2.803e-02
## `Buses: % of Comparable Pre-Pandemic Day`                 -1.224e+06  5.404e+04
## `LIRR: Total Estimated Ridership`                         -1.688e+00  2.628e-01
## `LIRR: % of Comparable Pre-Pandemic Day`                   4.378e+05  4.342e+04
## `Metro-North: Total Estimated Ridership`                   7.565e+00  2.454e-01
## `Metro-North: % of Comparable Pre-Pandemic Day`           -1.441e+06  5.306e+04
## `Access-A-Ride: Total Scheduled Trips`                     3.334e+01  1.510e+00
## `Access-A-Ride: % of Comparable Pre-Pandemic Day`         -7.891e+05  3.667e+04
## `Bridges and Tunnels: Total Traffic`                       2.647e-01  5.123e-02
## `Bridges and Tunnels: % of Comparable Pre-Pandemic Day`   -1.043e+05  5.211e+04
## `Staten Island Railway: Total Estimated Ridership`         7.646e+01  4.686e+00
## `Staten Island Railway: % of Comparable Pre-Pandemic Day` -2.511e+05  2.626e+04
##                                                           t value Pr(>|t|)
## (Intercept)                                                -2.451   0.0143 *
## Date                                                        2.246   0.0249 *
## `Subways: % of Comparable Pre-Pandemic Day`                65.055  < 2e-16 ***
## `Buses: Total Estimated Ridership`                         27.676  < 2e-16 ***
## `Buses: % of Comparable Pre-Pandemic Day`                 -22.644  < 2e-16 ***
## `LIRR: Total Estimated Ridership`                          -6.424 1.70e-10 ***
## `LIRR: % of Comparable Pre-Pandemic Day`                   10.084  < 2e-16 ***
## `Metro-North: Total Estimated Ridership`                   30.823  < 2e-16 ***
## `Metro-North: % of Comparable Pre-Pandemic Day`           -27.162  < 2e-16 ***
## `Access-A-Ride: Total Scheduled Trips`                     22.082  < 2e-16 ***
## `Access-A-Ride: % of Comparable Pre-Pandemic Day`         -21.517  < 2e-16 ***
## `Bridges and Tunnels: Total Traffic`                        5.167 2.65e-07 ***
## `Bridges and Tunnels: % of Comparable Pre-Pandemic Day`    -2.001   0.0456 *
## `Staten Island Railway: Total Estimated Ridership`         16.316  < 2e-16 ***
## `Staten Island Railway: % of Comparable Pre-Pandemic Day`  -9.563  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 79240 on 1761 degrees of freedom
## Multiple R-squared:  0.9945, Adjusted R-squared:  0.9945
## F-statistic: 2.289e+04 on 14 and 1761 DF,  p-value: < 2.2e-16
```

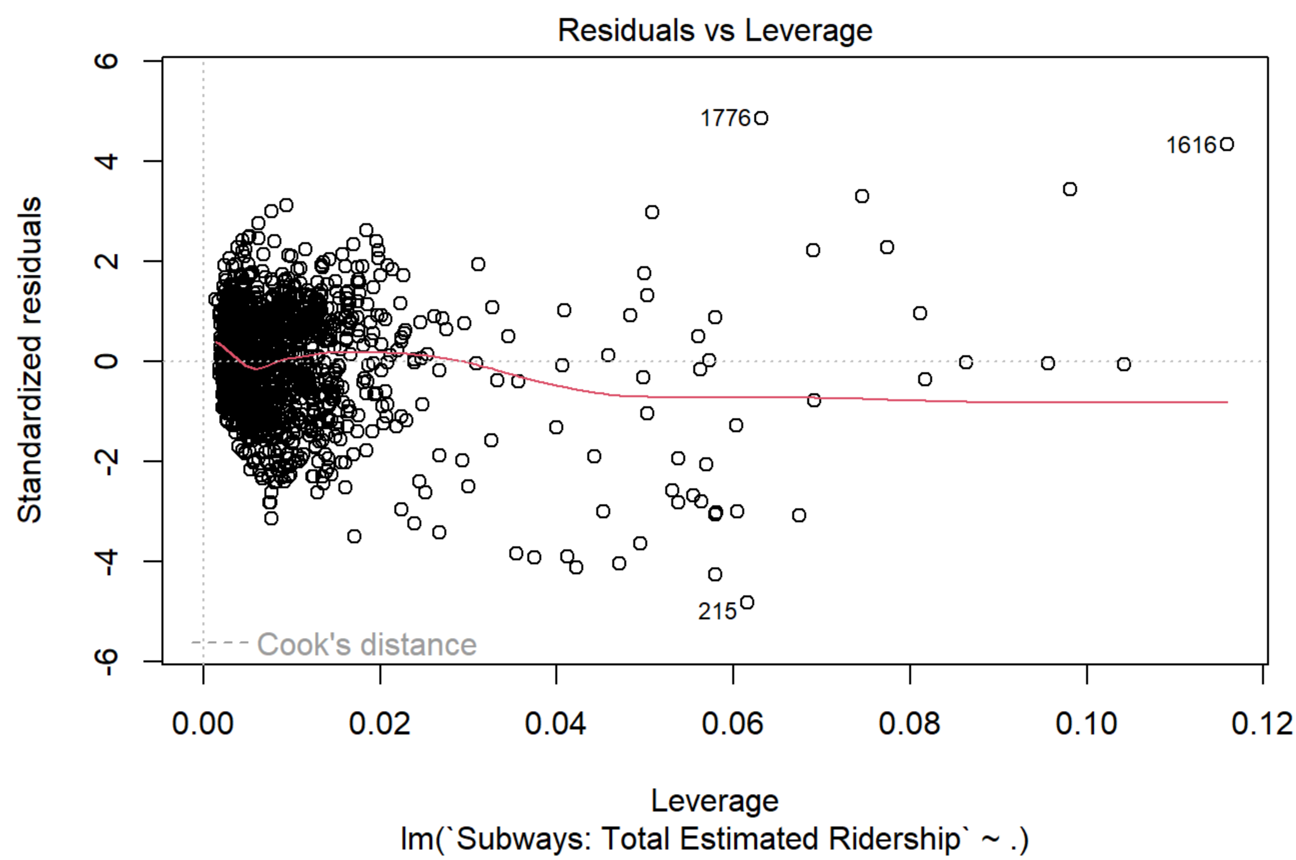
The model explains about 99.45% of the variability in subway ridership, which is extremely high. With an F-stat of 22,890 and a p-value < 2.2e-16, the overall model is highly statistically significant.

```
plot(full_model)
```


Residuals vs Fitted
Residuals
4e+05
2e+05
0e+00
-2e+05
-4e+05
1776
75
215
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
Fitted values
lm(`Subways: Total Estimated Ridership` ~ .)


Q-Q Residuals
Standardized residuals
6
4
2
0
-2
-4
1776
1616
215
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(`Subways: Total Estimated Ridership` ~ .)


Scale-Location
√|Standardized residuals|
215
1776
1616
0.0
0.5
1.0
1.5
2.0
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
Fitted values
lm(`Subways: Total Estimated Ridership` ~ .)

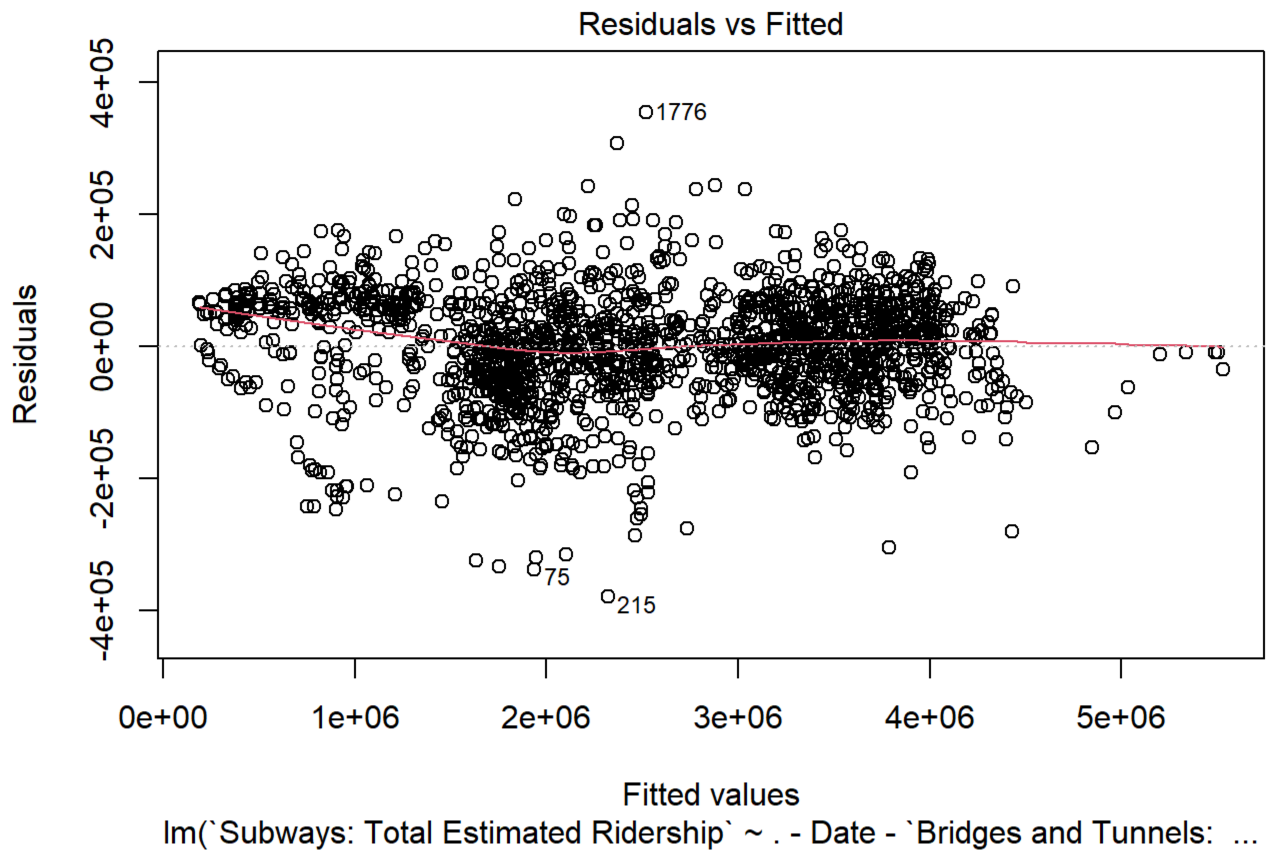

Residuals vs Leverage
Standardized residuals
1776
1616
215
Cook's distance
-6
-4
-2
0
2
4
6
0.00
0.02
0.04
0.06
0.08
0.10
0.12
Leverage
lm(`Subways: Total Estimated Ridership` ~ .)

