EDA_Apr22

Andrew Chung

2025-04-23

Exploratory Data Analysis, April 22

Andrew Chung, hc893

BTRY 4100 Final Project – Exploratory Data Analysis.

Line Performance Metrics Data EDA

```
line_data = read.csv("C:\\Users\\hychu\\OneDrive\\Desktop\\SP25\\_BTRY_4100\\_FINAL_PROJECT\\MTA-NYCSubw
line_data$month = as.numeric(format(as.POSIXct(line_data$month, format = "%Y-%m-%d"), "%m"))
head(line_data)
```

```
##   line month num_passengers additional_platform_time additional_train_time
## 1    1     1        6688313                0.6999114             0.5234027
## 2    1     2        5147344                1.0039914             0.5949038
## 3    2     1        4378540                1.1232483             0.5872850
## 4    2     2        3307910                1.1653521             0.5325663
## 5    3     1        3382492                0.8991957             0.5622657
## 6    3     2        2608766                0.6935546             0.5405329
##   over_five_mins_perc wait_assessment service_delivered
## 1          0.07730745       0.7787605         0.9906900
## 2          0.09609109       0.7481025         0.9746354
## 3          0.14276768       0.6507053         0.9361881
## 4          0.13321753       0.6777709         0.9422222
## 5          0.11015543       0.6982059         0.9438454
## 6          0.09561981       0.7052005         0.9531416
##   terminal_on_time_performance percent_late infra_critical noninfra_critical
## 1                    0.8372688   0.03394333              0                  0
## 2                    0.8084304   0.04146676              4                  1
## 3                    0.7133803   0.04169014              3                  2
## 4                    0.7185355   0.04396862              1                  2
## 5                    0.8223103   0.03185005              3                  2
## 6                    0.8009368   0.03927220              0                  0
##   infra_noncritical noninfra_noncritical
## 1              1082                  572
## 2              1200                  477
## 3              1145                  890
## 4              1042                  680
## 5               593                  573
## 6               686                  419
```

```r
line_data_jan = line_data[line_data$month == 1, ]
line_data_feb = line_data[line_data$month == 2, ]
```

Monthly Passenger Volume

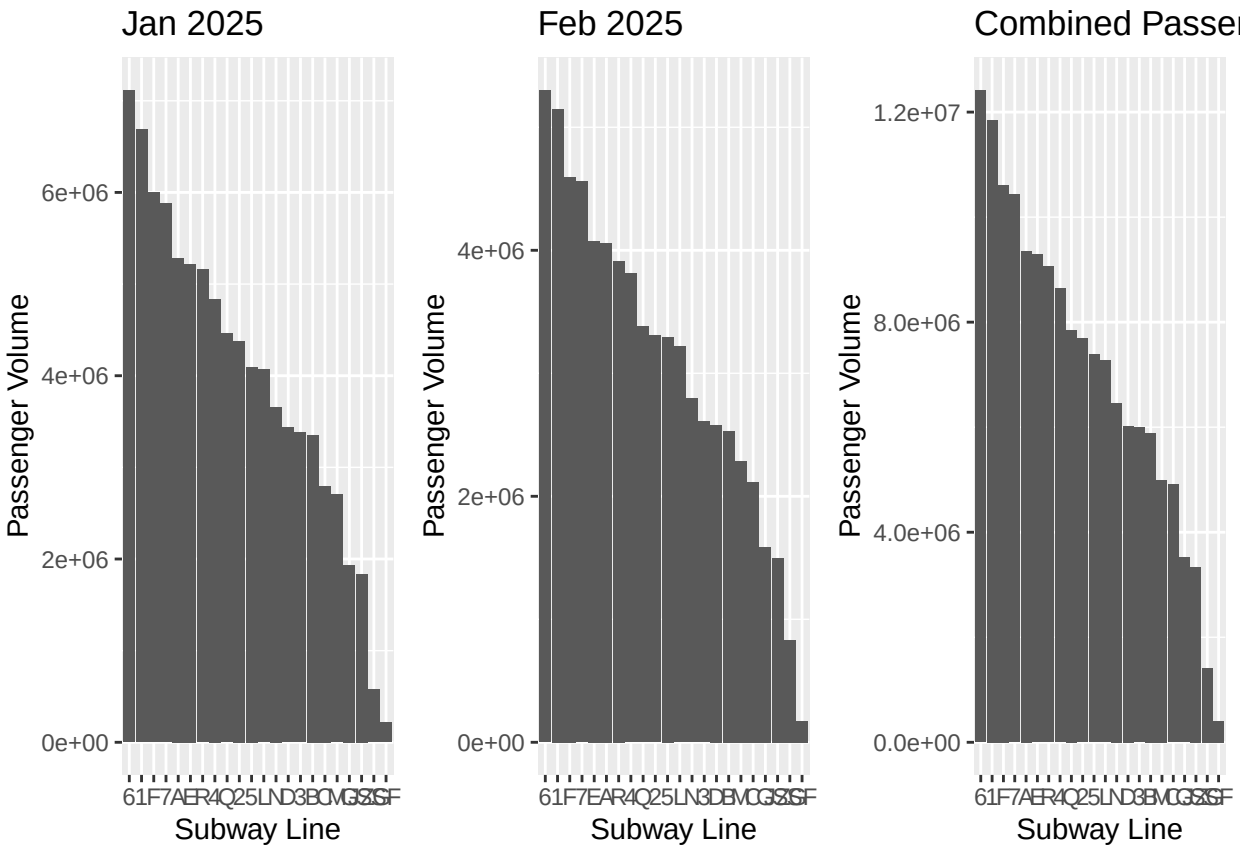
```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

```r
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.4.3
```

```r
plot_jan = ggplot(line_data_jan, aes(x = reorder(line, -num_passengers), y = num_passengers)) + geom_ba
    x = "Subway Line",
    y = "Passenger Volume",
    title = "Jan 2025" # Chart title
)
plot_feb = ggplot(line_data_feb, aes(x = reorder(line, -num_passengers), y = num_passengers)) + geom_ba
    x = "Subway Line",
    y = "Passenger Volume",
    title = "Feb 2025" # Chart title
)
plot_combined = ggplot(line_data, aes(x = reorder(line, -num_passengers), y = num_passengers)) + geom_b
    x = "Subway Line",
    y = "Passenger Volume",
    title = "Combined Passenger Volume" # Chart title
)
grid.arrange(plot_jan, plot_feb, plot_combined, nrow = 1)
```



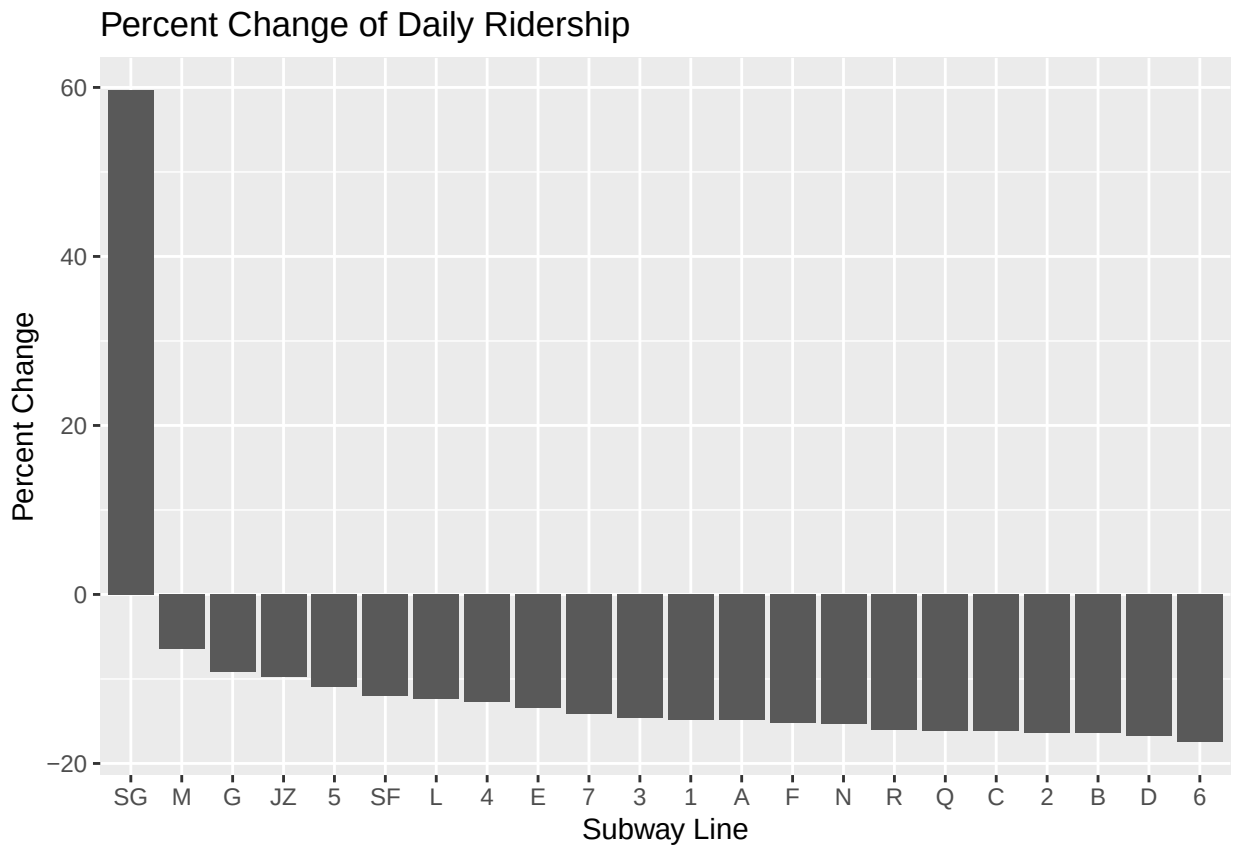
Change in Daily Passenger Volume from Jan to Feb

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.2

## Warning: package 'tidyr' was built under R version 4.4.2

## Warning: package 'readr' was built under R version 4.4.2

## Warning: package 'purrr' was built under R version 4.4.3

## Warning: package 'stringr' was built under R version 4.4.2

## Warning: package 'forcats' was built under R version 4.4.2

## Warning: package 'lubridate' was built under R version 4.4.3

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.4     v tibble    3.2.1
```

