Visual Story Telling Part 2: Capital Metro Data

Sarah Dominguez

2024-08-13

Visual story telling part 2: Capital Metro data

The file capmetro_UT.csv contains data from Austin's own Capital Metro bus network, including shuttles to, from, and around the UT campus. These data track ridership on buses in the UT area. Ridership is measured by an optical scanner that counts how many people embark and alight the bus at each stop. Each row in the data set corresponds to a 15-minute period between the hours of 6 AM and 10 PM, each and every day, from September through November 2018. The variables are:

- *timestamp*: the beginning of the 15-minute window for that row of data
- *boarding*: how many people got on board any Capital Metro bus on the UT campus in the specific 15 minute window
- *alighting*: how many people got off ("alit") any Capital Metro bus on the UT campus in the specific 15 minute window
- *day_of_week* and *weekend*: Monday, Tuesday, etc, as well as an indicator for whether it's a weekend.
- *temperature*: temperature at that time in degrees F
- *hour_of_day*: on 24-hour time, so 6 for 6 AM, 13 for 1 PM, 14 for 2 PM, etc.
- *month*: July through December

Your task is to create a figure, or set of related figures, that tell an interesting story about Capital Metro ridership patterns around the UT-Austin campus during the semester in question. Provide a clear annotation/caption for each figure, but the figure(s) should be more or less stand-alone, in that you shouldn't need many, many paragraphs to convey its meaning. Rather, the figure together with a concise caption should speak for itself as far as possible.

You have broad freedom to look at any variables you'd like here – try to find that sweet spot where you're showing genuinely interesting relationships among more than just two variables, but where the resulting figure or set of figures doesn't become overwhelming/confusing. (Faceting/panel plots might be especially useful here.)

```
library(ggplot2)
library(dplyr)
```

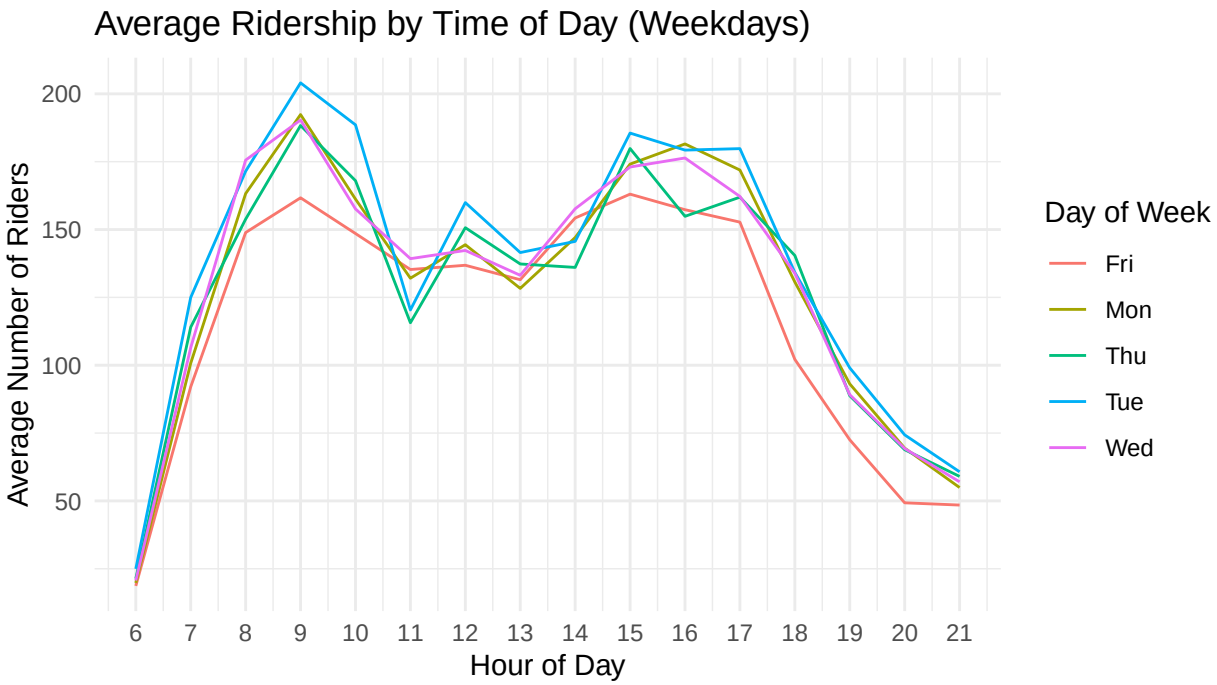
```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)

# Load the data
capmetro <- read.csv("capmetro_UT.csv")

# View the first few rows of the data
head(capmetro)
```