```r
#NO Date, Bridges and Tunnels: % of Comparable Pre-Pandemic Day
reduced_model <- lm(`Subways: Total Estimated Ridership` ~ . -Date -`Bridges and Tunnels: % of C
omparable Pre-Pandemic Day`, data = df)
summary(reduced_model)
```

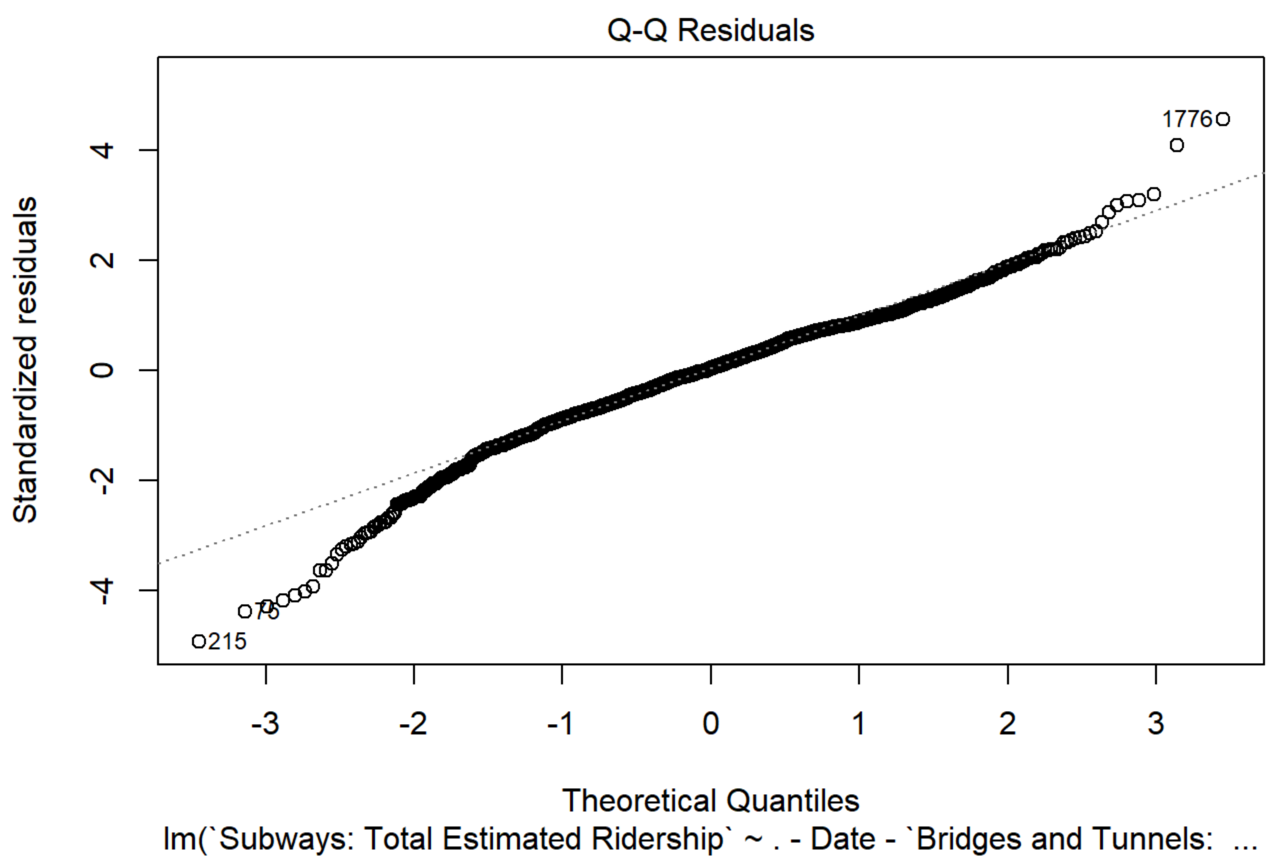
```
##
## Call:
## lm(formula = `Subways: Total Estimated Ridership` ~ . - Date -
##     `Bridges and Tunnels: % of Comparable Pre-Pandemic Day`,
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -379122  -46918    3771   54886  354444
##
## Coefficients:
##                                                           Estimate Std. Error
## (Intercept)                                             -6.092e+04  1.483e+04
## `Subways: % of Comparable Pre-Pandemic Day`              3.323e+06  4.814e+04
## `Buses: Total Estimated Ridership`                       8.052e-01  2.586e-02
## `Buses: % of Comparable Pre-Pandemic Day`               -1.292e+06  4.777e+04
## `LIRR: Total Estimated Ridership`                       -1.542e+00  2.526e-01
## `LIRR: % of Comparable Pre-Pandemic Day`                 4.359e+05  4.271e+04
## `Metro-North: Total Estimated Ridership`                 7.665e+00  2.414e-01
## `Metro-North: % of Comparable Pre-Pandemic Day`         -1.435e+06  5.236e+04
## `Access-A-Ride: Total Scheduled Trips`                   3.329e+01  1.507e+00
## `Access-A-Ride: % of Comparable Pre-Pandemic Day`       -7.673e+05  3.536e+04
## `Bridges and Tunnels: Total Traffic`                     1.603e-01  2.503e-02
## `Staten Island Railway: Total Estimated Ridership`       7.027e+01  3.947e+00
## `Staten Island Railway: % of Comparable Pre-Pandemic Day` -2.360e+05  2.565e+04
##                                                         t value Pr(>|t|)
## (Intercept)                                              -4.108 4.17e-05 ***
## `Subways: % of Comparable Pre-Pandemic Day`              69.024  < 2e-16 ***
## `Buses: Total Estimated Ridership`                       31.140  < 2e-16 ***
## `Buses: % of Comparable Pre-Pandemic Day`               -27.044  < 2e-16 ***
## `LIRR: Total Estimated Ridership`                        -6.107 1.25e-09 ***
## `LIRR: % of Comparable Pre-Pandemic Day`                 10.206  < 2e-16 ***
## `Metro-North: Total Estimated Ridership`                 31.754  < 2e-16 ***
## `Metro-North: % of Comparable Pre-Pandemic Day`         -27.405  < 2e-16 ***
## `Access-A-Ride: Total Scheduled Trips`                   22.093  < 2e-16 ***
## `Access-A-Ride: % of Comparable Pre-Pandemic Day`       -21.698  < 2e-16 ***
## `Bridges and Tunnels: Total Traffic`                      6.402 1.96e-10 ***
## `Staten Island Railway: Total Estimated Ridership`       17.802  < 2e-16 ***
## `Staten Island Railway: % of Comparable Pre-Pandemic Day`  -9.200  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 79350 on 1763 degrees of freedom
## Multiple R-squared:  0.9945, Adjusted R-squared:  0.9945
## F-statistic: 2.663e+04 on 12 and 1763 DF,  p-value: < 2.2e-16
```

This model still explains about 99.45% of the variability in subway ridership, which is extremely high. The F-stat is still pretty high at 26,630 and a low p-value. The overall model is still statistically significant.

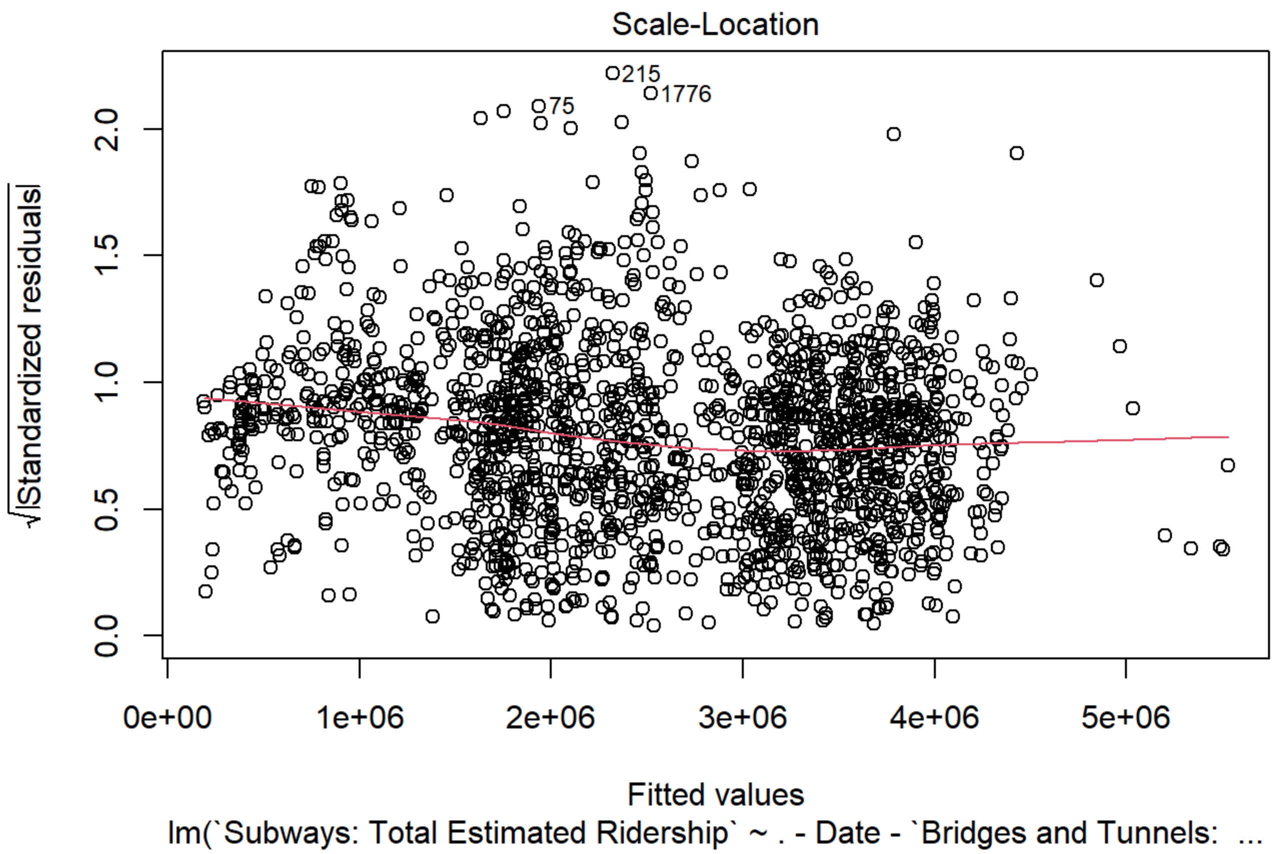
```
plot(reduced_model)
```

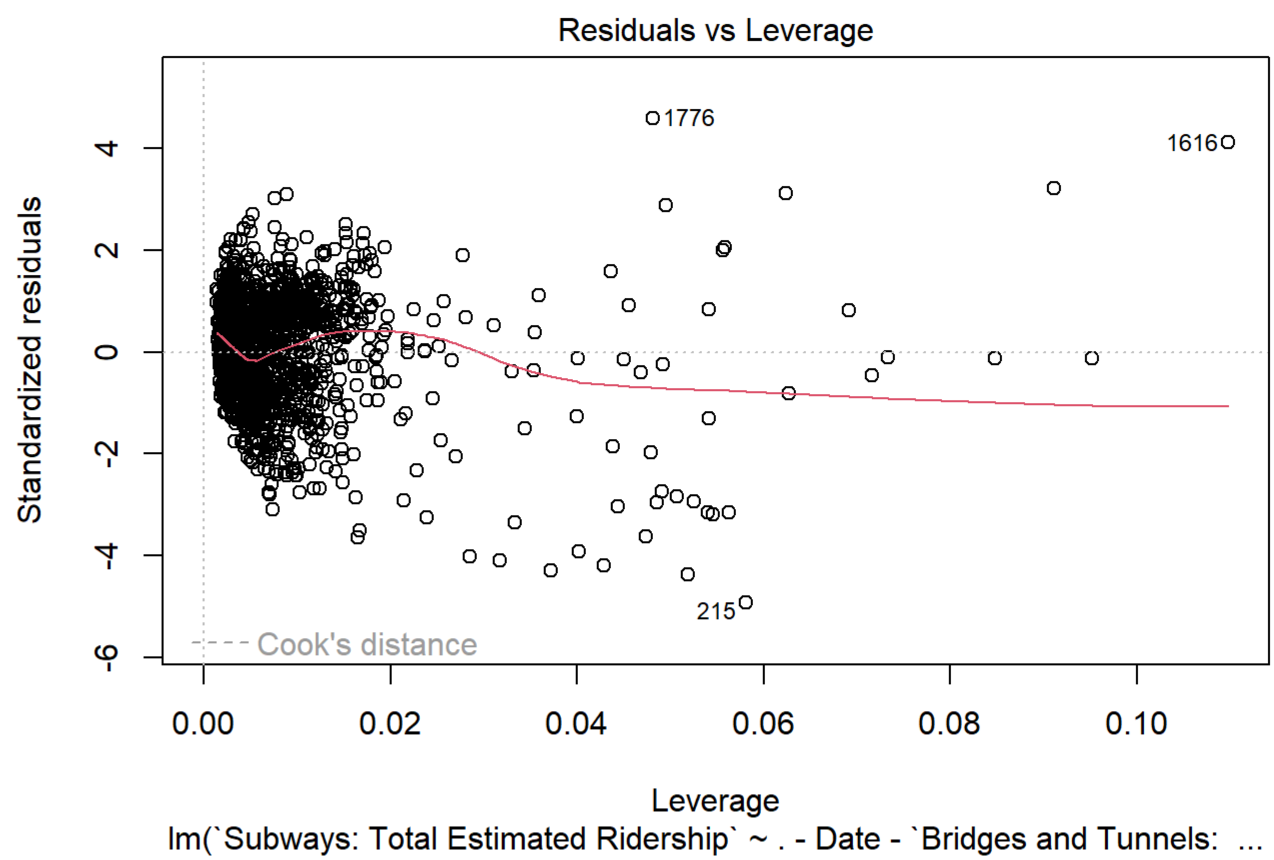


Residuals vs Fitted
Residuals
Fitted values
lm(`Subways: Total Estimated Ridership` ~ . - Date - `Bridges and Tunnels: ...
1776
75
215
4e+05
2e+05
0e+00
-2e+05
-4e+05
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06



Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(`Subways: Total Estimated Ridership` ~ . - Date - `Bridges and Tunnels: ...
1776
75
215
4
2
0
-2
-4
-3
-2
-1
0
1
2
3


Scale-Location
√|Standardized residuals|
Fitted values
lm(`Subways: Total Estimated Ridership` ~ . - Date - `Bridges and Tunnels: ...
215
1776
75


Residuals vs Leverage
Standardized residuals
Leverage
lm(`Subways: Total Estimated Ridership` ~ . - Date - `Bridges and Tunnels: ...
Cook's distance
1776
1616
215