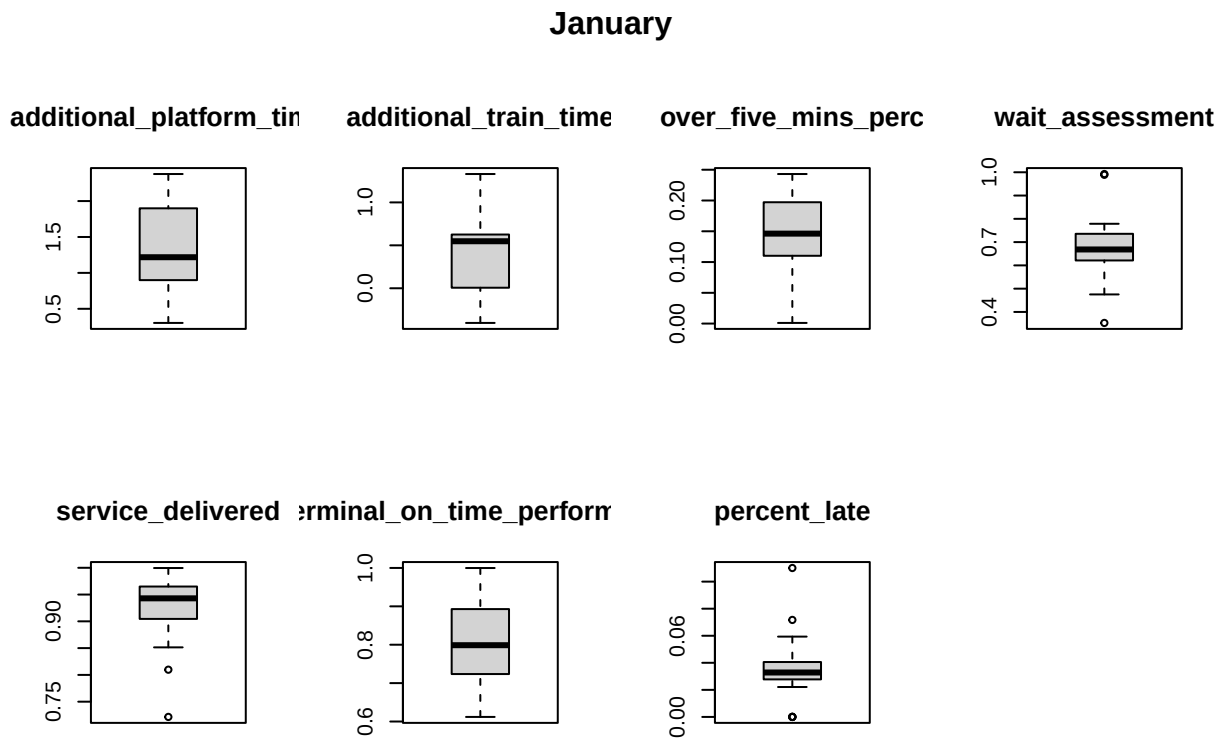
```
## v purrr     1.0.4     v tidyr     1.3.1
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::combine() masks gridExtra::combine()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
delta_data = merge(
  line_data_jan[, c("line", "num_passengers")],
  line_data_feb[, c("line", "num_passengers")],
  by = "line"
) %>%
  rename(
    jan = num_passengers.x,
    feb = num_passengers.y
  ) %>%
  mutate(jan = jan/31, feb = feb/28) %>%
  mutate(percent_change = 100 * (feb-jan)/jan)
ggplot(delta_data, aes(x = reorder(line, -percent_change), y = percent_change)) + geom_bar(stat = "iden
    x = "Subway Line",
    y = "Percent Change",
    title = "Percent Change of Daily Ridership" # Chart title
)
```

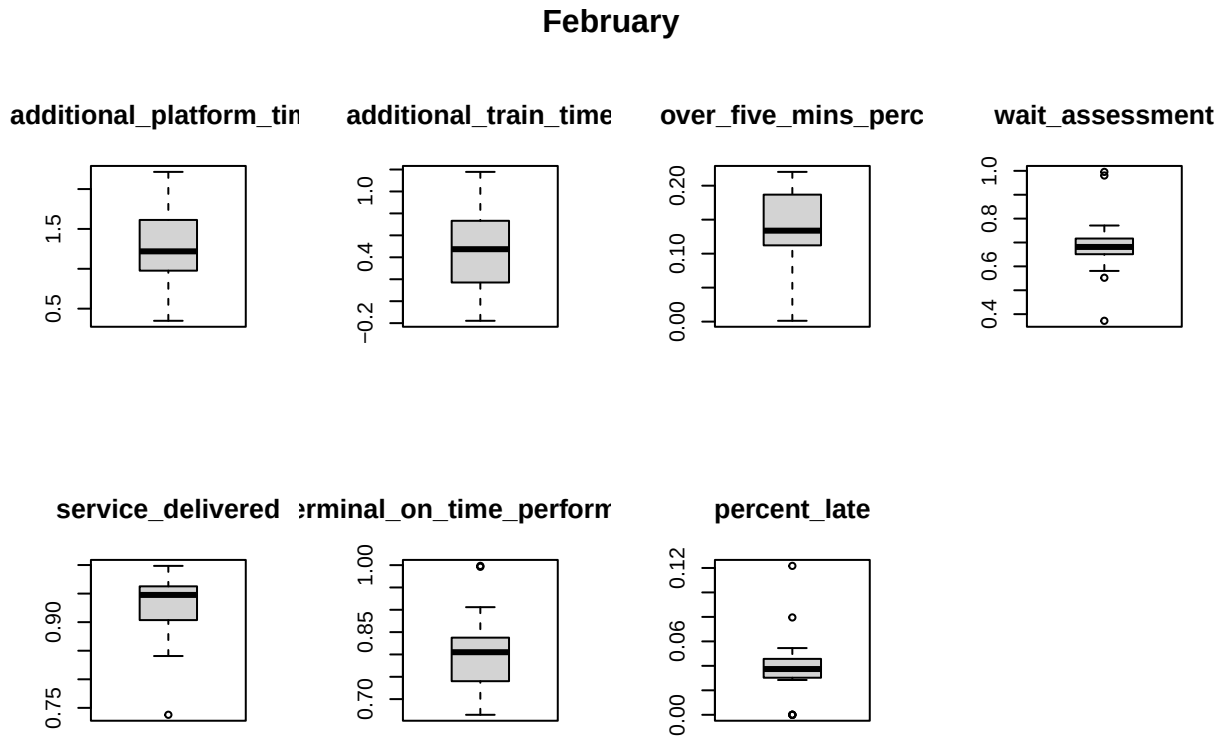


Customer Journey Metrics and Percent-based Performance Metrics

```
par(mfrow = c(2,4), oma = c(0, 0, 3, 0))
for (metric in colnames(line_data_jan)[4:10]){
  boxplot(line_data_jan[, metric], main = metric)
}
mtext("January", side = 3, line = 1, outer = TRUE, cex = 1, font = 2)

plot.new()
```



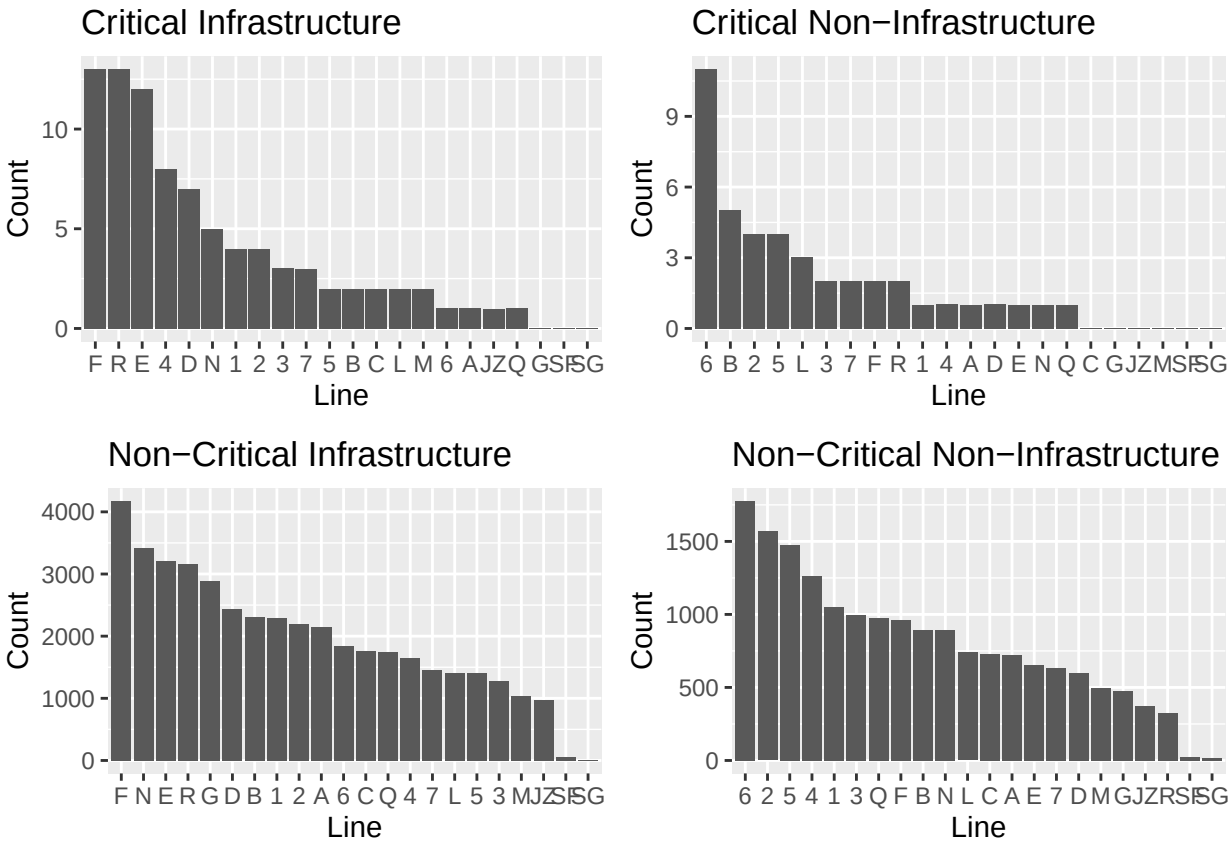

```
par(mfrow = c(2,4), oma = c(0, 0, 3, 0))
for (metric in colnames(line_data_jan)[4:10]){
  boxplot(line_data_feb[, metric], main = metric)
}
mtext("February", side = 3, line = 1, outer = TRUE, cex = 1, font = 2)
```