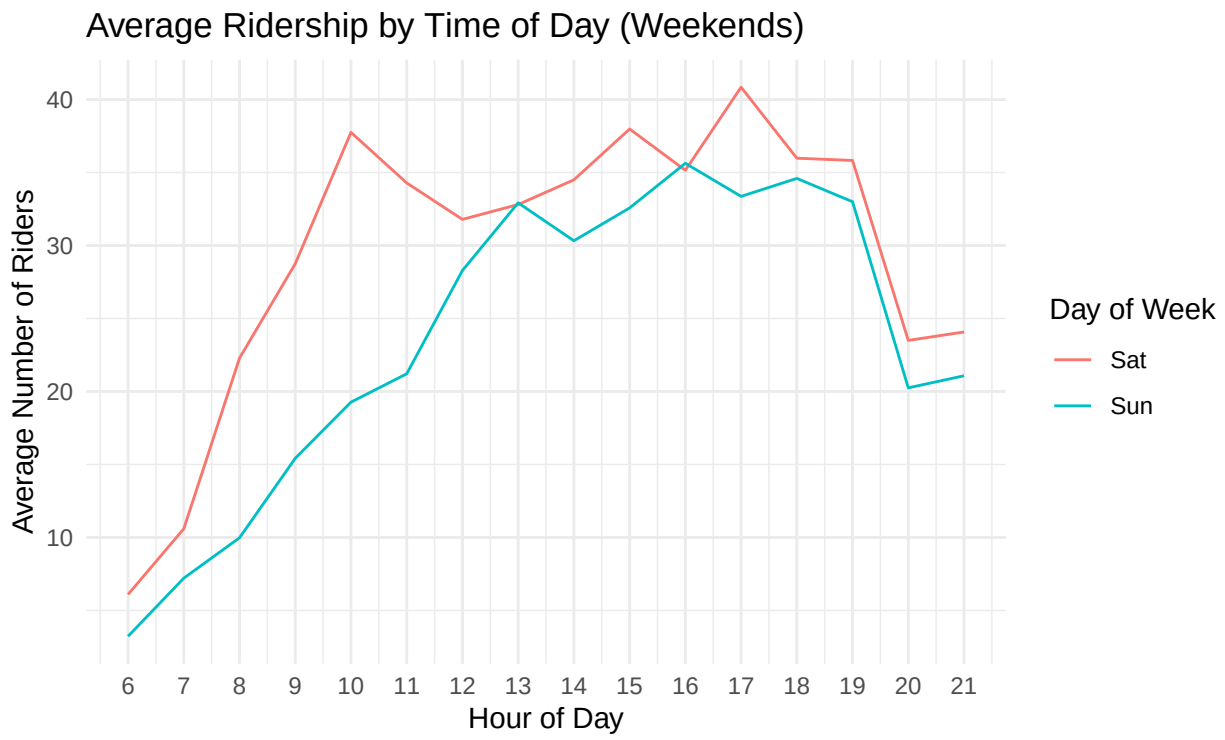
```
##             timestamp boarding alighting day_of_week temperature hour_of_day
## 1 2018-09-01 06:00:00        0         1         Sat       74.82           6
## 2 2018-09-01 06:15:00        2         1         Sat       74.82           6
## 3 2018-09-01 06:30:00        3         4         Sat       74.82           6
## 4 2018-09-01 06:45:00        3         4         Sat       74.82           6
## 5 2018-09-01 07:00:00        2         4         Sat       74.39           7
## 6 2018-09-01 07:15:00        4         4         Sat       74.39           7
##   month weekend
## 1   Sep weekend
## 2   Sep weekend
## 3   Sep weekend
## 4   Sep weekend
## 5   Sep weekend
## 6   Sep weekend
```

```
# Splitting the data into weekday and weekend
weekday_data <- capmetro %>% filter(weekend == "weekday")
weekend_data <- capmetro %>% filter(weekend == "weekend")


ggplot(weekday_data, aes(x = hour_of_day, y = boarding + alighting, color = day_of_week)) +
  geom_line(stat = "summary", fun = "mean") +
  scale_x_continuous(breaks = seq(6, 22, by = 1)) +  # Display all hours
  labs(title = "Average Ridership by Time of Day (Weekdays)",
       x = "Hour of Day",
       y = "Average Number of Riders",
       color = "Day of Week",
       caption = "This chart illustrates the ridership patterns for Capital Metro buses from Monday to
  theme_minimal()
```