```r
reduced_model_step <- step(reduced_model)
```

```
## Start:  AIC=40085.44
## `Subways: Total Estimated Ridership` ~ (Date + `Subways: % of Comparable Pre-Pandemic Day` +
##     `Buses: Total Estimated Ridership` + `Buses: % of Comparable Pre-Pandemic Day` +
##     `LIRR: Total Estimated Ridership` + `LIRR: % of Comparable Pre-Pandemic Day` +
##     `Metro-North: Total Estimated Ridership` + `Metro-North: % of Comparable Pre-Pandemic Day
` +
##     `Access-A-Ride: Total Scheduled Trips` + `Access-A-Ride: % of Comparable Pre-Pandemic Day
` +
##     `Bridges and Tunnels: Total Traffic` + `Bridges and Tunnels: % of Comparable Pre-Pandemic
Day` +
##     `Staten Island Railway: Total Estimated Ridership` + `Staten Island Railway: % of Compara
ble Pre-Pandemic Day`) -
##     Date - `Bridges and Tunnels: % of Comparable Pre-Pandemic Day`
##
##                                                               Df  Sum of Sq
## <none>
## - `LIRR: Total Estimated Ridership`                            1 2.3481e+11
## - `Bridges and Tunnels: Total Traffic`                         1 2.5809e+11
## - `Staten Island Railway: % of Comparable Pre-Pandemic Day`  1 5.3301e+11
## - `LIRR: % of Comparable Pre-Pandemic Day`                     1 6.5589e+11
## - `Staten Island Railway: Total Estimated Ridership`           1 1.9957e+12
## - `Access-A-Ride: % of Comparable Pre-Pandemic Day`            1 2.9646e+12
## - `Access-A-Ride: Total Scheduled Trips`                       1 3.0736e+12
## - `Buses: % of Comparable Pre-Pandemic Day`                    1 4.6055e+12
## - `Metro-North: % of Comparable Pre-Pandemic Day`              1 4.7293e+12
## - `Buses: Total Estimated Ridership`                           1 6.1063e+12
## - `Metro-North: Total Estimated Ridership`                     1 6.3492e+12
## - `Subways: % of Comparable Pre-Pandemic Day`                  1 3.0001e+13
##                                                                      RSS   AIC
## <none>                                                        1.1102e+13 40085
## - `LIRR: Total Estimated Ridership`                            1.1336e+13 40121
## - `Bridges and Tunnels: Total Traffic`                         1.1360e+13 40124
## - `Staten Island Railway: % of Comparable Pre-Pandemic Day` 1.1635e+13 40167
## - `LIRR: % of Comparable Pre-Pandemic Day`                     1.1757e+13 40185
## - `Staten Island Railway: Total Estimated Ridership`           1.3097e+13 40377
## - `Access-A-Ride: % of Comparable Pre-Pandemic Day`            1.4066e+13 40504
## - `Access-A-Ride: Total Scheduled Trips`                       1.4175e+13 40517
## - `Buses: % of Comparable Pre-Pandemic Day`                    1.5707e+13 40700
## - `Metro-North: % of Comparable Pre-Pandemic Day`              1.5831e+13 40714
## - `Buses: Total Estimated Ridership`                           1.7208e+13 40862
## - `Metro-North: Total Estimated Ridership`                     1.7451e+13 40887
## - `Subways: % of Comparable Pre-Pandemic Day`                  4.1103e+13 42408
```

```r
reduced_model_step
```

```
##
## Call:
## lm(formula = `Subways: Total Estimated Ridership` ~ (Date + `Subways: % of Comparable Pre-Pan
demic Day` +
##     `Buses: Total Estimated Ridership` + `Buses: % of Comparable Pre-Pandemic Day` +
##     `LIRR: Total Estimated Ridership` + `LIRR: % of Comparable Pre-Pandemic Day` +
##     `Metro-North: Total Estimated Ridership` + `Metro-North: % of Comparable Pre-Pandemic Day
` +
##     `Access-A-Ride: Total Scheduled Trips` + `Access-A-Ride: % of Comparable Pre-Pandemic Day
` +
##     `Bridges and Tunnels: Total Traffic` + `Bridges and Tunnels: % of Comparable Pre-Pandemic
Day` +
##     `Staten Island Railway: Total Estimated Ridership` + `Staten Island Railway: % of Compara
ble Pre-Pandemic Day`) -
##     Date - `Bridges and Tunnels: % of Comparable Pre-Pandemic Day`,
##     data = df)
##
## Coefficients:
##                                            (Intercept)
##                                             -6.092e+04
##              `Subways: % of Comparable Pre-Pandemic Day`
##                                              3.323e+06
##                       `Buses: Total Estimated Ridership`
##                                              8.052e-01
##                `Buses: % of Comparable Pre-Pandemic Day`
##                                             -1.292e+06
##                        `LIRR: Total Estimated Ridership`
##                                             -1.542e+00
##                 `LIRR: % of Comparable Pre-Pandemic Day`
##                                              4.359e+05
##                 `Metro-North: Total Estimated Ridership`
##                                              7.665e+00
##          `Metro-North: % of Comparable Pre-Pandemic Day`
##                                             -1.435e+06
##                   `Access-A-Ride: Total Scheduled Trips`
##                                              3.329e+01
##        `Access-A-Ride: % of Comparable Pre-Pandemic Day`
##                                             -7.673e+05
##                     `Bridges and Tunnels: Total Traffic`
##                                              1.602e-01
##       `Staten Island Railway: Total Estimated Ridership`
##                                              7.027e+01
## `Staten Island Railway: % of Comparable Pre-Pandemic Day`
##                                             -2.360e+05
```

Best to not drop any variables - dropping them only increases AIC.

```
anova(full_model, reduced_model_step)
```

```
## Analysis of Variance Table
##
## Model 1: `Subways: Total Estimated Ridership` ~ Date + `Subways: % of Comparable Pre-Pandemic
Day` +
##     `Buses: Total Estimated Ridership` + `Buses: % of Comparable Pre-Pandemic Day` +
##     `LIRR: Total Estimated Ridership` + `LIRR: % of Comparable Pre-Pandemic Day` +
##     `Metro-North: Total Estimated Ridership` + `Metro-North: % of Comparable Pre-Pandemic Day
` +
##     `Access-A-Ride: Total Scheduled Trips` + `Access-A-Ride: % of Comparable Pre-Pandemic Day
` +
##     `Bridges and Tunnels: Total Traffic` + `Bridges and Tunnels: % of Comparable Pre-Pandemic
Day` +
##     `Staten Island Railway: Total Estimated Ridership` + `Staten Island Railway: % of Compara
ble Pre-Pandemic Day`
## Model 2: `Subways: Total Estimated Ridership` ~ (Date + `Subways: % of Comparable Pre-Pandemi
c Day` +
##     `Buses: Total Estimated Ridership` + `Buses: % of Comparable Pre-Pandemic Day` +
##     `LIRR: Total Estimated Ridership` + `LIRR: % of Comparable Pre-Pandemic Day` +
##     `Metro-North: Total Estimated Ridership` + `Metro-North: % of Comparable Pre-Pandemic Day
` +
##     `Access-A-Ride: Total Scheduled Trips` + `Access-A-Ride: % of Comparable Pre-Pandemic Day
` +
##     `Bridges and Tunnels: Total Traffic` + `Bridges and Tunnels: % of Comparable Pre-Pandemic
Day` +
##     `Staten Island Railway: Total Estimated Ridership` + `Staten Island Railway: % of Compara
ble Pre-Pandemic Day`) -
##     Date - `Bridges and Tunnels: % of Comparable Pre-Pandemic Day`
##   Res.Df        RSS Df   Sum of Sq     F  Pr(>F)
## 1   1761 1.1056e+13
## 2   1763 1.1102e+13 -2 -4.5279e+10 3.606 0.02736 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Reduced model actually worsened statistically. So, keeping full model performs better.

```
vif_full <- vif(full_model)
vif_full
```

```
##                                                         Date
##                                                    13.817066
##              `Subways: % of Comparable Pre-Pandemic Day`
##                                                    29.573673
##                        `Buses: Total Estimated Ridership`
##                                                    42.427523
##                `Buses: % of Comparable Pre-Pandemic Day`
##                                                    30.397454
##                         `LIRR: Total Estimated Ridership`
##                                                   101.893187
##                 `LIRR: % of Comparable Pre-Pandemic Day`
##                                                    50.592303
##                 `Metro-North: Total Estimated Ridership`
##                                                    76.978996
##          `Metro-North: % of Comparable Pre-Pandemic Day`
##                                                    58.925456
##                    `Access-A-Ride: Total Scheduled Trips`
##                                                    43.671497
##        `Access-A-Ride: % of Comparable Pre-Pandemic Day`
##                                                    25.254651
##                     `Bridges and Tunnels: Total Traffic`
##                                                    14.796732
##  `Bridges and Tunnels: % of Comparable Pre-Pandemic Day`
##                                                    16.123313
##       `Staten Island Railway: Total Estimated Ridership`
##                                                    45.261540
## `Staten Island Railway: % of Comparable Pre-Pandemic Day`
##                                                     7.962559
```