Incident Count by Line

```
# Create your individual ggplot barplots
plot1 <- ggplot(line_data, aes(x = reorder(line, -infra_critical), y = infra_critical)) +
  geom_bar(stat = "identity") +
  labs(
    x = "Line",
    y = "Count",
    title = "Critical Infrastructure"
  )

plot2 <- ggplot(line_data, aes(x = reorder(line, -noninfra_critical), y = noninfra_critical)) +
  geom_bar(stat = "identity") +
  labs(
    x = "Line",
    y = "Count",
    title = "Critical Non-Infrastructure"
  )

plot3 <- ggplot(subset(line_data, infra_noncritical > 0), aes(x = reorder(line, -infra_noncritical), y =
  geom_bar(stat = "identity") +
  labs(
    x = "Line",
    y = "Count",
```

```
    title = "Non-Critical Infrastructure"
  )

plot4 <- ggplot(subset(line_data, noninfra_noncritical > 0), aes(x = reorder(line, -noninfra_noncritical
  geom_bar(stat = "identity") +
  labs(
    x = "Line",
    y = "Count",
    title = "Non-Critical Non-Infrastructure"
  )

# Arrange the plots in a 2x2 grid
grid.arrange(plot1, plot2, plot3, plot4, nrow = 2, ncol = 2)
```



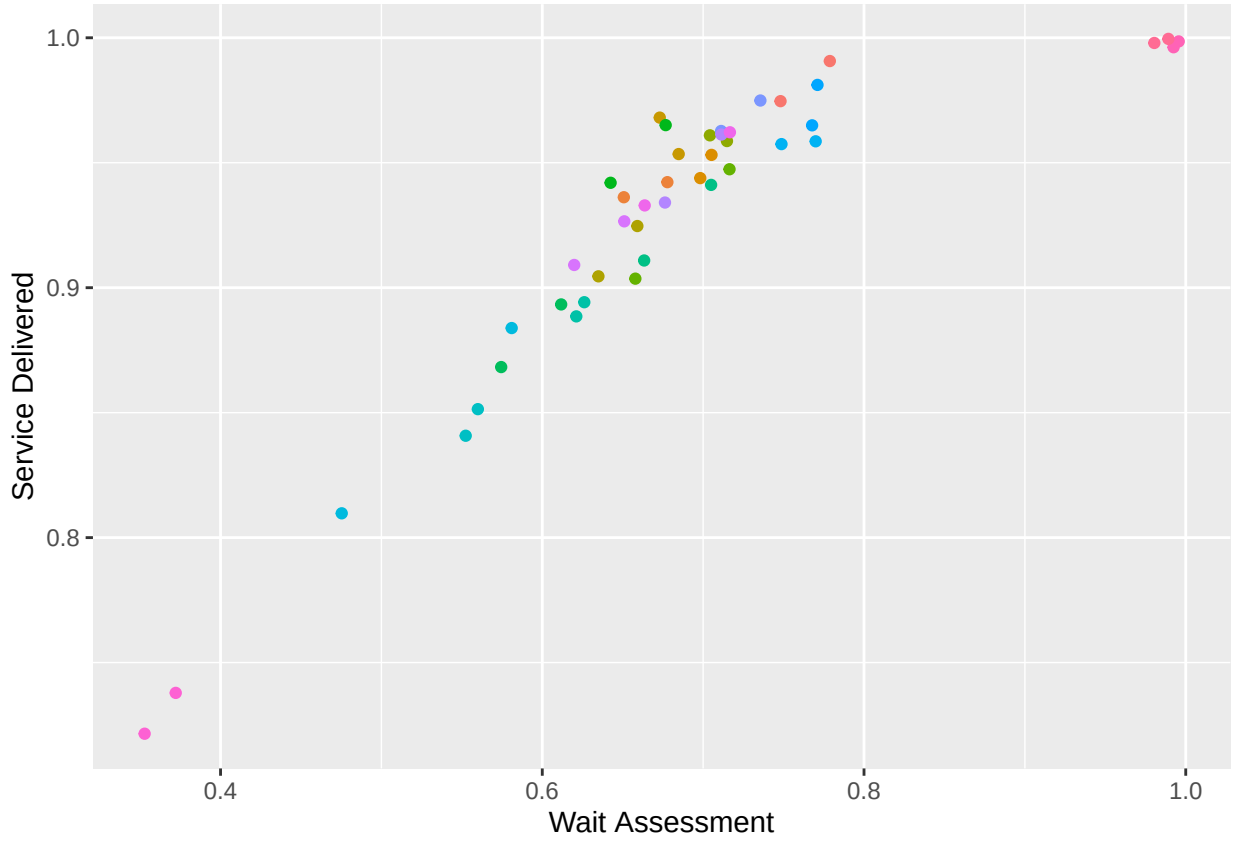
Some Correlation Examination

```
# row/cols in order: additional_platform_time, additional_train_time, over_five_mins_perc, wait_assessm
unname(cor(line_data[, colnames(line_data_jan)[4:10]]))
```

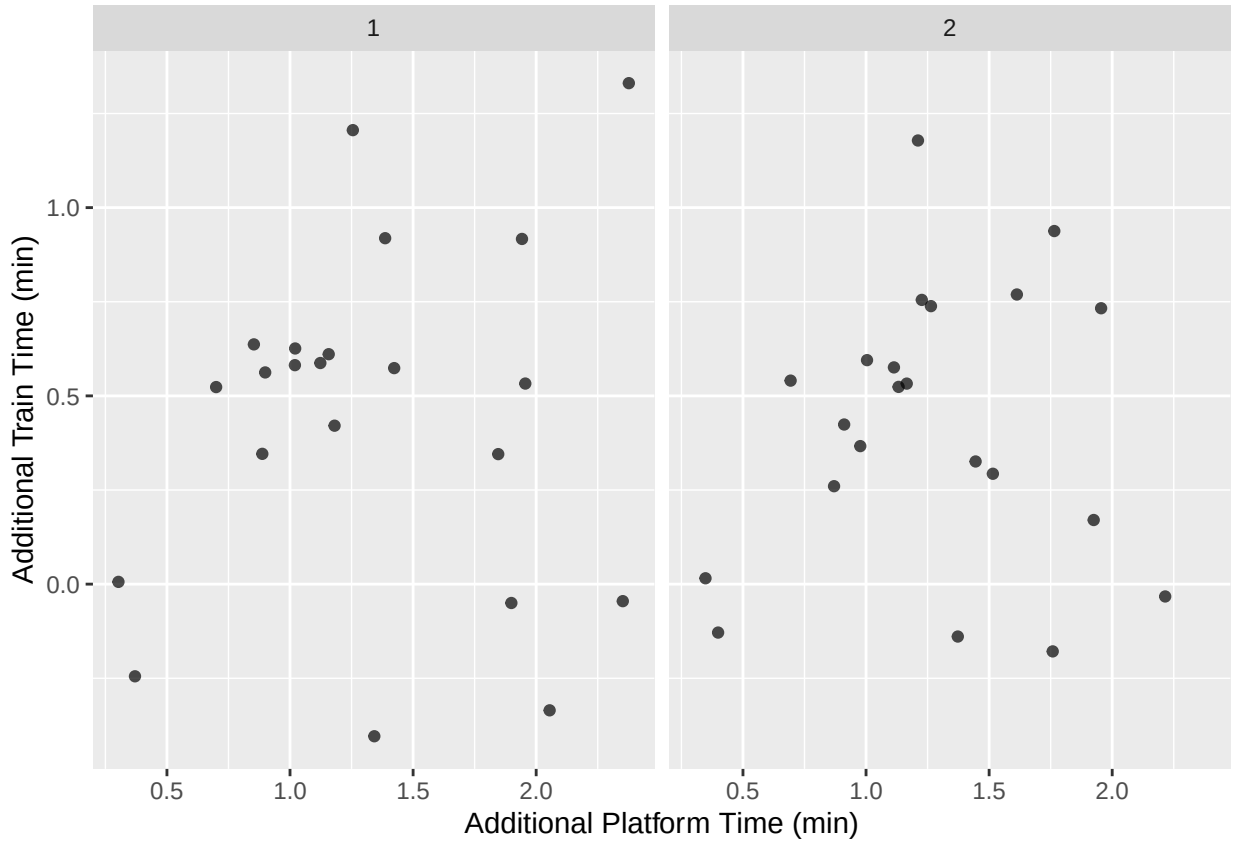
```
##            [,1]         [,2]       [,3]         [,4]       [,5]       [,6]
## [1,]  1.0000000  0.102316691  0.8423074 -0.781189081 -0.7250257 -0.6685811
## [2,]  0.1023167  1.000000000  0.3718904  0.007375179  0.2322656 -0.2992953
## [3,]  0.8423074  0.371890373  1.0000000 -0.696123290 -0.5661341 -0.7423563
```

```
## [4,] -0.7811891  0.007375179 -0.6961233  1.000000000  0.8825299  0.7610935
## [5,] -0.7250257  0.232265572 -0.5661341  0.882529899  1.0000000  0.6294668
## [6,] -0.6685811 -0.299295345 -0.7423563  0.761093532  0.6294668  1.0000000
## [7,]  0.2632206  0.594262004  0.4637880 -0.247434307 -0.1144114 -0.5479765
##             [,7]
## [1,]  0.2632206
## [2,]  0.5942620
## [3,]  0.4637880
## [4,] -0.2474343
## [5,] -0.1144114
## [6,] -0.5479765
## [7,]  1.0000000
```