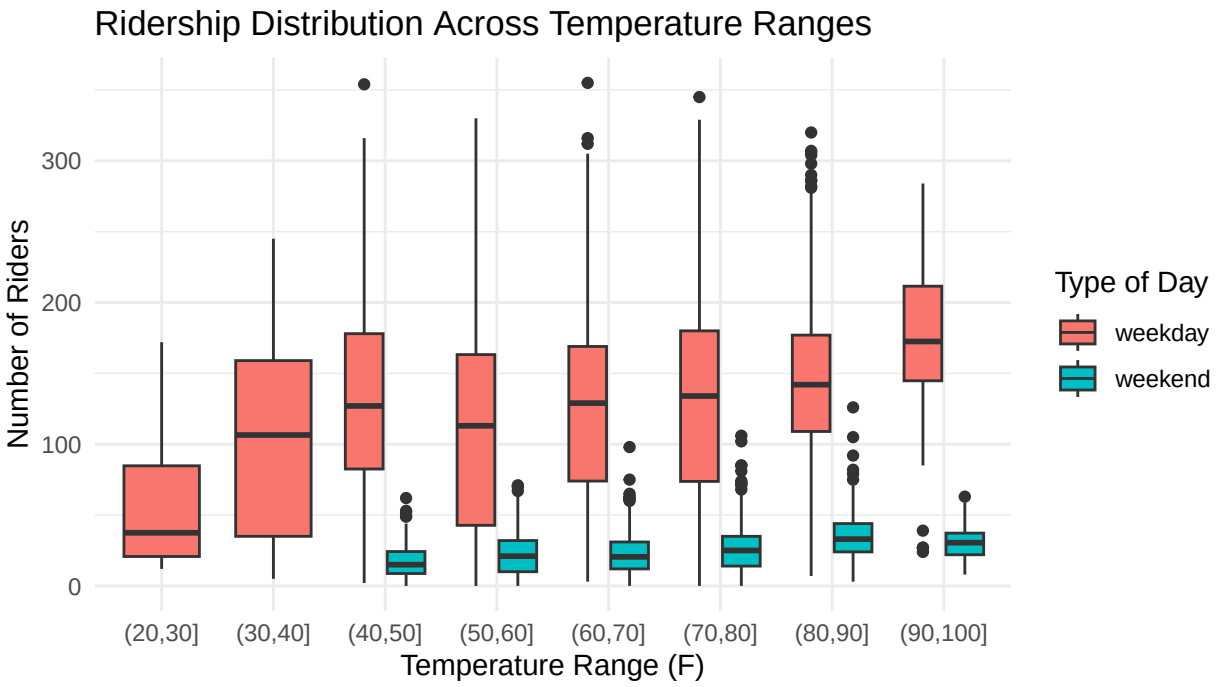
This chart illustrates the ridership patterns for Capital Metro buses from Monday to Friday, highlighting distinct morning and evening peaks corresponding to typical commuter rush hours. The morning peak is the most pronounced, occurring between 7 AM and 9 AM, while the evening peak occurs from 4 PM to 6 PM. There is a slight variation in ridership on Fridays, particularly in the evening, which could reflect different commuter behaviors as the weekend approaches.

```
ggplot(weekend_data, aes(x = hour_of_day, y = boarding + alighting, color = day_of_week)) +
  geom_line(stat = "summary", fun = "mean") +
  scale_x_continuous(breaks = seq(6, 22, by = 1)) +  # Display all hours
  labs(title = "Average Ridership by Time of Day (Weekends)",
       x = "Hour of Day",
       y = "Average Number of Riders",
       color = "Day of Week",
       caption = "This chart illustrates the weekend ridership patterns for Capital Metro\nbuses, showi
  theme_minimal()
```