Some multicollinearity was detected, especially with `LIRR Total Estimated Ridership` having the highest vif.

```
predicted_full <- predict(full_model, df)

rmse_full <- sqrt(mean((df$`Subways: Total Estimated Ridership` - predicted_full)^2))
rmse_full
```

```
## [1] 78901.18
```

```
mse_full <- (mean((df$`Subways: Total Estimated Ridership` - predicted_full)^2))
mse_full
```

```
## [1] 6225396224
```

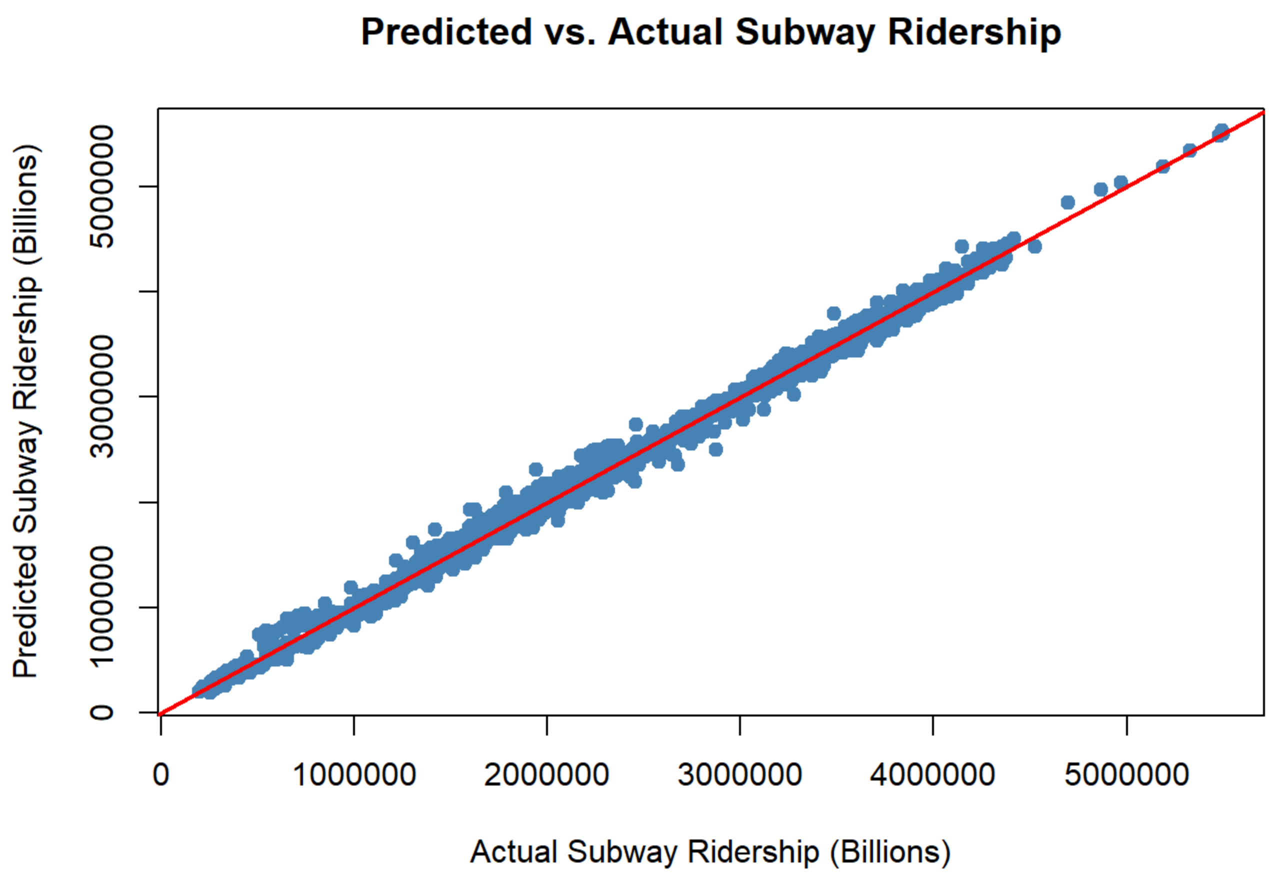
The RMSE shows that there is an average daily prediction error of approximately 78,900 riders.

```
set.seed(1234)
train_control <- trainControl(method = "cv", number = 10)

model_cv <- train(
  `Subways: Total Estimated Ridership` ~ .,
  data = df,
  method = "lm",
  trControl = train_control
)

print(model_cv)
```

```
## Linear Regression
##
## 1776 samples
##   14 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1598, 1598, 1599, 1598, 1597, 1598, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   79974.23  0.9943963  61104.66
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Similar to predicted RMSE for the full model, the cross-validated RMSE shows that there is an average daily prediction error of approximately 80,000 riders. The R^2 is very high, and suggests that the model explains over 99% of the variability in subway ridership.

```
options(scipen = 999)
actual <- df$`Subways: Total Estimated Ridership`
predicted <- predicted_full

plot(actual, predicted,
     xlab = "Actual Subway Ridership (Billions)",
     ylab = "Predicted Subway Ridership (Billions)",
     main = "Predicted vs. Actual Subway Ridership",
     pch = 19, col = "steelblue")
abline(a = 0, b = 1, col = "red", lwd = 2)
```



Classification Question

Updated - Can predict “low,” “average,” or “high” ridership by using classification statistical learning methods?

```
### combining ridership
df$total_ridership <- (df$`Subways: Total Estimated Ridership`+
                         df$`Buses: Total Estimated Ridership`+
                         df$`LIRR: Total Estimated Ridership`+
                         df$`Metro-North: Total Estimated Ridership`+
                         df$`Staten Island Railway: Total Estimated Ridership`)
```

```
quantiles <- quantile(df$total_ridership, probs = c(0.33, 0.66))

df$total_ridership_level <- cut(
  df$total_ridership,
  breaks = c(-Inf, quantiles[1], quantiles[2], Inf),
  labels = c("Low", "Average", "High")
)
```

```
predictors <- c(
  "Subways: % of Comparable Pre-Pandemic Day",
  "Buses: % of Comparable Pre-Pandemic Day",
  "LIRR: % of Comparable Pre-Pandemic Day",
  "Metro-North: % of Comparable Pre-Pandemic Day",
  "Access-A-Ride: % of Comparable Pre-Pandemic Day",
  "Bridges and Tunnels: % of Comparable Pre-Pandemic Day",
  "Staten Island Railway: % of Comparable Pre-Pandemic Day"
)
```

—- KNN —-

```
df_knn <- na.omit(df[, c(predictors, "total_ridership_level")])
```

Because KNN is distance-based it could be good to standardize predictors by scaling them, such as

df_knn_scaled <- df_knn %>%

mutate(across(all_of(predictors), scale))

```
set.seed(1234)

Z = sample(nrow(df_knn), .5*nrow(df_knn))

train = df_knn[Z, predictors]

test = df_knn[-Z, predictors]

cl = df_knn$total_ridership_level[Z]
test_cl = df_knn$total_ridership_level[-Z]
```

```
Yhat <- knn(train, test, cl, k = 3)
```

```
conf_matx <- table(Predicted = Yhat, Actual = test_cl)
conf_matx
```

```
##          Actual
## Predicted Low Average High
##   Low     262      35    1
##   Average  33     230   15
##   High      1      35  276
```

```
accuracy <- sum(diag(conf_matx)) / sum(conf_matx)
accuracy
```

```
## [1] 0.8648649
```

Errors mostly happen around the Low v Average and Average v High boundary. Pretty high accuracy.

```
class_rates <- numeric(100)

for(k in 1:100){
  Yhat_k <- knn(train, test, cl, k = k)

  conf_matx_k <- table(Predicted = Yhat_k, Actual = test_cl)

  class_rates[k] <- sum(diag(conf_matx_k)) / sum(conf_matx_k)
}
```

```
best_k <- which.max(class_rates)
best_k
```

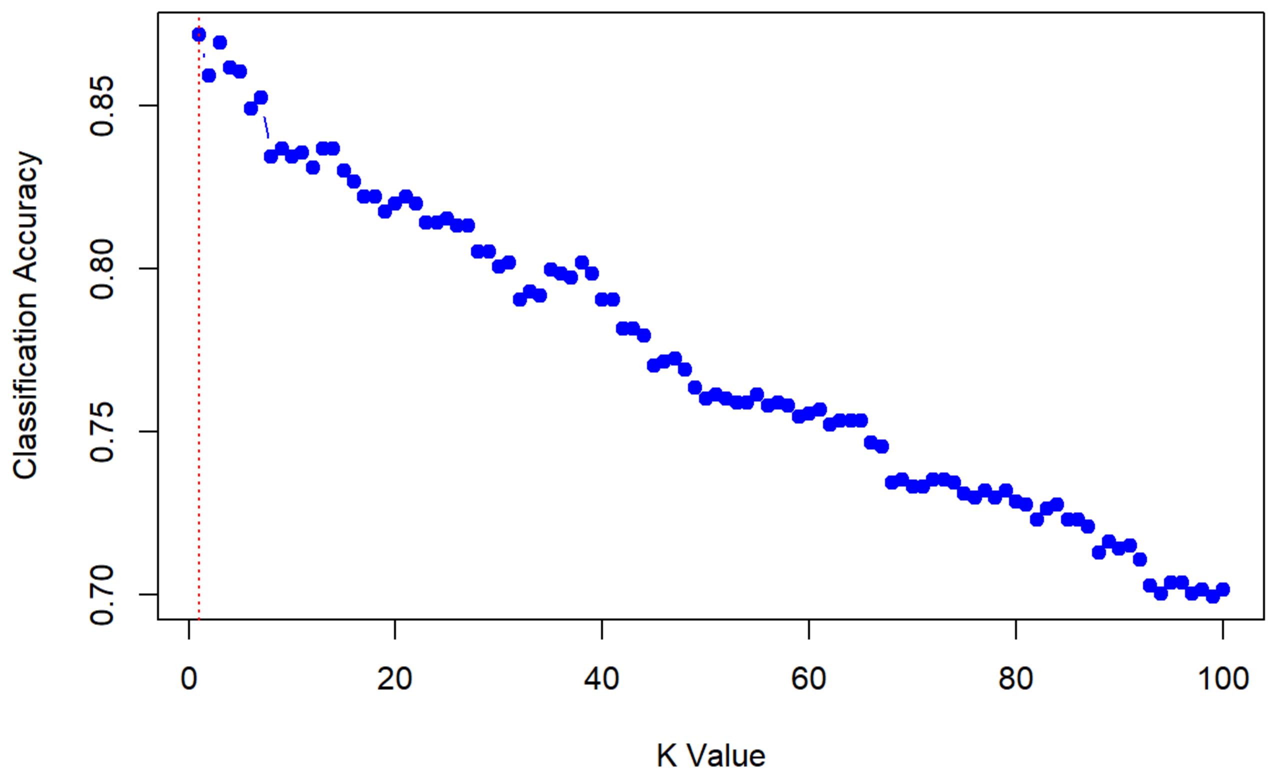
```
## [1] 1
```

k = 1 performed the best but this could be indicative that the model may be overfitting.