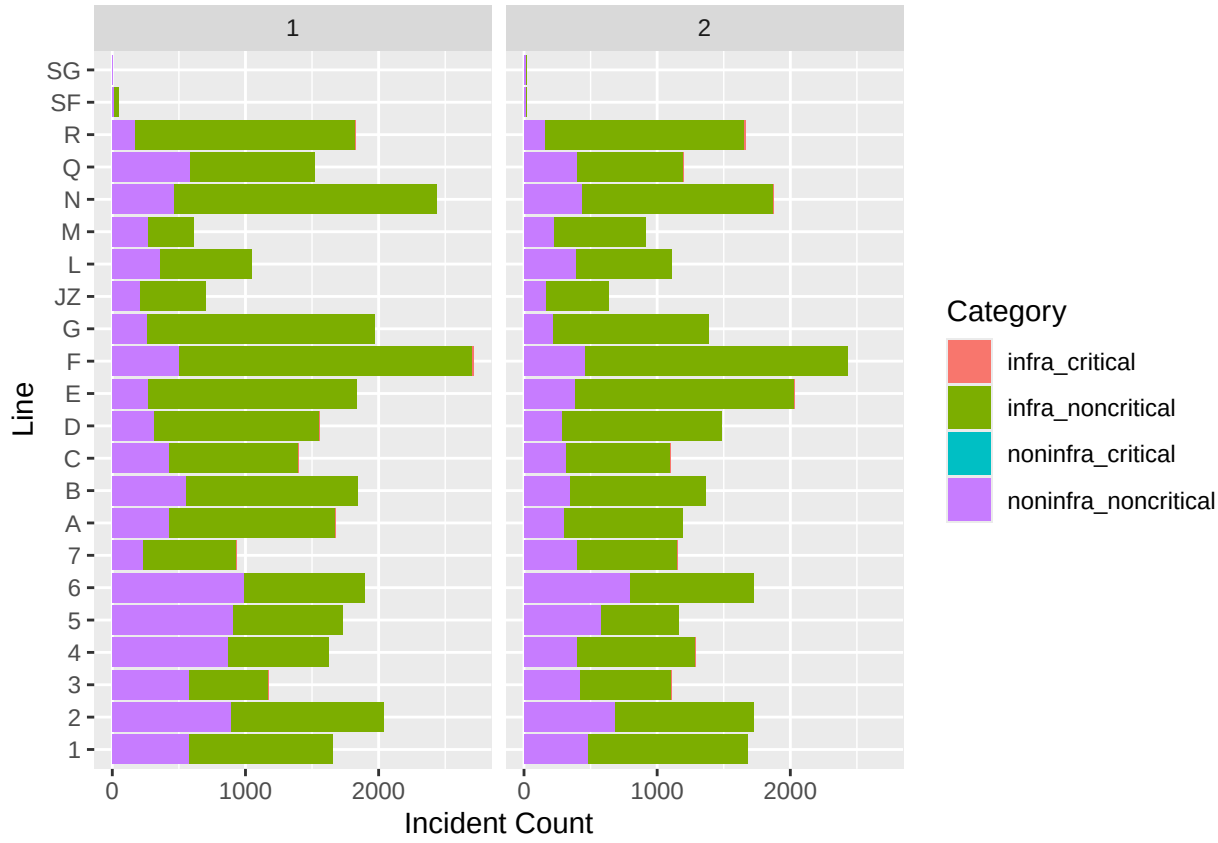
```
ggplot(line_data, aes(x = wait_assessment, y = service_delivered, color = line)) +
  geom_point() +
  theme(legend.position = "none") +
labs(x = "Wait Assessment", y = "Service Delivered")
```



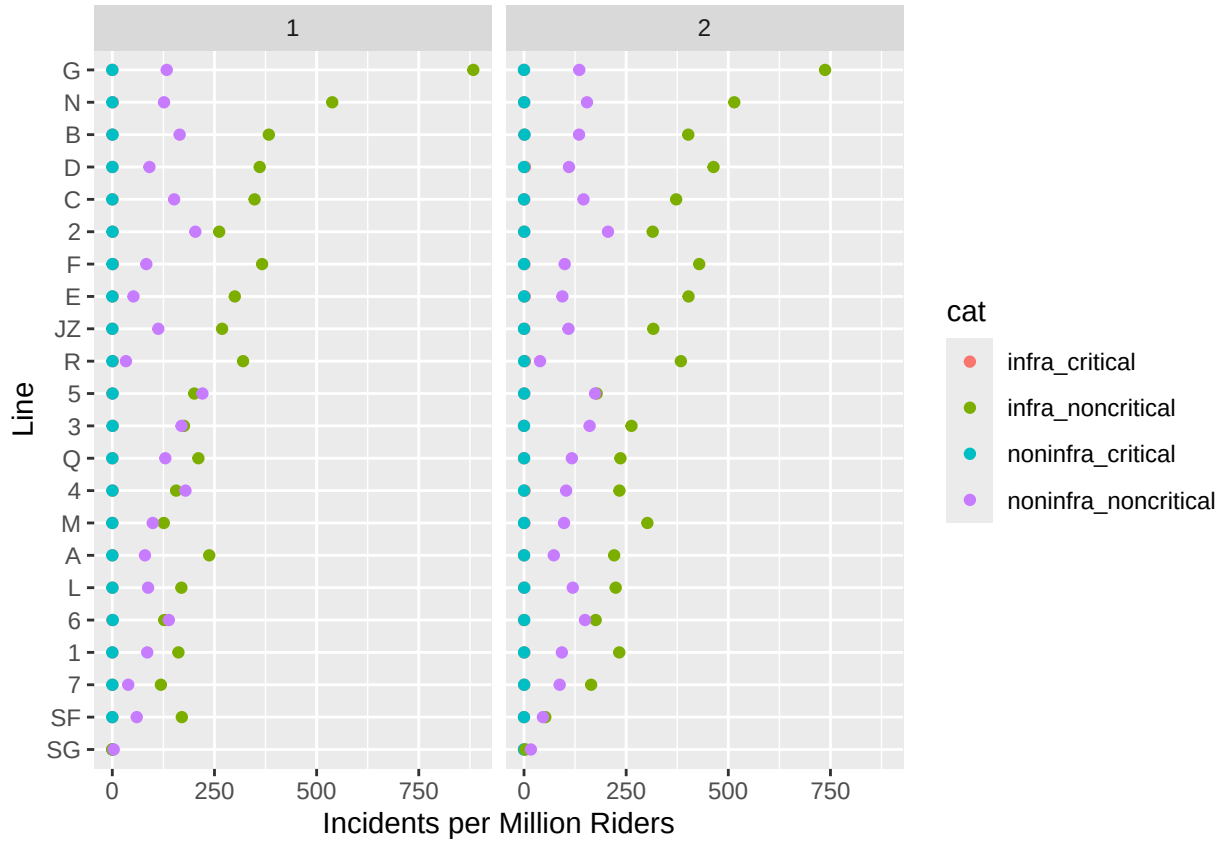

```
ggplot(line_data, aes(x = additional_platform_time, y = additional_train_time)) +
  geom_point(alpha = .7) +
  facet_wrap(~ month) +
  labs(x = "Additional Platform Time (min)",
       y = "Additional Train Time (min)")
```




```r
df_long_inc <- line_data %>%
  select(line, month, infra_critical:noninfra_noncritical) %>%
  pivot_longer(cols = infra_critical:noninfra_noncritical,
               names_to = "category", values_to = "count")

df_long_inc %>%
  ggplot(aes(x = line, y = count, fill = category)) +
  geom_col(position = "stack") +
  facet_wrap(~ month) +
  coord_flip() +
  labs(x = "Line", y = "Incident Count", fill = "Category")
```




```r
# 4b) Incidents per million riders
df_inc_rate <- line_data %>%
  mutate_at(vars(infra_critical:noninfra_noncritical),
            ~ . / num_passengers * 1e6)

df_inc_rate %>%
  select(line, month, infra_critical:noninfra_noncritical) %>%
  pivot_longer(cols = infra_critical:noninfra_noncritical,
               names_to = "cat", values_to = "rate") %>%
  ggplot(aes(x = rate, y = reorder(line, rate), color = cat)) +
  geom_point() +
  facet_wrap(~ month) +
  labs(x = "Incidents per Million Riders", y = "Line")
```