This chart illustrates the weekend ridership patterns for Capital Metro buses, showing a gradual increase in the morning and a stable ridership pattern during the y hours. Saturday exhibits higher ridership, especially during the evening, compared to Sunday. ‹ ridership on both days occurs between 4 PM and 6 PM, after which ridership declines sharply.

```
capmetro_clean2 <- capmetro %>%
  mutate(temperature = cut(temperature, breaks = c(10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110)))

capmetro_clean2 <- capmetro_clean2 %>%
  filter(!is.na(temperature))

ggplot(capmetro_clean2, aes(x = temperature, y = boarding + alighting, fill = weekend)) +
  geom_boxplot() +
  labs(title = "Ridership Distribution Across Temperature Ranges",
       x = "Temperature Range (F)",
       y = "Number of Riders",
       fill = "Type of Day",
       caption = "This boxplot illustrates the ridership patterns for Capital Metro buses across differ
  theme_minimal()
```

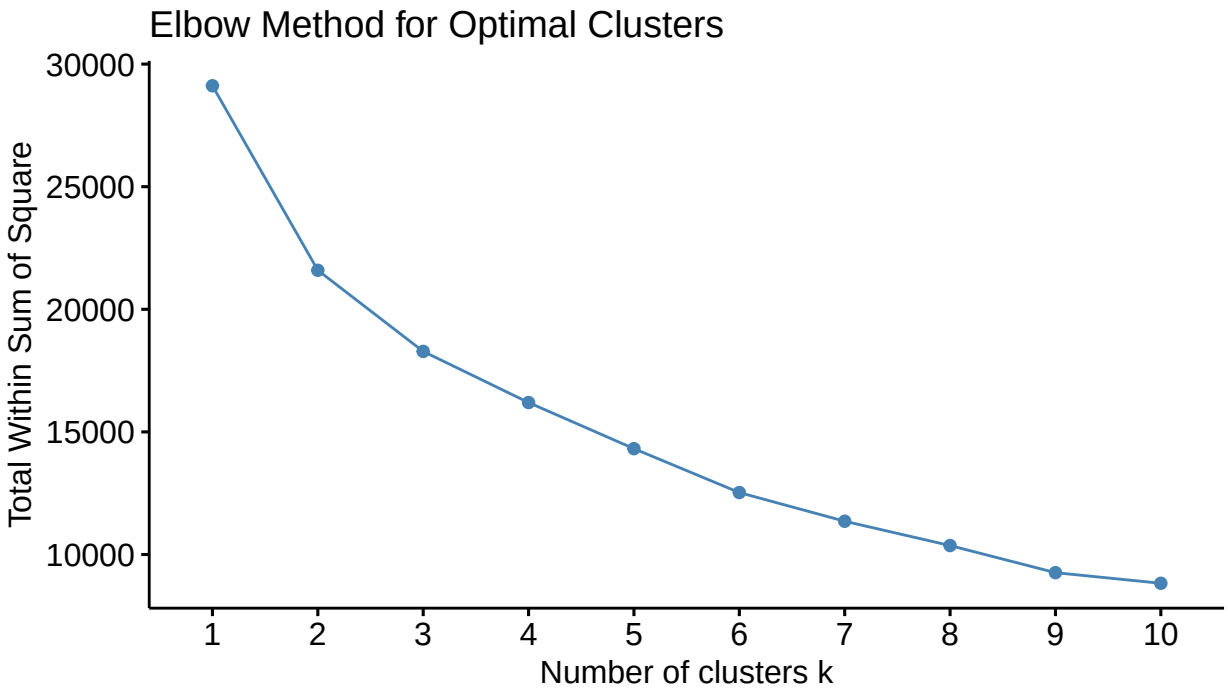


This boxplot illustrates the ridership patterns for Capital Metro buses across different temperature ranges on weekdays and weekends.Weekday ridership is consistently higher than weekend ridership across all temperatures. The difference is especially noticeable in moderate temperatures (50°F − 70°F). As temperatures rise, ridership increases slightly for both weekdays and weekends, but weekdays still see more riders. Some days have unusually high ridership, especially when it's warmer.

```
library(dplyr)
library(ggplot2)
library(cluster)
library(factoextra)
```

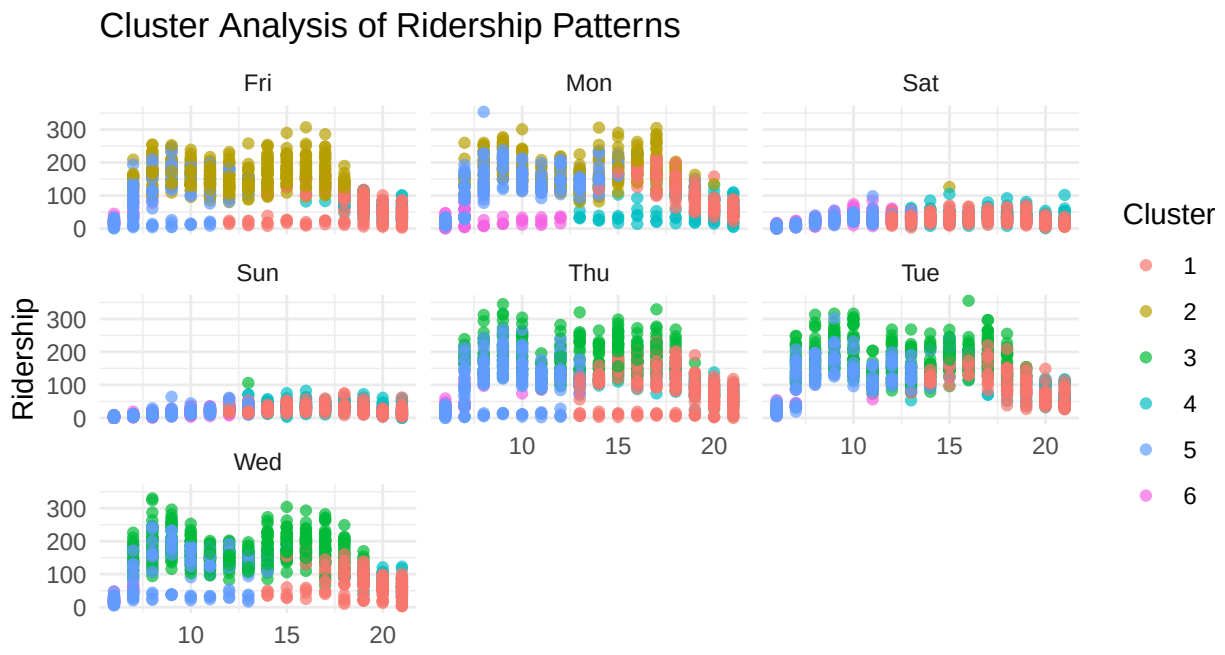
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
# Combine 'boarding' and 'alighting' into a single 'ridership' metric
capmetro <- capmetro %>%
  mutate(ridership = boarding + alighting)