```
#Kennedy's comment
"I think there could be some potential overfitting of the training data implied by best_k = 1. T
he model might be learning the noise and specific details of the training set rather than the un
derlying generalizable patterns and i think this will probably lead to poor performance on new,
unseen data. A k=1 model is highly sensitive to outliers

Even if the cross-validation accuracy was high with k=1, it's still susceptible to overfitting,
especially if the dataset has noise or outliers. I have done previous projects with KNN before w
here I get low k values and high accuracy, when those two combine whether or not the model is ac
ceptable will depend on the project's context."
```

```
## [1] "I think there could be some potential overfitting of the training data implied by best_k
= 1. The model might be learning the noise and specific details of the training set rather than
the underlying generalizable patterns and i think this will probably lead to poor performance on
new, unseen data. A k=1 model is highly sensitive to outliers\n\nEven if the cross-validation ac
curacy was high with k=1, it's still susceptible to overfitting, especially if the dataset has n
oise or outliers. I have done previous projects with KNN before where I get low k values and hig
h accuracy, when those two combine whether or not the model is acceptable will depend on the pro
ject's context."
```

```
plot(1:100, class_rates, type = "b", pch = 19, col = "blue",
     xlab = "K Value", ylab = "Classification Accuracy")

abline(v = best_k, col = "red", lty = 3)
```



#

Kennedy’s

```
url <- "https://data.ny.gov/api/views/vxuj-8kew/rows.csv?accessType=DOWNLOAD"
df<- read_csv(url)
```

```
## Rows: 1776 Columns: 15
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (1): Date
## dbl (14): Subways: Total Estimated Ridership, Subways: % of Comparable Pre-P...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
options(scipen = 999)
```

Preparing Data: Regression Section

```
response_var <- "Subways: Total Estimated Ridership"
predictor_vars <- c("Buses: Total Estimated Ridership",
                    "LIRR: Total Estimated Ridership",
                    "Metro-North: Total Estimated Ridership",
                    "Access-A-Ride: Total Scheduled Trips",
                    "Bridges and Tunnels: Total Traffic",
                    "Staten Island Railway: Total Estimated Ridership")

df_clean <- na.omit(df[, c(response_var, predictor_vars)])

# Creating matrices
x <- as.matrix(df_clean[, predictor_vars])
y <- df_clean[[response_var]]

# Train/Test Split
set.seed(168)
train_indices <- sample(1:nrow(x), 0.8 * nrow(x))
x_train <- x[train_indices, ]
y_train <- y[train_indices]
x_test  <- x[-train_indices, ]
y_test  <- y[-train_indices]
```

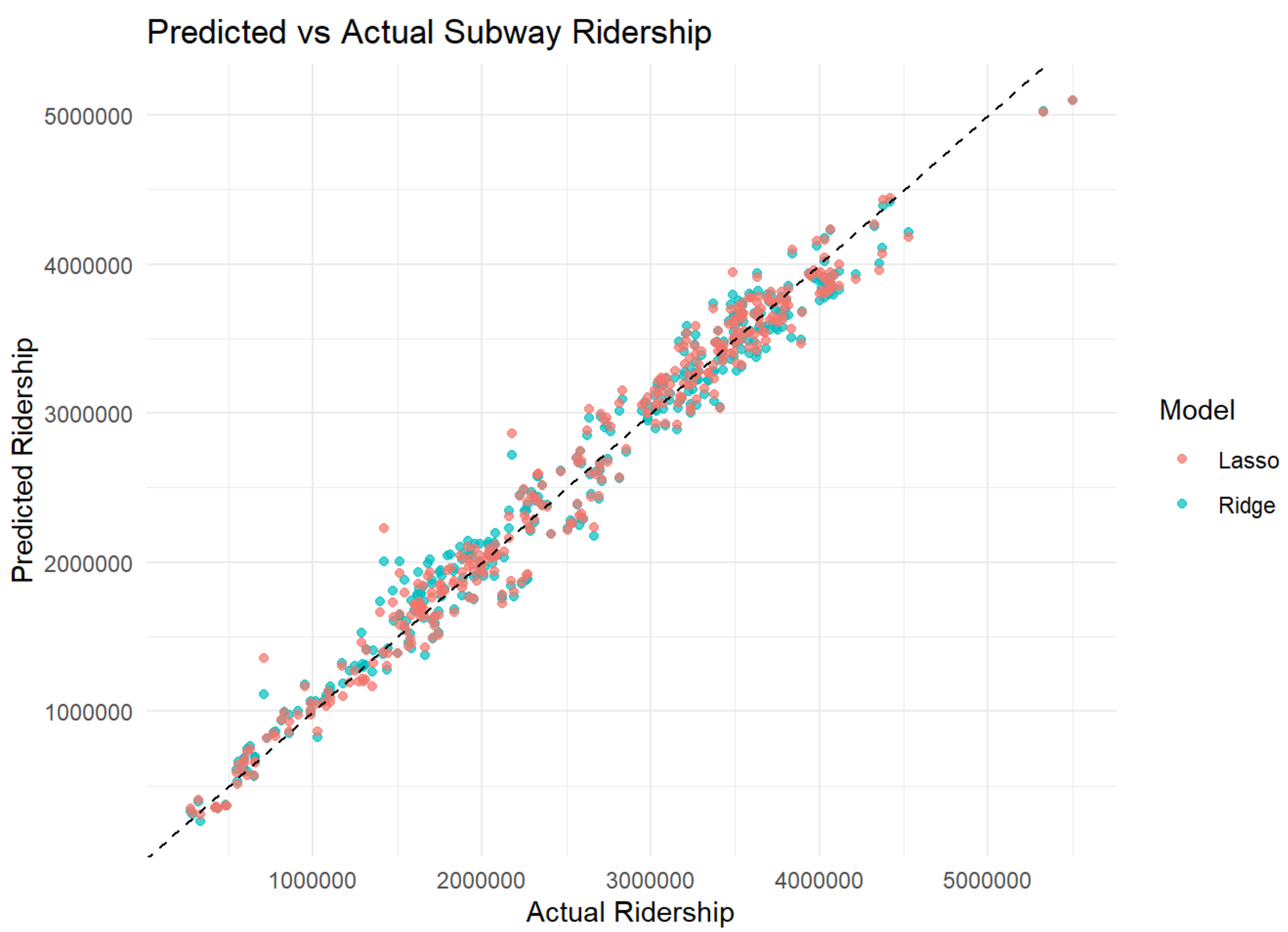
Ridge and Lasso Regression

```
# Ridge
ridge_model <- cv.glmnet(x_train, y_train, alpha = 0)
ridge_pred <- predict(ridge_model, s = ridge_model$lambda.min, newx = x_test)
ridge_mse <- mean((ridge_pred - y_test)^2)

# Lasso
lasso_model <- cv.glmnet(x_train, y_train, alpha = 1)
lasso_pred <- predict(lasso_model, s = lasso_model$lambda.min, newx = x_test)
lasso_mse <- mean((lasso_pred - y_test)^2)

ridge_r2 <- 1 - sum((y_test - ridge_pred)^2) / sum((y_test - mean(y_test))^2)
ridge_rmse <- sqrt(mean((y_test - ridge_pred)^2))

lasso_r2 <- 1 - sum((y_test - lasso_pred)^2) / sum((y_test - mean(y_test))^2)
lasso_rmse <- sqrt(mean((y_test - lasso_pred)^2))


model_summary <- data.frame(
  Model = c("Ridge", "Lasso"),
  R_squared = c(ridge_r2, lasso_r2),
  RMSE = c(ridge_rmse, lasso_rmse),
  MSE = c(ridge_mse, lasso_mse) # Add MSE to the data frame
)

print(model_summary)
```

```
##   Model R_squared     RMSE         MSE
## 1 Ridge 0.9737211 175128.2 30669871224
## 2 Lasso 0.9751850 170180.5 28961399985
```

Ridge vs Lasso: Predicted vs Actual Plot

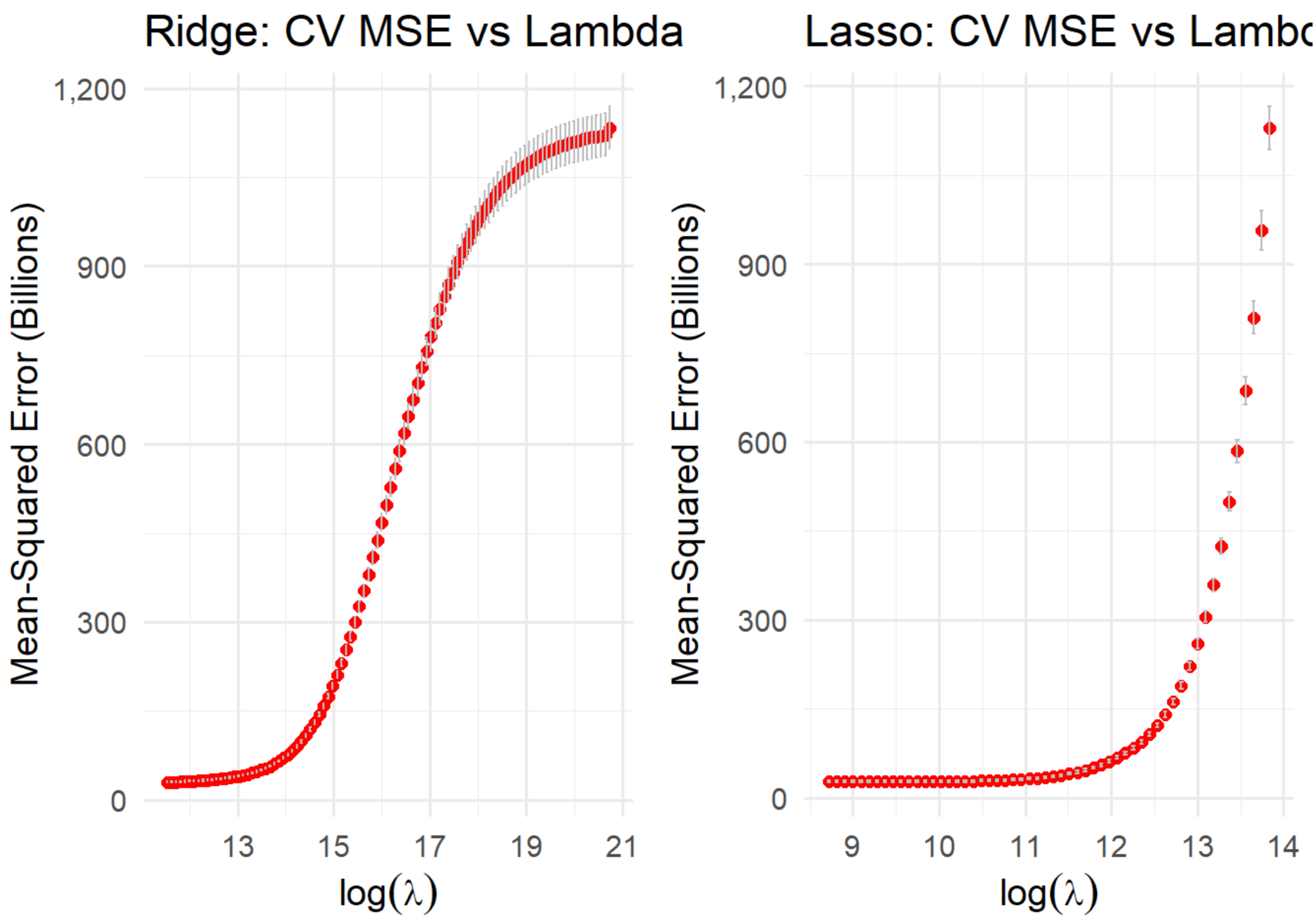
```
ridge_plot_df <- data.frame(Actual = y_test, Predicted = as.numeric(ridge_pred), Model = "Ridg
e")
lasso_plot_df <- data.frame(Actual = y_test, Predicted = as.numeric(lasso_pred), Model = "Lass
o")
all_predictions_df <- rbind(ridge_plot_df, lasso_plot_df)

ggplot(all_predictions_df, aes(x = Actual, y = Predicted, color = Model)) +
  geom_point(alpha = 0.7) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "black") +
  labs(
    title = "Predicted vs Actual Subway Ridership",
    x = "Actual Ridership",
    y = "Predicted Ridership",
    color = "Model"
  ) +
  theme_minimal()
```



```
# This predicted vs actual subway ridership plot indicates that our Ridge and Lasso regression m
odels are performing reasonably well in predicting subway ridership based on the other transport
ation metrics.