PCA

```
library(FactoMineR)    # for PCA
```

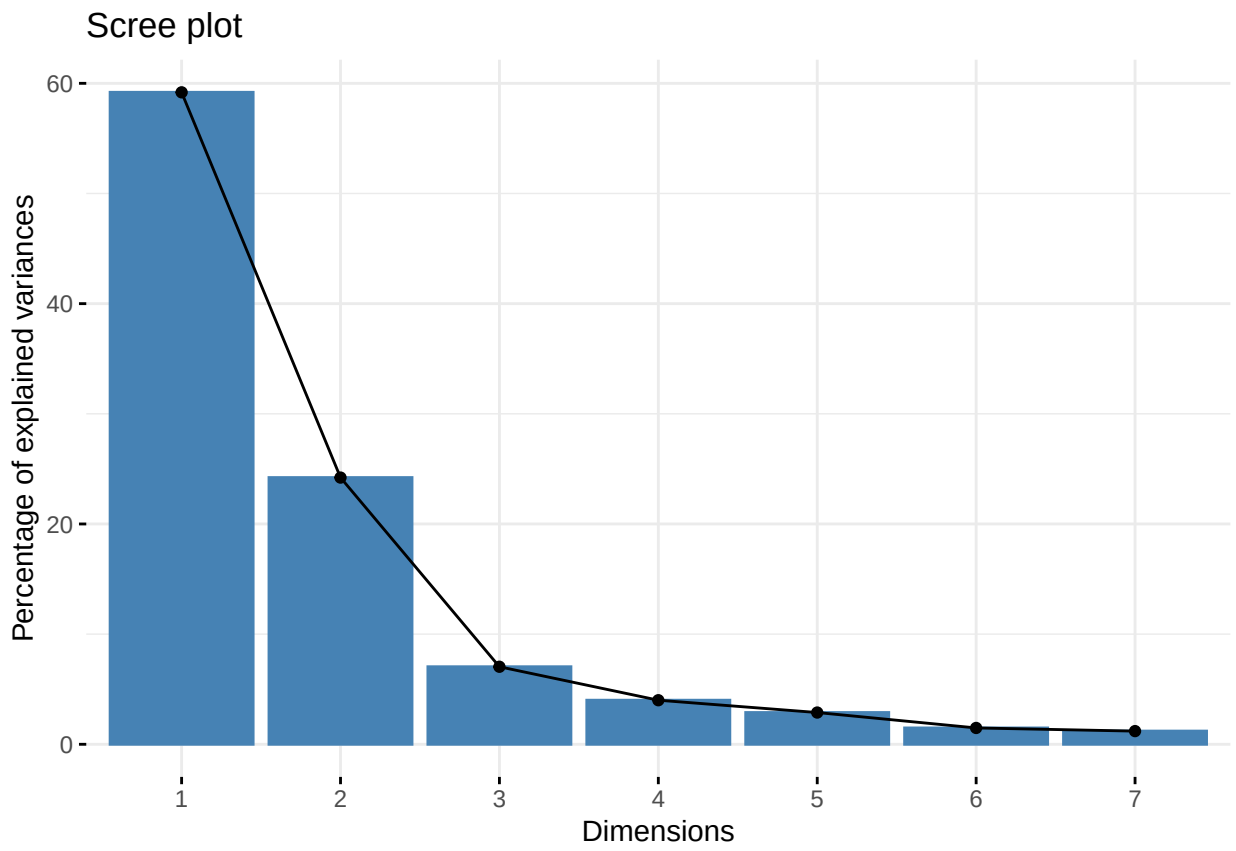
```
## Warning: package 'FactoMineR' was built under R version 4.4.3
```

```
library(factoextra)    # for PCA visualization
```

```
## Warning: package 'factoextra' was built under R version 4.4.3
```

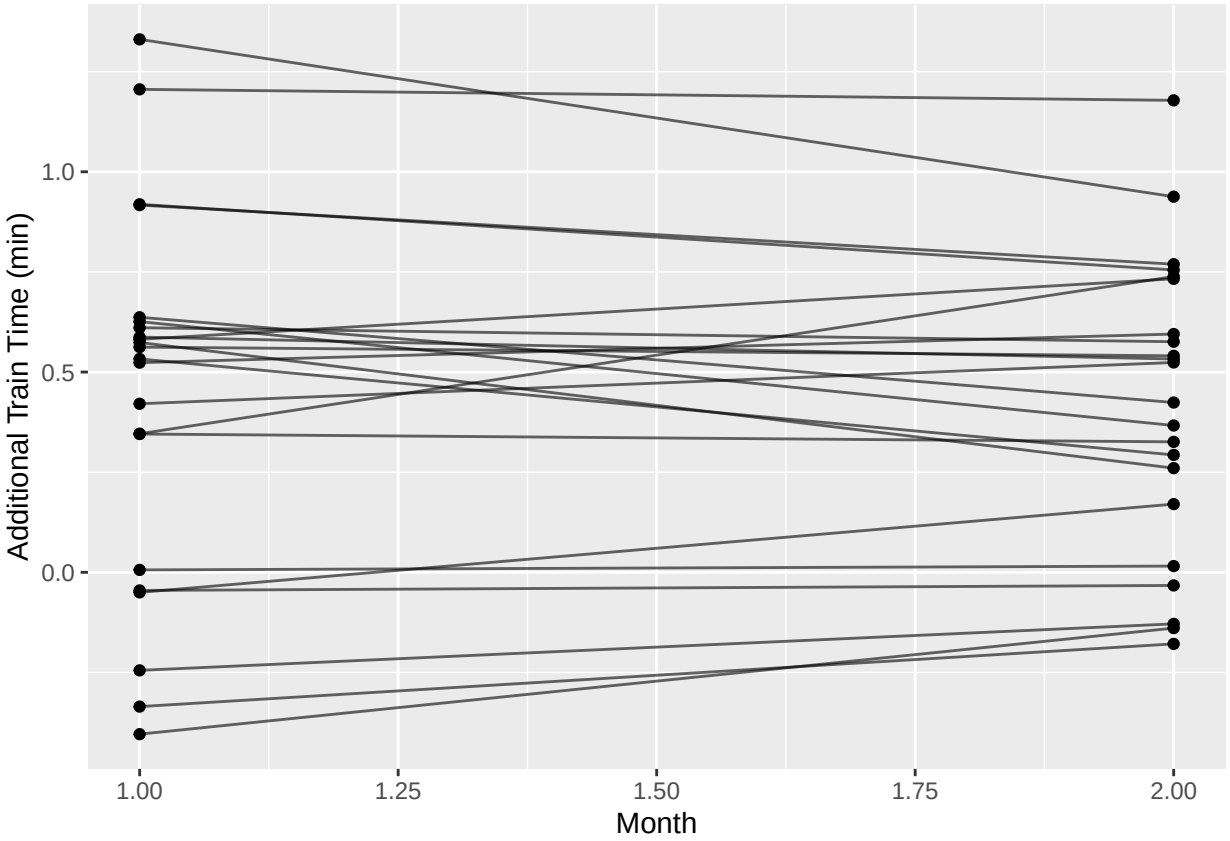
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
cont.vars = line_data[, colnames(line_data_jan)[4:10]]
res.pca = PCA(cont.vars, graph = FALSE)
fviz_eig(res.pca)
```




```
#fviz_pca_ind(res.pca,
              #geom.ind = "point",
              #habillage = line_data$line,
              #repel = TRUE) +
  #labs(title = "PCA: Lines in PC space")
```

```
line_data %>%
  filter(line %in% unique(line_data$line)) %>%
  select(line, month, additional_train_time) %>%
  ggplot(aes(x = month, y = additional_train_time, group = line)) +
  geom_line(alpha = .6) +
  geom_point() +
  labs(x = "Month", y = "Additional Train Time (min)")
```



Station Ridership Data EDA

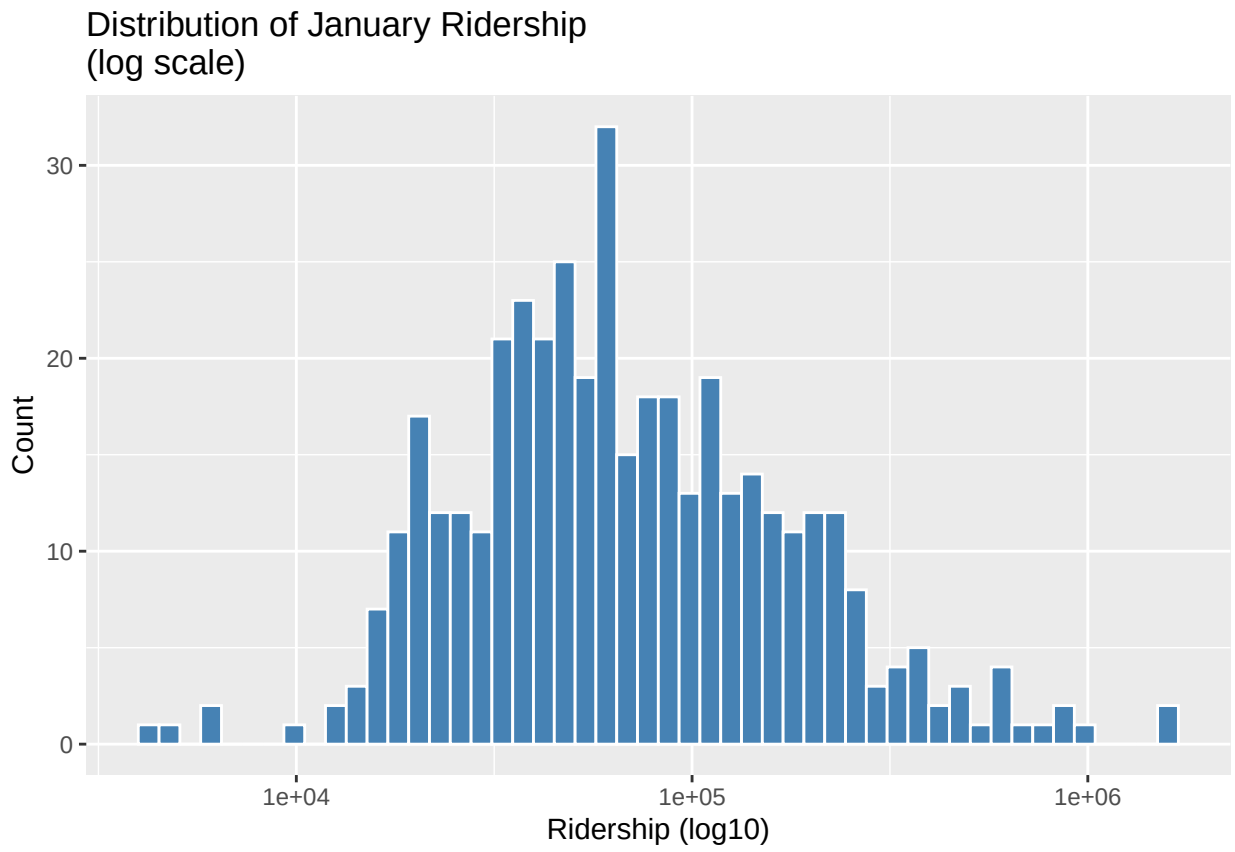
```
ridership_data = read.csv("C:\\Users\\hychu\\OneDrive\\Desktop\\SP25\\_BTRY_4100\\_FINAL_PROJECT\\MTA-N
  mutate(
    lines_count = str_count(lines, ",") + 1
  )
ridership_data_jan = ridership_data[ridership_data$month == 1, ]
ridership_data_feb = ridership_data[ridership_data$month == 2, ]