# Select relevant variables and scale them for clustering
clustering_data <- capmetro %>%
  select(hour_of_day, day_of_week, temperature, month, ridership) %>%
  mutate(day_of_week = as.numeric(as.factor(day_of_week)),  # Convert categorical to numeric
         month = as.numeric(as.factor(month))) %>%  # Convert categorical to numeric
  scale()  # Standardize the data

# Elbow method to find the optimal number of clusters
fviz_nbclust(clustering_data, kmeans, method = "wss") +
  labs(title = "Elbow Method for Optimal Clusters",
       caption = "Choosing the Right Number of Clusters: This graph shows how the variation within clus
```



Choosing the Right Number of Clusters: This graph shows how the variation within clusters decreases as the number of clusters increases. The ideal number of clusters is typically at the 'elbow' of the curve, where adding more clusters doesn't significantly reduce variation. In this case, the elbow seems to be around 6 clusters, suggesting that 6 clusters might be a good choice for grouping the data effectively.

```r
set.seed(123)

# Perform k-means clustering with the best elbow
kmeans_result <- kmeans(clustering_data, centers = 6, nstart = 25)

# Add cluster to the original data
capmetro <- capmetro %>%
  mutate(cluster = as.factor(kmeans_result$cluster))

# Visualize clusters (hour_of_day vs. ridership)
ggplot(capmetro, aes(x = hour_of_day, y = ridership, color = cluster)) +
  geom_point(alpha = 0.7) +
  facet_wrap(~ day_of_week) +
  labs(title = "Cluster Analysis of Ridership Patterns",
       x = "Hour of Day",
       y = "Ridership",
       color = "Cluster",
       caption = "This graph visualizes the results of a k-means clustering analysis, identifying six\n
  theme_minimal()
```



This graph visualizes the results of a k−means clustering analysis, identifying six distinct ridership patterns across different days of the week. Weekdays show clear morning and evening rush hour peaks, with specific clusters dominating during these times, indicating high and consistent ridership. In contrast, weekends display more varied ridership patterns, with lower overall ridership and more spread−out usage throughout the day.

```
# Print the cluster centers
print(kmeans_result$centers)
```

```
##   hour_of_day day_of_week temperature      month  ridership
## 1   0.9802746  0.02836295  -0.6989597 -0.9543540 -0.4826003
## 2  -0.1545961 -1.27678524   0.3756299  0.1941977  0.9236055
## 3  -0.1602828  1.07175426   0.4720615  0.3693245  1.0622529
## 4   0.9879803 -0.16088916   0.8927088  0.8103033 -0.6063190
## 5  -0.9333127  0.05508446  -1.3361793 -1.0427697 -0.1216477
## 6  -1.0841393 -0.19740241   0.2563410  0.6991313 -1.0038884
```

```
# Summary of the number of points in each cluster
table(capmetro$cluster)
```

```
##
##    1    2    3    4    5    6
## 1040  818 1197 1041  953  775
```

```
library(ggplot2)
library(dplyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