# Comparing our MSE, we can see that the Lasso regression model has a slightly lower Mean Square
d Error than the Ridge regression model on our test data. A lower MSE indicates that, on averag
e, the predictions made by the Lasso model are closer to the actual subway ridership values in t
he test set compared to the Ridge model. The difference in MSE, while present, might not be dras
tically large. It suggests that Lasso has a marginal improvement in predictive accuracy for this
specific dataset and the chosen model parameters.
```

Ridge/Lasso CV Plots

```
# Creating data frames for ridge and lasso
ridge_df <- data.frame(
  log_lambda = log(ridge_model$lambda),
  mse_mean = ridge_model$cvm,
  mse_upper = ridge_model$cvup,
  mse_lower = ridge_model$cvlo
)

lasso_df <- data.frame(
  log_lambda = log(lasso_model$lambda),
  mse_mean = lasso_model$cvm,
  mse_upper = lasso_model$cvup,
  mse_lower = lasso_model$cvlo
)

ridge_df <- ridge_df %>% mutate(across(c(mse_mean, mse_upper, mse_lower), ~ .x / 1e9))
lasso_df <- lasso_df %>% mutate(across(c(mse_mean, mse_upper, mse_lower), ~ .x / 1e9))

# Ridge plot
ridge_plot <- ggplot(ridge_df, aes(x = log_lambda, y = mse_mean)) +
  geom_point(color = "red", size = 2) +
  geom_errorbar(aes(ymin = mse_lower, ymax = mse_upper), width = 0.05, color = "gray") +
  labs(title = "Ridge: CV MSE vs Lambda",
       x = expression(log(lambda)),
       y = "Mean-Squared Error (Billions)") +
  scale_y_continuous(labels = scales::comma) +  # <-- nice labels: 0, 200, 400
  theme_minimal(base_size = 14) # slightly larger base font

# Lasso plot
lasso_plot <- ggplot(lasso_df, aes(x = log_lambda, y = mse_mean)) +
  geom_point(color = "red", size = 2) +
  geom_errorbar(aes(ymin = mse_lower, ymax = mse_upper), width = 0.05, color = "gray") +
  labs(title = "Lasso: CV MSE vs Lambda",
       x = expression(log(lambda)),
       y = "Mean-Squared Error (Billions)") +
  scale_y_continuous(labels = scales::comma) +
  theme_minimal(base_size = 14)

grid.arrange(ridge_plot, lasso_plot, ncol = 2)
```




```
"Ridge: 'Hockey stick' MSE curve vs. log(λ), low MSE at low λ (overfitting), increases with regu
larization (underfitting), plateaus at high λ. Optimal λ suggests moderate regularization preven
ts overfitting. Small error bars indicate consistent CV performance. Minimum MSE in lower tens o
f billions.

Lasso: MSE increases with log(λ) from minimum, potentially steeper than Ridge. Optimal λ lower,
suggesting less regularization needed. Small error bars. Minimum MSE in lower tens of billions,
possibly slightly better than Ridge.

Comparison: Similar minimum MSE for both (billions indicate substantial unexplained variance). L
asso prefers less regularization, implying potential feature selection of less influential predi
ctors, offering insight into key factors. Both handle multicollinearity. Lasso's slight edge hin
ts at removing less relevant variables."
```

```
## [1] "Ridge: 'Hockey stick' MSE curve vs. log(λ), low MSE at low λ (overfitting), increases wi
th regularization (underfitting), plateaus at high λ. Optimal λ suggests moderate regularization
prevents overfitting. Small error bars indicate consistent CV performance. Minimum MSE in lower
tens of billions.\n\nLasso: MSE increases with log(λ) from minimum, potentially steeper than Rid
ge. Optimal λ lower, suggesting less regularization needed. Small error bars. Minimum MSE in low
er tens of billions, possibly slightly better than Ridge.\n\nComparison: Similar minimum MSE for
both (billions indicate substantial unexplained variance). Lasso prefers less regularization, im
plying potential feature selection of less influential predictors, offering insight into key fac
tors. Both handle multicollinearity. Lasso's slight edge hints at removing less relevant variabl
es."
```

Classification: Decision Tree and Random Forest

```
# Creating categorical target variable
quantiles <- quantile(df_clean[[response_var]], probs = c(0.33, 0.66))
df_clean$ridership_class <- cut(df_clean[[response_var]],
                                breaks = c(-Inf, quantiles, Inf),
                                labels = c("Low", "Medium", "High"),
                                right = TRUE)

# Class distribution
table(df_clean$ridership_class)
```

```
##
##    Low Medium   High
##    586    586    604
```

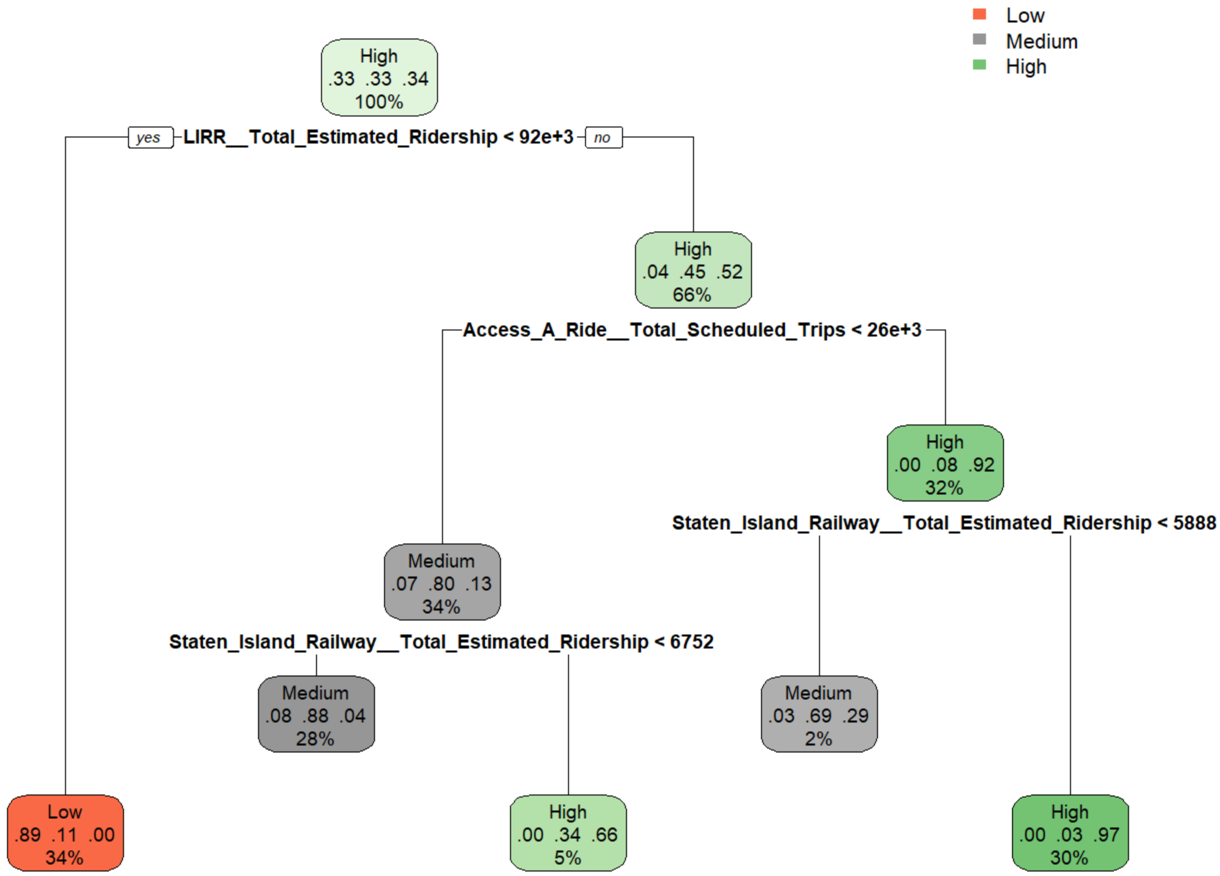
```
# Train/Test split for classification
set.seed(168)
train_index <- createDataPartition(df_clean$ridership_class, p = 0.8, list = FALSE)

train_data <- df_clean[train_index, ]
test_data  <- df_clean[-train_index, ]

# Cleaning column names (no colons/spaces)
train_data_clean <- train_data %>%
  rename_with(~ gsub("[^[:alnum:]_]", "_", .))

test_data_clean <- test_data %>%
  rename_with(~ gsub("[^[:alnum:]_]", "_", .))
```

Training Decision Tree

```
tree_model <- rpart(ridership_class ~ Buses__Total_Estimated_Ridership +
                                    LIRR__Total_Estimated_Ridership +
                                    Metro_North__Total_Estimated_Ridership +
                                    Access_A_Ride__Total_Scheduled_Trips +
                                    Bridges_and_Tunnels__Total_Traffic +
                                    Staten_Island_Railway__Total_Estimated_Ridership,
                    data = train_data_clean,
                    method = "class",
                    control = rpart.control(maxdepth = 5, minsplit = 30))

# Checking the complexity parameter to prune
printcp(tree_model)
```

```
##
## Classification tree:
## rpart(formula = ridership_class ~ Buses__Total_Estimated_Ridership +
##     LIRR__Total_Estimated_Ridership + Metro_North__Total_Estimated_Ridership +
##     Access_A_Ride__Total_Scheduled_Trips + Bridges_and_Tunnels__Total_Traffic +
##     Staten_Island_Railway__Total_Estimated_Ridership, data = train_data_clean,
##     method = "class", control = rpart.control(maxdepth = 5, minsplit = 30))
##
## Variables actually used in tree construction:
## [1] Access_A_Ride__Total_Scheduled_Trips
## [2] LIRR__Total_Estimated_Ridership
## [3] Metro_North__Total_Estimated_Ridership
## [4] Staten_Island_Railway__Total_Estimated_Ridership
##
## Root node error: 938/1422 = 0.65963
##
## n= 1422
##
##         CP nsplit rel error  xerror     xstd
## 1 0.463753      0   1.00000 1.03625 0.018698
## 2 0.339019      1   0.53625 0.54051 0.019256
## 3 0.024520      2   0.19723 0.22495 0.014291
## 4 0.014925      3   0.17271 0.18230 0.013076
## 5 0.010661      4   0.15778 0.17271 0.012773
## 6 0.010000      5   0.14712 0.17591 0.012875
```