ridership_data_jan = ridership_data_jan[ridership_data_jan$station_complex %in% intersect(
  ridership_data_jan$station_complex,
  ridership_data_feb$station_complex
), ]
head(ridership_data)
```

```
##   station_complex month   borough ridership lines lines_count
## 1        1 Av (L)     1 Manhattan    256485     L           1
## 2        1 Av (L)     2 Manhattan    238896     L           1
## 3      103 St (1)     1 Manhattan    113094     1           1
## 4      103 St (1)     2 Manhattan    106107     1           1
## 5      103 St (6)     1 Manhattan    110822     6           1
## 6      103 St (6)     2 Manhattan    103738     6           1
```

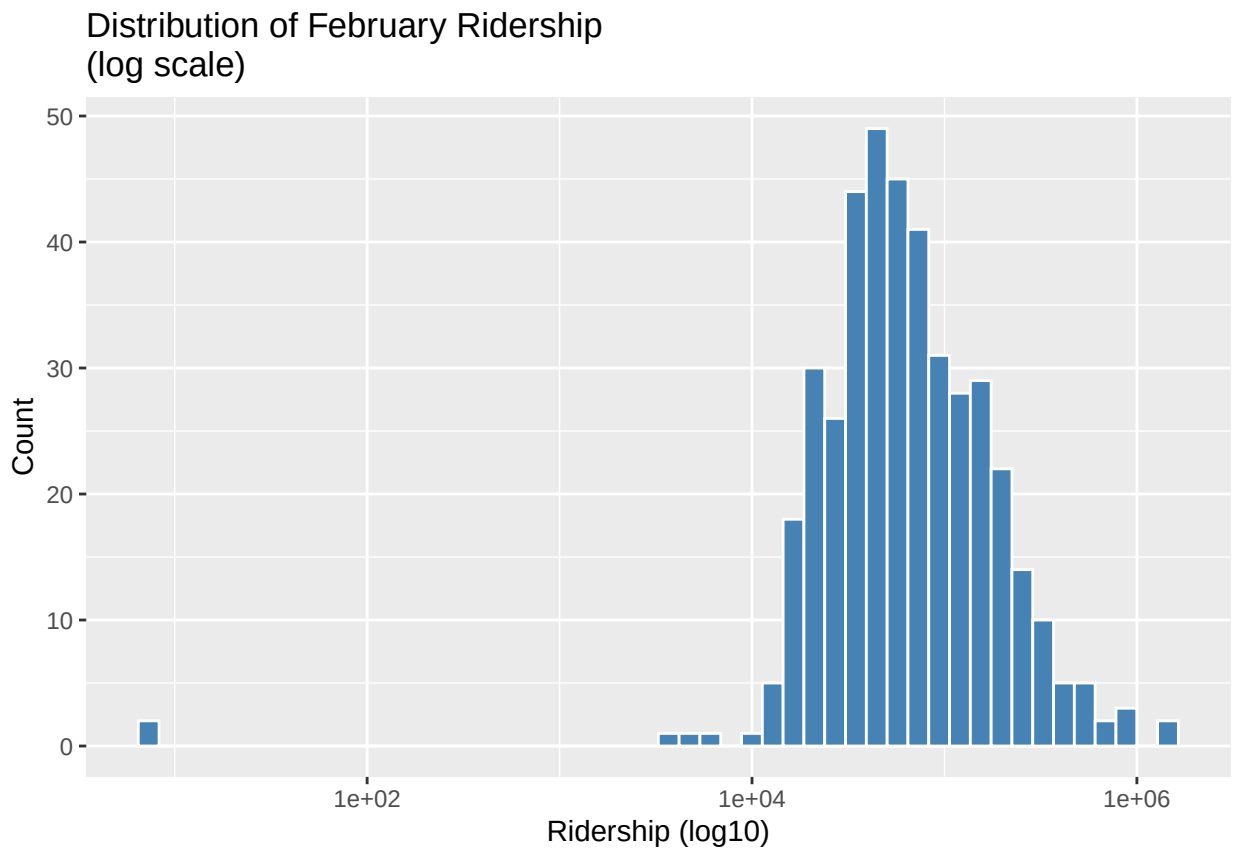
```
stations_jan = unique(ridership_data_jan$station_complex)
stations_feb = unique(ridership_data_feb$station_complex)
setdiff(stations_jan, stations_feb) # to be removed
```

```
## character(0)
```

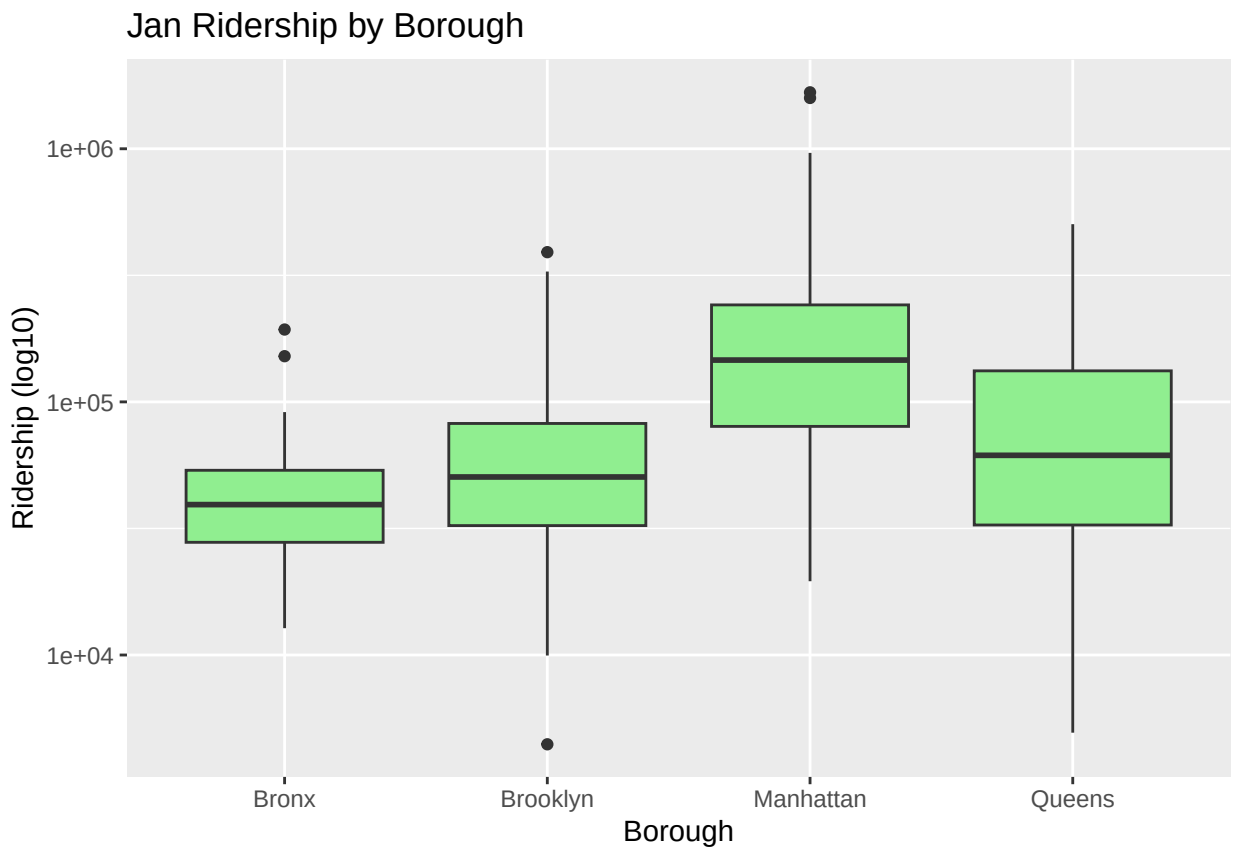
```
# 1a) Histogram (log-scale) of ridership
ggplot(ridership_data_jan, aes(x = ridership)) +
  geom_histogram(bins = 50, fill = "steelblue", color = "white") +
  scale_x_log10() +
  labs(title = "Distribution of January Ridership\n(log scale)",
       x = "Ridership (log10)", y = "Count")
```



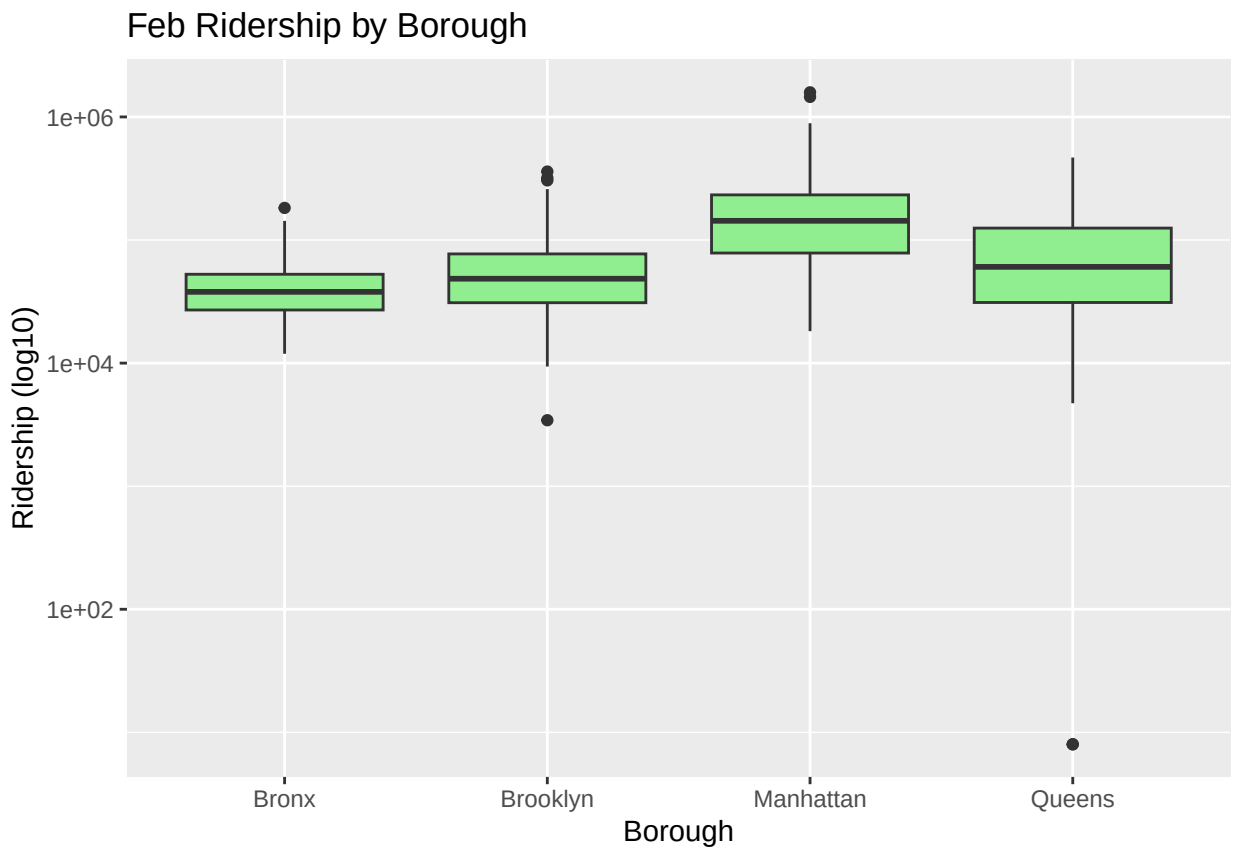