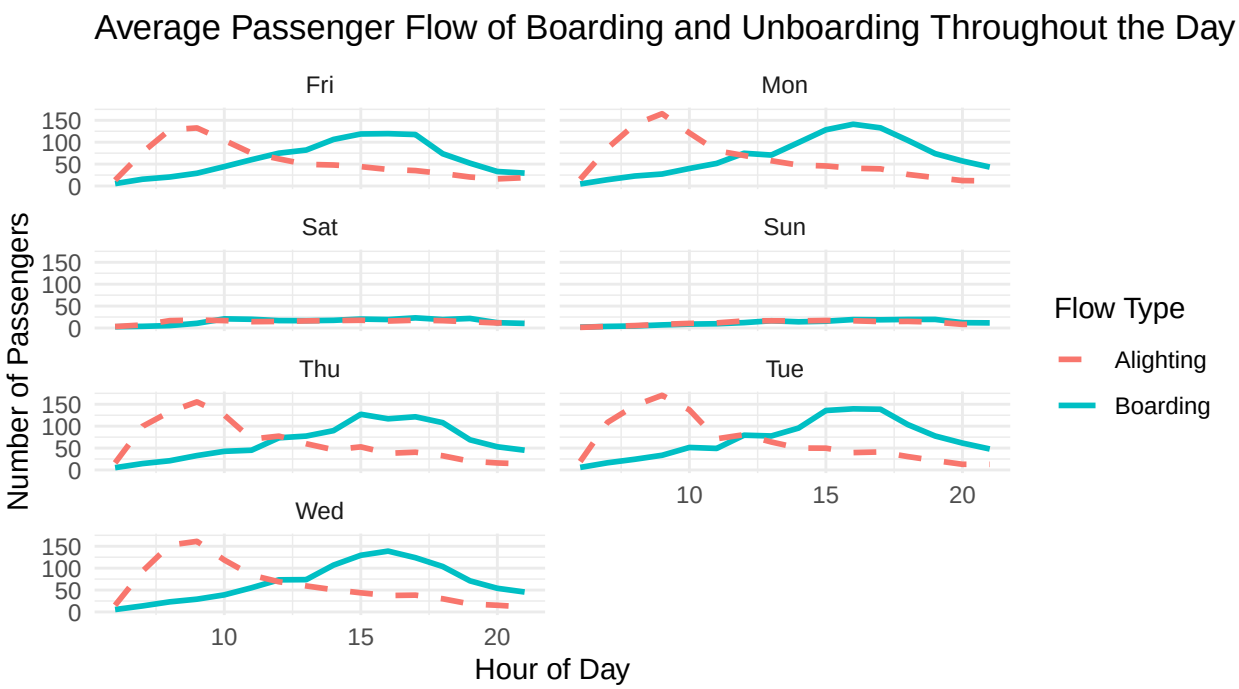
```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
# turn data into by day of the week and hour
capmetro_hourly <- capmetro %>%
  mutate(hour = hour(timestamp)) %>%
  group_by(day_of_week, hour) %>%
  summarise(
    total_boarding = mean(boarding),
    total_alighting = mean(alighting)
  ) %>%
  ungroup() %>%
  arrange(day_of_week, hour)
```

```
## `summarise()` has grouped output by 'day_of_week'. You can override using the
## `.groups` argument.
```

```
# Create a line plot to visualize boarding vs. alighting over time
ggplot(capmetro_hourly, aes(x = hour)) +
  geom_line(aes(y = total_boarding, color = "Boarding"), size = 1) +
  geom_line(aes(y = total_alighting, color = "Alighting"), size = 1, linetype = "dashed") +
  facet_wrap(~ day_of_week, ncol = 2) +
  labs(title = "Average Passenger Flow of Boarding and Unboarding Throughout the Day",
       x = "Hour of Day",
       y = "Number of Passengers",
       color = "Flow Type",
       caption = "These plots show the average number of passengers boarding and alighting buses at\ndi
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



These plots show the average number of passengers boarding and alighting buses at different times of the day for each day of the week. Weekdays (Monday to Friday) exhibit clear peaks during morning (8−9 AM) and afternoon (4−6 PM) commute times, with boarding d alighting patterns closely mirroring each other. On weekends (Saturday and Sunday), ridership is lower and more stable throughout the day. These trends can guide bus scheduling to better match passenger demand, especially during peak hours on weekdays.