```
# Pruning
pruned_tree <- prune(tree_model, cp = tree_model$cptable[which.min(tree_model$cptable[,"xerro
r"]),"CP"])

rpart.plot(pruned_tree, type = 2, extra = 104, fallen.leaves = TRUE)
```




```
"The decision tree classifies daily subway ridership (low, average, high) using other MTA servic
es' ridership. LIRR, Metro-North, Access-A-Ride, and Staten Island Railway ridership were key pr
edictors, suggesting interconnected demand. Surprisingly, bus ridership and bridge/tunnel traffi
c were not significant in this model. The initial guessing error was high (66%), highlighting th
e model's value. By pruning the tree using cross-validation error, we aim for a model that accur
ately classifies future subway ridership based on these relationships. The tree's rules offer in
sights into how demand across different transit modes interacts, which can inform future, integr
ated transportation policies and service planning."
```


```
## [1] "The decision tree classifies daily subway ridership (low, average, high) using other MTA
services' ridership. LIRR, Metro-North, Access-A-Ride, and Staten Island Railway ridership were
key predictors, suggesting interconnected demand. Surprisingly, bus ridership and bridge/tunnel
traffic were not significant in this model. The initial guessing error was high (66%), highlight
ing the model's value. By pruning the tree using cross-validation error, we aim for a model that
accurately classifies future subway ridership based on these relationships. The tree's rules off
er insights into how demand across different transit modes interacts, which can inform future, i
ntegrated transportation policies and service planning."
```

Train Random Forest

```
rf_model <- randomForest(ridership_class ~ Buses__Total_Estimated_Ridership +
                                          LIRR__Total_Estimated_Ridership +
                                          Metro_North__Total_Estimated_Ridership +
                                          Access_A_Ride__Total_Scheduled_Trips +
                                          Bridges_and_Tunnels__Total_Traffic +
                                          Staten_Island_Railway__Total_Estimated_Ridership,
                         data = train_data_clean,
                         ntree = 500,
                         importance = TRUE)

print(rf_model)
```

```
##
## Call:
##  randomForest(formula = ridership_class ~ Buses__Total_Estimated_Ridership +      LIRR__Total
_Estimated_Ridership + Metro_North__Total_Estimated_Ridership +      Access_A_Ride__Total_Schedu
led_Trips + Bridges_and_Tunnels__Total_Traffic +      Staten_Island_Railway__Total_Estimated_Rid
ership, data = train_data_clean,      ntree = 500, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 5.98%
## Confusion matrix:
##        Low Medium High class.error
## Low    447     22    0  0.04690832
## Medium  21    427   21  0.08955224
## High     0     21  463  0.04338843
```

Evaluating Tree and RF

```
tree_pred <- predict(tree_model, test_data_clean, type = "class")
rf_pred   <- predict(rf_model, test_data_clean)

# Confusion Matrices
cat("Decision Tree Confusion Matrix:\n")
```

```
## Decision Tree Confusion Matrix:
```

```
print(confusionMatrix(tree_pred, test_data_clean$ridership_class))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Low Medium High
##     Low    111     11    0
##     Medium   6    100   15
##     High     0      6  105
##
## Overall Statistics
##
##                Accuracy : 0.8927
##                  95% CI : (0.8556, 0.9229)
##     No Information Rate : 0.339
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.839
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Low Class: Medium Class: High
## Sensitivity              0.9487        0.8547      0.8750
## Specificity              0.9536        0.9114      0.9744
## Pos Pred Value           0.9098        0.8264      0.9459
## Neg Pred Value           0.9741        0.9270      0.9383
## Prevalence               0.3305        0.3305      0.3390
## Detection Rate           0.3136        0.2825      0.2966
## Detection Prevalence     0.3446        0.3418      0.3136
## Balanced Accuracy        0.9512        0.8830      0.9247
```

```
cat("\nRandom Forest Confusion Matrix:\n")
```

```
##
## Random Forest Confusion Matrix:
```

```
print(confusionMatrix(rf_pred, test_data_clean$ridership_class))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Low Medium High
##     Low    111      3    0
##     Medium   6    107    8
##     High     0      7  112
##
## Overall Statistics
##
##                Accuracy : 0.9322
##                  95% CI : (0.9008, 0.9561)
##     No Information Rate : 0.339
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.8983
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Low Class: Medium Class: High
## Sensitivity              0.9487        0.9145      0.9333
## Specificity              0.9873        0.9409      0.9701
## Pos Pred Value           0.9737        0.8843      0.9412
## Neg Pred Value           0.9750        0.9571      0.9660
## Prevalence               0.3305        0.3305      0.3390
## Detection Rate           0.3136        0.3023      0.3164
## Detection Prevalence     0.3220        0.3418      0.3362
## Balanced Accuracy        0.9680        0.9277      0.9517
```

```
"Overall Accuracy: Random Forest achieves a higher accuracy (93.22%) compared to the Decision Tr
ee (89.27%), indicating a greater ability to correctly classify ridership levels.

Kappa Statistic: Random Forest has a substantially higher Kappa (0.8983 vs. 0.839), suggesting b
etter agreement between predicted and actual classifications beyond chance.

Sensitivity (Recall): While the sensitivity for the 'Low' class is similar for both, Random Fore
st shows higher sensitivity for 'Medium' (91.45% vs. 85.47%) and 'High' (93.33% vs. 87.50%) ride
rship, meaning it's better at correctly identifying these categories.

Specificity: Random Forest exhibits higher specificity across all classes, particularly for 'Lo
w' (98.73% vs. 95.36%) and 'Medium' (94.09% vs. 91.14%), indicating a better ability to correctl
y identify days that do not belong to each ridership level.

Positive Predictive Value (Precision): Random Forest has a higher precision for 'Low' (97.37% v
s. 90.98%) and 'Medium' (88.43% vs. 82.64%) ridership, meaning when it predicts these levels, it
is more likely to be correct. The precision for 'High' is comparable.

Balanced Accuracy: Random Forest demonstrates higher balanced accuracy across all classes, espec
ially for 'Medium' (92.77% vs. 88.30%) and 'High' (95.17% vs. 92.47%), indicating better perform
ance when class imbalance is considered."
```

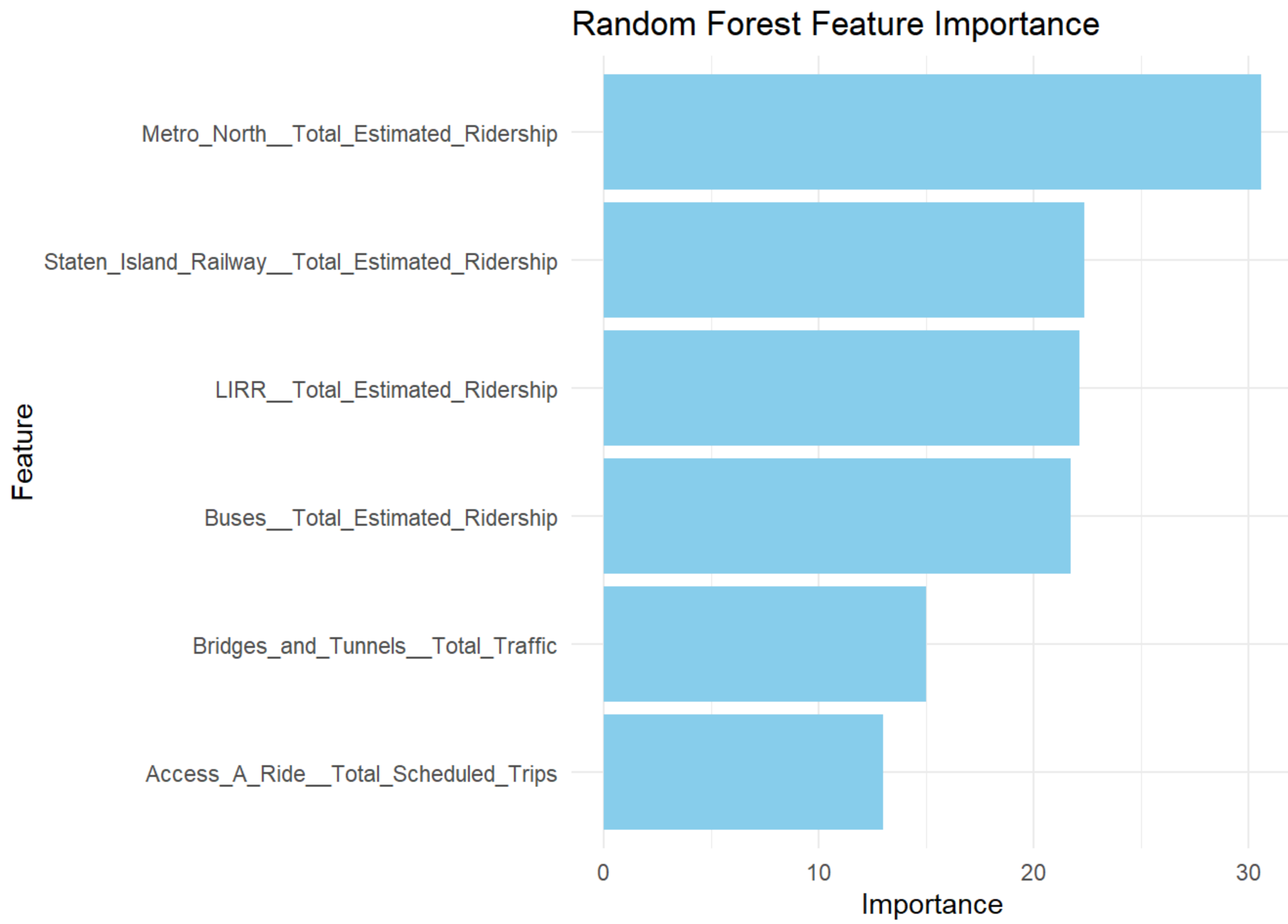
```
## [1] "Overall Accuracy: Random Forest achieves a higher accuracy (93.22%) compared to the Deci
sion Tree (89.27%), indicating a greater ability to correctly classify ridership levels.\n\nKapp
a Statistic: Random Forest has a substantially higher Kappa (0.8983 vs. 0.839), suggesting bette
r agreement between predicted and actual classifications beyond chance.\n\nSensitivity (Recall):
While the sensitivity for the 'Low' class is similar for both, Random Forest shows higher sensit
ivity for 'Medium' (91.45% vs. 85.47%) and 'High' (93.33% vs. 87.50%) ridership, meaning it's be
tter at correctly identifying these categories.\n\nSpecificity: Random Forest exhibits higher sp
ecificity across all classes, particularly for 'Low' (98.73% vs. 95.36%) and 'Medium' (94.09% v
s. 91.14%), indicating a better ability to correctly identify days that do not belong to each ri
dership level.\n\nPositive Predictive Value (Precision): Random Forest has a higher precision fo
r 'Low' (97.37% vs. 90.98%) and 'Medium' (88.43% vs. 82.64%) ridership, meaning when it predicts
these levels, it is more likely to be correct. The precision for 'High' is comparable.\n\nBalanc
ed Accuracy: Random Forest demonstrates higher balanced accuracy across all classes, especially
for 'Medium' (92.77% vs. 88.30%) and 'High' (95.17% vs. 92.47%), indicating better performance w
hen class imbalance is considered."
```

Variable Importance (Random Forest)