```
ggplot(ridership_data_feb, aes(x = ridership)) +
  geom_histogram(bins = 50, fill = "steelblue", color = "white") +
  scale_x_log10() +
  labs(title = "Distribution of February Ridership\n(log scale)",
       x = "Ridership (log10)", y = "Count")
```



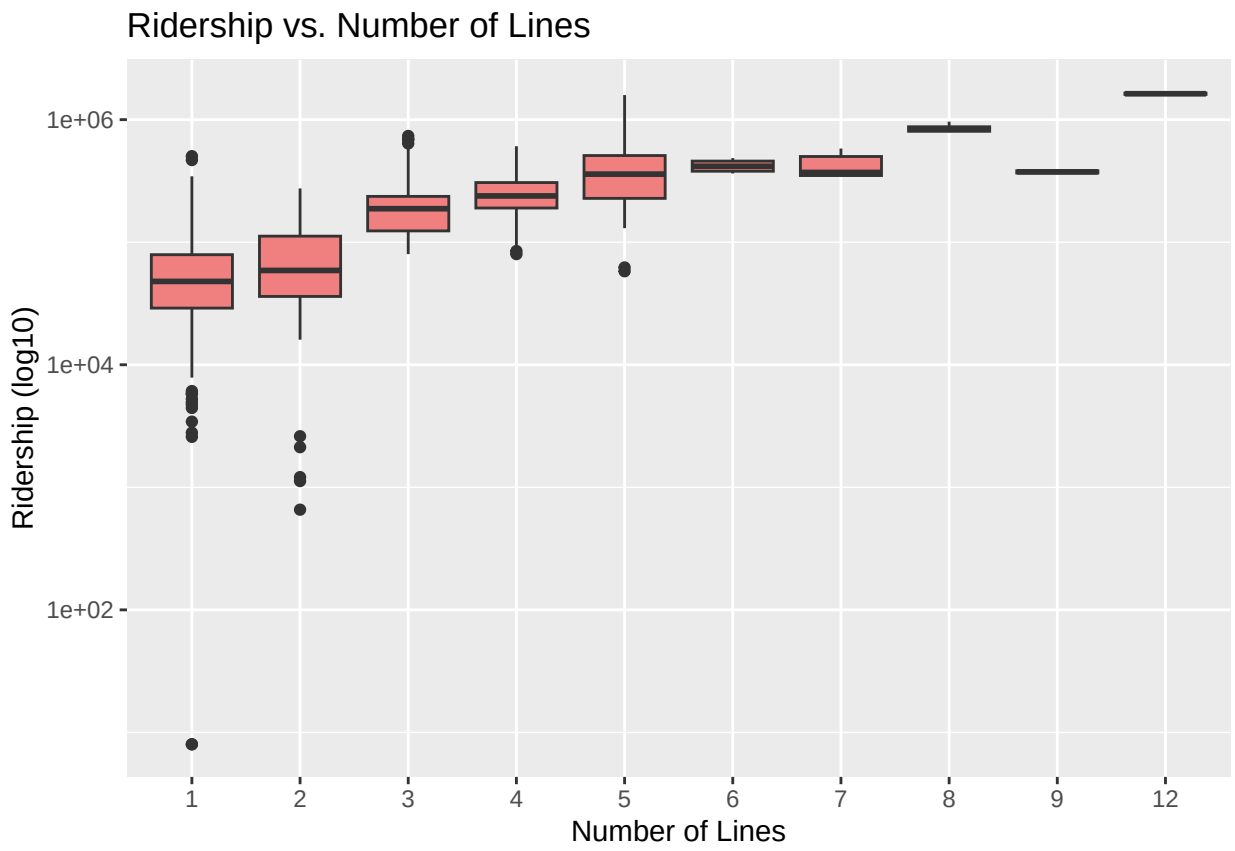

```r
ggplot(ridership_data_jan, aes(x = borough, y = ridership)) +
  geom_boxplot(fill = "lightgreen") +
  scale_y_log10() +
  labs(title = "Jan Ridership by Borough", x = "Borough", y = "Ridership (log10)")
```



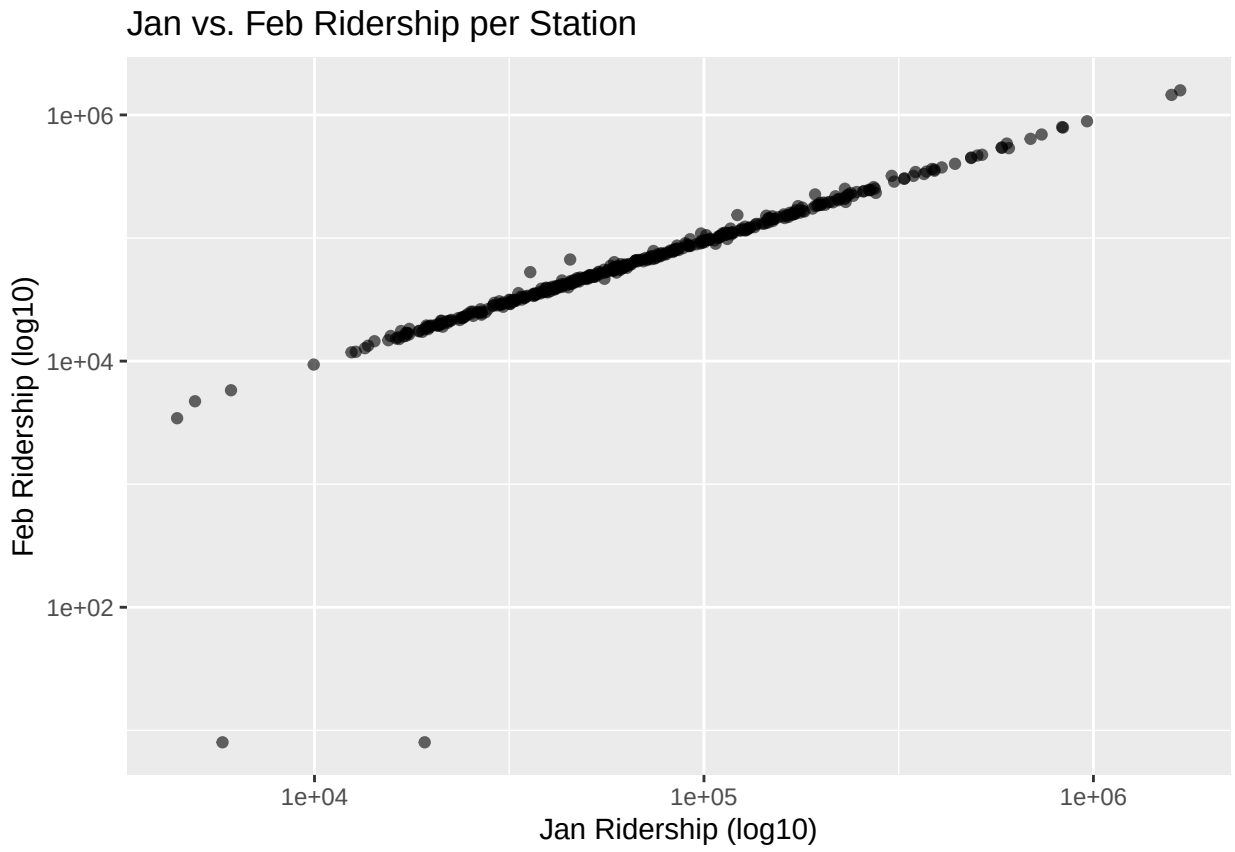