```
importance(rf_model)
```

```
##                                                     Low   Medium      High
## Buses__Total_Estimated_Ridership                21.71787 22.71041 24.249959
## LIRR__Total_Estimated_Ridership                 22.15495 21.74889 21.982620
## Metro_North__Total_Estimated_Ridership          30.60738 30.80731 23.963968
## Access_A_Ride__Total_Scheduled_Trips            12.99542 20.86296 31.528280
## Bridges_and_Tunnels__Total_Traffic              14.97612 13.81227  8.229463
## Staten_Island_Railway__Total_Estimated_Ridership 22.36018 18.10143 33.079928
##                                                 MeanDecreaseAccuracy
## Buses__Total_Estimated_Ridership                            36.78954
## LIRR__Total_Estimated_Ridership                             32.22560
## Metro_North__Total_Estimated_Ridership                      40.02196
## Access_A_Ride__Total_Scheduled_Trips                        36.90594
## Bridges_and_Tunnels__Total_Traffic                          20.28672
## Staten_Island_Railway__Total_Estimated_Ridership            41.31072
##                                                 MeanDecreaseGini
## Buses__Total_Estimated_Ridership                       118.02604
## LIRR__Total_Estimated_Ridership                        212.14257
## Metro_North__Total_Estimated_Ridership                 202.82681
## Access_A_Ride__Total_Scheduled_Trips                   168.55599
## Bridges_and_Tunnels__Total_Traffic                      55.01518
## Staten_Island_Railway__Total_Estimated_Ridership       190.67102
```

```
rf_importance <- importance(rf_model)
rf_importance_df <- data.frame(Feature = rownames(rf_importance), Importance = rf_importance[,
1])
rf_importance_df <- rf_importance_df[order(-rf_importance_df$Importance),]

ggplot(rf_importance_df, aes(x = reorder(Feature, Importance), y = Importance)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  coord_flip() +
  labs(title = "Random Forest Feature Importance", x = "Feature", y = "Importance") +
  theme_minimal()
```



Accuracy

```
tree_accuracy <- sum(tree_pred == test_data_clean$ridership_class) / length(tree_pred)
rf_accuracy <- sum(rf_pred == test_data_clean$ridership_class) / length(rf_pred)

cat(sprintf("Decision Tree Accuracy: %.2f%%\n", tree_accuracy * 100))
```

```
## Decision Tree Accuracy: 89.27%
```

```
cat(sprintf("Random Forest Accuracy: %.2f%%\n", rf_accuracy * 100))
```

```
## Random Forest Accuracy: 93.22%
```

Cross-Validation: K-Fold CV for Decision Trees

```
set.seed(168)
train_control <- trainControl(method = "cv", number = 10)

# Train CV Tree model
tree_cv_model <- train(ridership_class ~ Buses__Total_Estimated_Ridership +
                                    LIRR__Total_Estimated_Ridership +
                                    Metro_North__Total_Estimated_Ridership +
                                    Access_A_Ride__Total_Scheduled_Trips +
                                    Bridges_and_Tunnels__Total_Traffic +
                                    Staten_Island_Railway__Total_Estimated_Ridership,
                       data = train_data_clean,
                       method = "rpart",
                       trControl = train_control)

# CV Model Summary
print(tree_cv_model)
```

```
## CART
##
## 1422 samples
##    6 predictor
##    3 classes: 'Low', 'Medium', 'High'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1280, 1280, 1279, 1280, 1280, 1279, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.02452026  0.8741740  0.8112331
##   0.33901919  0.7092105  0.5628808
##   0.46375267  0.5859267  0.3752488
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.02452026.
```

```
# Prediction using CV model
tree_pred_cv <- predict(tree_cv_model, test_data_clean)

# Confusion Matrix
cat("Decision Tree (CV) Confusion Matrix:\n")
```

```
## Decision Tree (CV) Confusion Matrix:
```

```
print(confusionMatrix(tree_pred_cv, test_data_clean$ridership_class))
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction Low Medium High
##     Low    111     11    0
##     Medium   6     94   15
##     High     0     12  105
##
## Overall Statistics
##
##                Accuracy : 0.8757
##                  95% CI : (0.8368, 0.9082)
##     No Information Rate : 0.339
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.8136
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Low Class: Medium Class: High
## Sensitivity              0.9487        0.8034      0.8750
## Specificity              0.9536        0.9114      0.9487
## Pos Pred Value           0.9098        0.8174      0.8974
## Neg Pred Value           0.9741        0.9038      0.9367
## Prevalence               0.3305        0.3305      0.3390
## Detection Rate           0.3136        0.2655      0.2966
## Detection Prevalence     0.3446        0.3249      0.3305
## Balanced Accuracy        0.9512        0.8574      0.9119
```

```
"Accuracy: Random Forest (93.22%) still significantly outperforms the cross-validated Decision T
ree (87.57%).

Kappa: Random Forest (0.8983) shows substantially better agreement than the cross-validated Deci
sion Tree (0.8136).

Sensitivity: Random Forest generally maintains higher or comparable sensitivity across the class
es, 'particularly for 'Medium' and 'High' ridership.

Specificity: Random Forest consistently shows higher specificity, indicating a better ability to
correctly identify days not belonging to each class.

Balanced Accuracy: Random Forest exhibits higher balanced accuracy across all classes, suggestin
g more robust performance when considering potential class imbalance."
```

```
## [1] "Accuracy: Random Forest (93.22%) still significantly outperforms the cross-validated Dec
ision Tree (87.57%).\n\nKappa: Random Forest (0.8983) shows substantially better agreement than
the cross-validated Decision Tree (0.8136).\n\nSensitivity: Random Forest generally maintains hi
gher or comparable sensitivity across the classes, 'particularly for 'Medium' and 'High' ridersh
ip.\n\nSpecificity: Random Forest consistently shows higher specificity, indicating a better abi
lity to correctly identify days not belonging to each class.\n\nBalanced Accuracy: Random Forest
exhibits higher balanced accuracy across all classes, suggesting more robust performance when co
nsidering potential class imbalance."
```

```
" Over all, The strong performance of the Random Forest model highlights just how many different
factors influence subway ridership. Both models show that commuter rail services like the LIRR a
nd Metro-North play a major role, even though each model weighs their importance a little differ
ently. This points to a strong connection between regional commuting patterns and how people use
the subway.

Access-A-Ride also showed up as a key factor, suggesting that paratransit demand might be linked
to specific levels of subway ridership, maybe reflecting the needs of certain groups of riders w
ho rely more heavily on these services.

On the other hand, bus ridership and bridge/tunnel traffic were less important in the Random For
est model, and they didn’t even appear in the decision tree structure. This suggests that while
buses and car traffic might affect transit patterns in general, they’re not the strongest indica
tors when it comes to classifying daily subway ridership as 'low,' 'average,' or 'high.' That sa
id, they could still be valuable for predicting the actual number of riders (as we saw in the re
gressiolln models) or for understanding more localized transit behaviors.

The Random Forest's high accuracy rate (93.22%) gives us a lot of confidence in its ability to p
redict daily ridership categories moving forward. This could be incredibly useful for the MTA wh
en it comes to planning ahead, such as adjusting train schedules, staffing, and resource allocat
ion based on expected rider volumes. For example, if we can predict a 'high' ridership day ahead
of time based on trends in commuter rail or paratransit usage, the MTA could proactively add ser
vice and better manage crowding.

Finally, the insights from the feature importance analysis can help guide bigger policy decision
s. Knowing which modes of transit have the strongest ties to subway ridership could support more
integrated planning and investment strategies across the system. For instance, encouraging commu
ter rail use might have predictable ripple effects on subway demand."
```

```
## [1] " Over all, The strong performance of the Random Forest model highlights just how many di
fferent factors influence subway ridership. Both models show that commuter rail services like th
e LIRR and Metro-North play a major role, even though each model weighs their importance a littl
e differently. This points to a strong connection between regional commuting patterns and how pe
ople use the subway.\n\nAccess-A-Ride also showed up as a key factor, suggesting that paratransi
t demand might be linked to specific levels of subway ridership, maybe reflecting the needs of c
ertain groups of riders who rely more heavily on these services.\n\nOn the other hand, bus rider
ship and bridge/tunnel traffic were less important in the Random Forest model, and they didn’t e
ven appear in the decision tree structure. This suggests that while buses and car traffic might 
affect transit patterns in general, they’re not the strongest indicators when it comes to classi
fying daily subway ridership as 'low,' 'average,' or 'high.' That said, they could still be valu
able for predicting the actual number of riders (as we saw in the regressiolln models) or for un
derstanding more localized transit behaviors.\n\nThe Random Forest's high accuracy rate (93.22%)
gives us a lot of confidence in its ability to predict daily ridership categories moving forwar
d. This could be incredibly useful for the MTA when it comes to planning ahead, such as adjustin
g train schedules, staffing, and resource allocation based on expected rider volumes. For exampl
e, if we can predict a 'high' ridership day ahead of time based on trends in commuter rail or pa
ratransit usage, the MTA could proactively add service and better manage crowding.\n\nFinally, t
he insights from the feature importance analysis can help guide bigger policy decisions. Knowing
which modes of transit have the strongest ties to subway ridership could support more integrated
planning and investment strategies across the system. For instance, encouraging commuter rail us
e might have predictable ripple effects on subway demand."
```

combined regression plot

```
response_var <- "Subways: Total Estimated Ridership"
predictor_vars <- c("Buses: Total Estimated Ridership",
                    "LIRR: Total Estimated Ridership",
                    "Metro-North: Total Estimated Ridership",
                    "Access-A-Ride: Total Scheduled Trips",
                    "Bridges and Tunnels: Total Traffic",
                    "Staten Island Railway: Total Estimated Ridership")

df_clean <- na.omit(df[, c(response_var, predictor_vars)])


x <- as.matrix(df_clean[, predictor_vars])
y <- df_clean[[response_var]]


set.seed(1234)
train_indices <- sample(1:nrow(x), 0.8 * nrow(x))
x_train <- x[train_indices, ]
y_train <- y[train_indices]
x_test  <- x[-train_indices, ]
y_test  <- y[-train_indices]


ridge_model <- cv.glmnet(x_train, y_train, alpha = 0)
ridge_pred <- predict(ridge_model, s = ridge_model$lambda.min, newx = x_test)


lasso_model <- cv.glmnet(x_train, y_train, alpha = 1)
lasso_pred <- predict(lasso_model, s = lasso_model$lambda.min, newx = x_test)


full_model <- lm(`Subways: Total Estimated Ridership` ~ ., data = df)
predicted_full <- predict(full_model, df)


mlr_plot_df <- data.frame(Actual = df$`Subways: Total Estimated Ridership`,
                          Predicted = predicted_full,
                          Model = "Linear Regression")

ridge_plot_df <- data.frame(Actual = y_test,
                            Predicted = as.numeric(ridge_pred),
                            Model = "Ridge Regression")

lasso_plot_df <- data.frame(Actual = y_test,
                            Predicted = as.numeric(lasso_pred),
                            Model = "Lasso Regression")


combined_df <- rbind(mlr_plot_df, ridge_plot_df, lasso_plot_df)


ggplot(combined_df, aes(x = Actual, y = Predicted, color = Model)) +
```

```
  geom_point(alpha = 0.6) +
  geom_abline(intercept = 0, slope = 1, color = "black", linetype = "dashed") +
  labs(
    title = "\n          Predicted vs Actual Subway Ridership",
    x = "Actual Ridership (Billions)",
    y = "Predicted Ridership (Billions)",
    color = "Model"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, face = "bold"),
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10)
  )
```