```
ggplot(ridership_data_feb, aes(x = borough, y = ridership)) +
  geom_boxplot(fill = "lightgreen") +
  scale_y_log10() +
  labs(title = "Feb Ridership by Borough", x = "Borough", y = "Ridership (log10)")
```



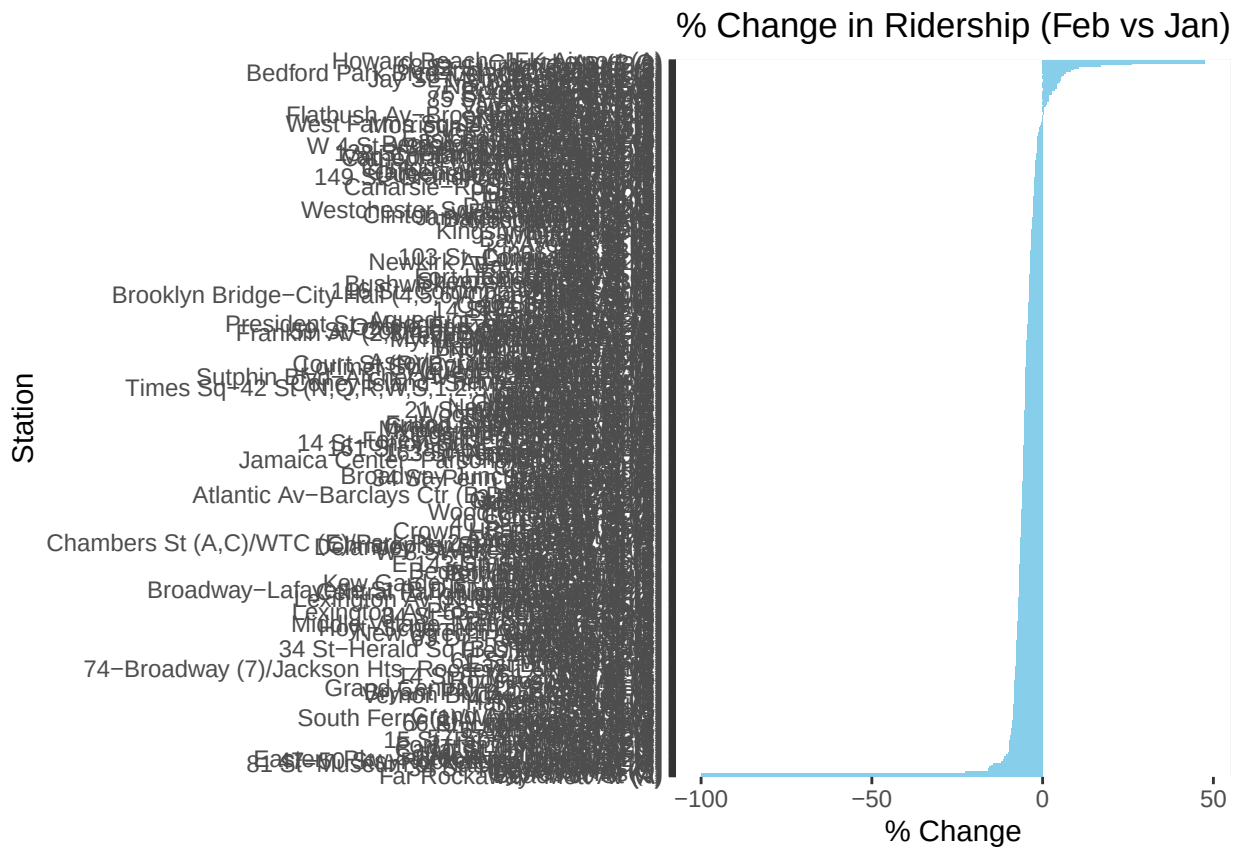

```r
# Boxplot of ridership by how many lines serve the station
ggplot(ridership_data, aes(x = factor(lines_count), y = ridership)) +
  geom_boxplot(fill = "lightcoral") +
  scale_y_log10() +
  labs(title = "Ridership vs. Number of Lines",
       x = "Number of Lines", y = "Ridership (log10)")
```



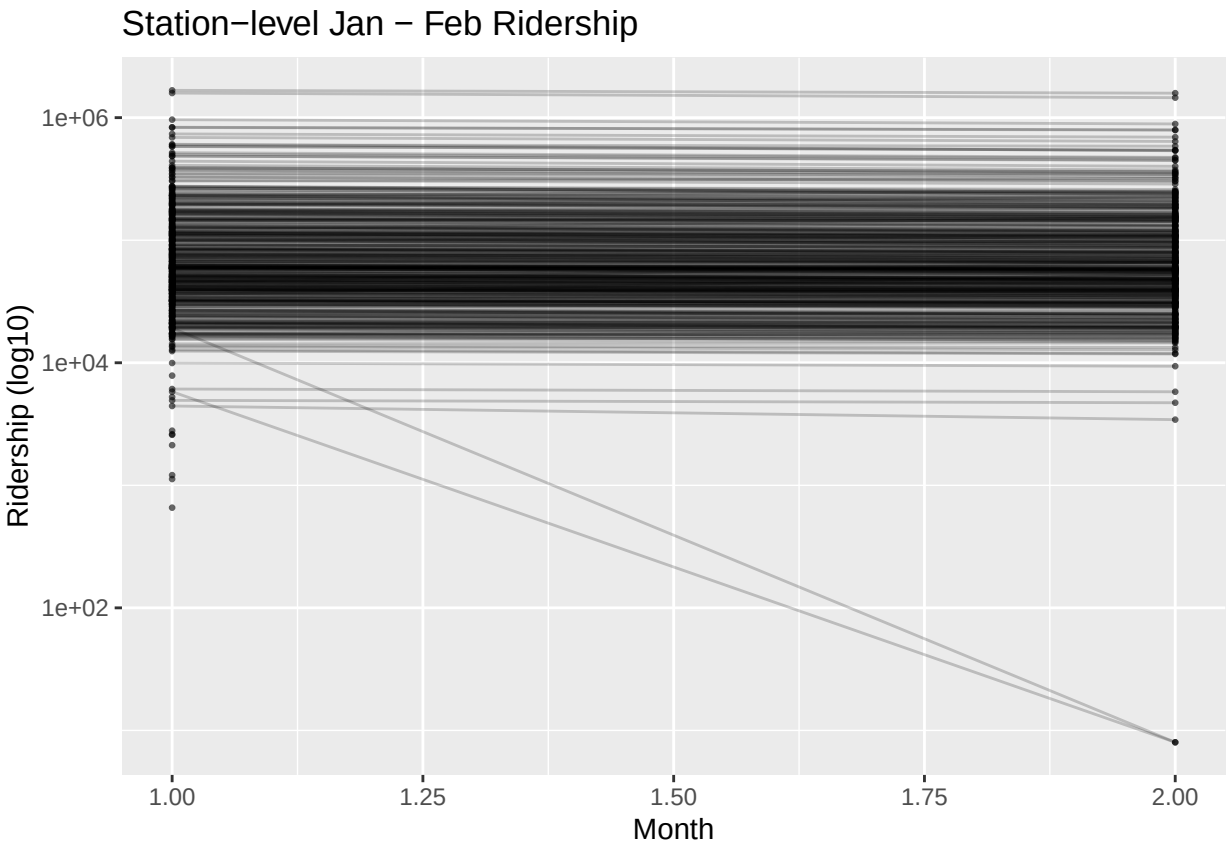

```
df_diff = data.frame(
  station_complex = ridership_data_feb$station_complex,
  jan = ridership_data_jan[order(ridership_data_jan$station_complex), ]$ridership,
  feb = ridership_data_feb[order(ridership_data_feb$station_complex), ]$ridership
)

ggplot(df_diff, aes(x = jan, y = feb)) +
  geom_point(alpha = 0.6) +
  #geom_smooth(method = "lm", se = FALSE, color = "darkred") +
  scale_x_log10() + scale_y_log10() +
  labs(title = "Jan vs. Feb Ridership per Station",
       x = "Jan Ridership (log10)", y = "Feb Ridership (log10)")
```




```r
# MONTH-TO-MONTH CHANGE
df_delta <- df_diff %>%
  mutate(
    pct_change = (feb - jan) / jan * 100
  )

# Bar chart of % change
ggplot(df_delta, aes(x = reorder(station_complex, pct_change), y = pct_change)) +
  geom_col(fill = "skyblue") +
  coord_flip() +
  labs(title = "% Change in Ridership (Feb vs Jan)",
       x = "Station", y = "% Change")
```




```
# PAIRED-POINT PLOT
# show how each station moved from Jan→Feb
ggplot(ridership_data, aes(x = month, y = ridership, group = station_complex)) +
  geom_line(alpha = 0.2) +
  geom_point(alpha = 0.6, size = 0.5) +
  scale_y_log10() +
  labs(title = "Station-level Jan - Feb Ridership",
       x = "Month", y = "Ridership (log10)")
```


Station−level Jan − Feb Ridership
Ridership (log10)
1e+06
1e+04
1e+02
1.00
1.25
1.50
1.75
2.00
Month