Assignment 8

The following code is used to prepare the dataset:

```
library(tidyverse)
```

```
## -- 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 ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## 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 error
```

```
library(lubridate)

circ <- read_csv("Charm_City_Circulator_Ridership.csv")
```

```
## Rows: 1146 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): day, date
## dbl (13): orangeBoardings, orangeAlightings, orangeAverage, purpleBoardings,...
##
## 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.
```

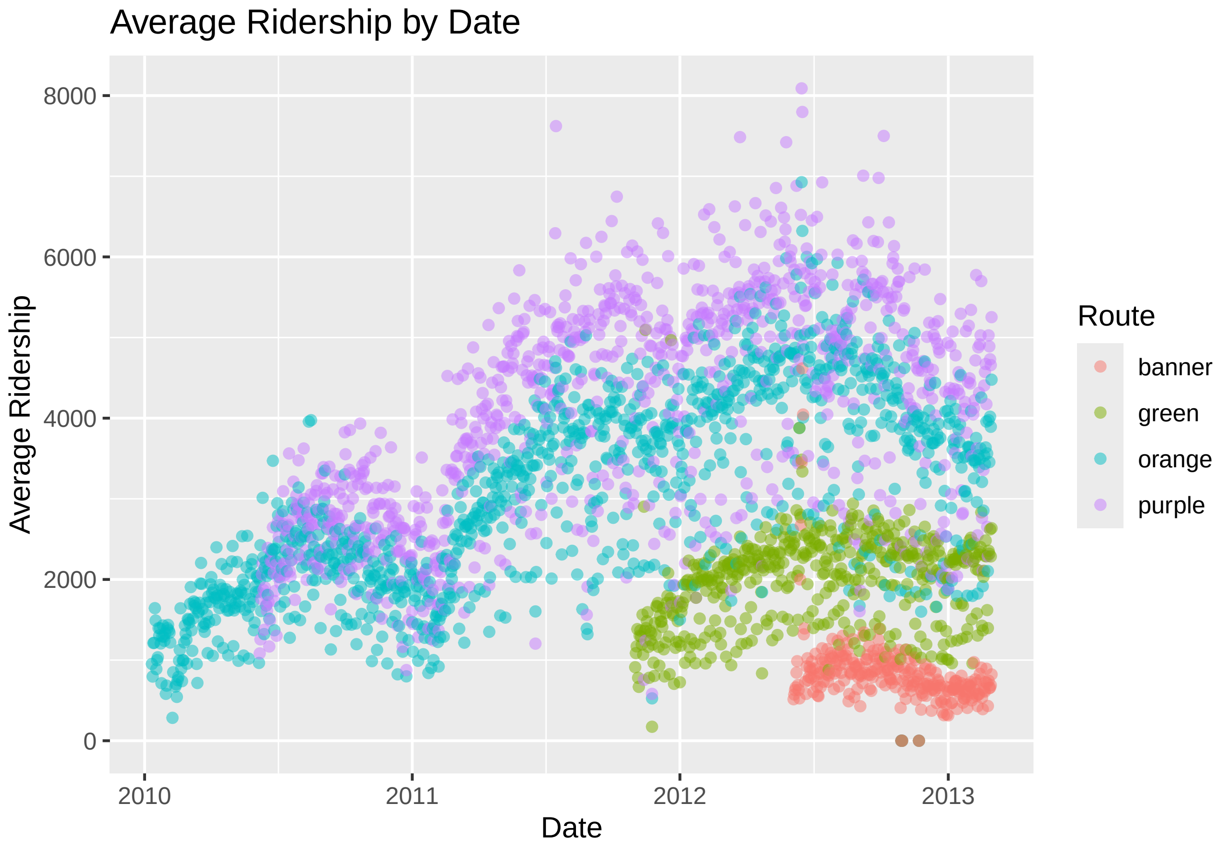
```
# Convert dates
circ <- circ |> mutate(date = mdy(date))

# Change column names for reshaping
colnames(circ) <- colnames(circ) |>
  str_replace("Board", ".Board") |>
  str_replace("Alight", ".Alight") |>
  str_replace("Average", ".Average")

# Make long format by route/type
long <- circ |>
  pivot_longer(
    c(
      starts_with("orange"),
      starts_with("purple"),
      starts_with("green"),
```

```r
      starts_with("banner")
    ),
    names_to = "var",
    values_to = "number"
  )

# Separate route/type into route and type
long <- long |>
  separate(var, into = c("route", "type"), sep = "[.]")

# Make wide format by type
type_wide <- long |>
  spread(type, value = number)

# Extract just average ridership per day
avg <- long |>
  filter(type == "Average", !is.na(number)) |>
  select(-type) |>
  rename(Average = number)
```

Problem 1

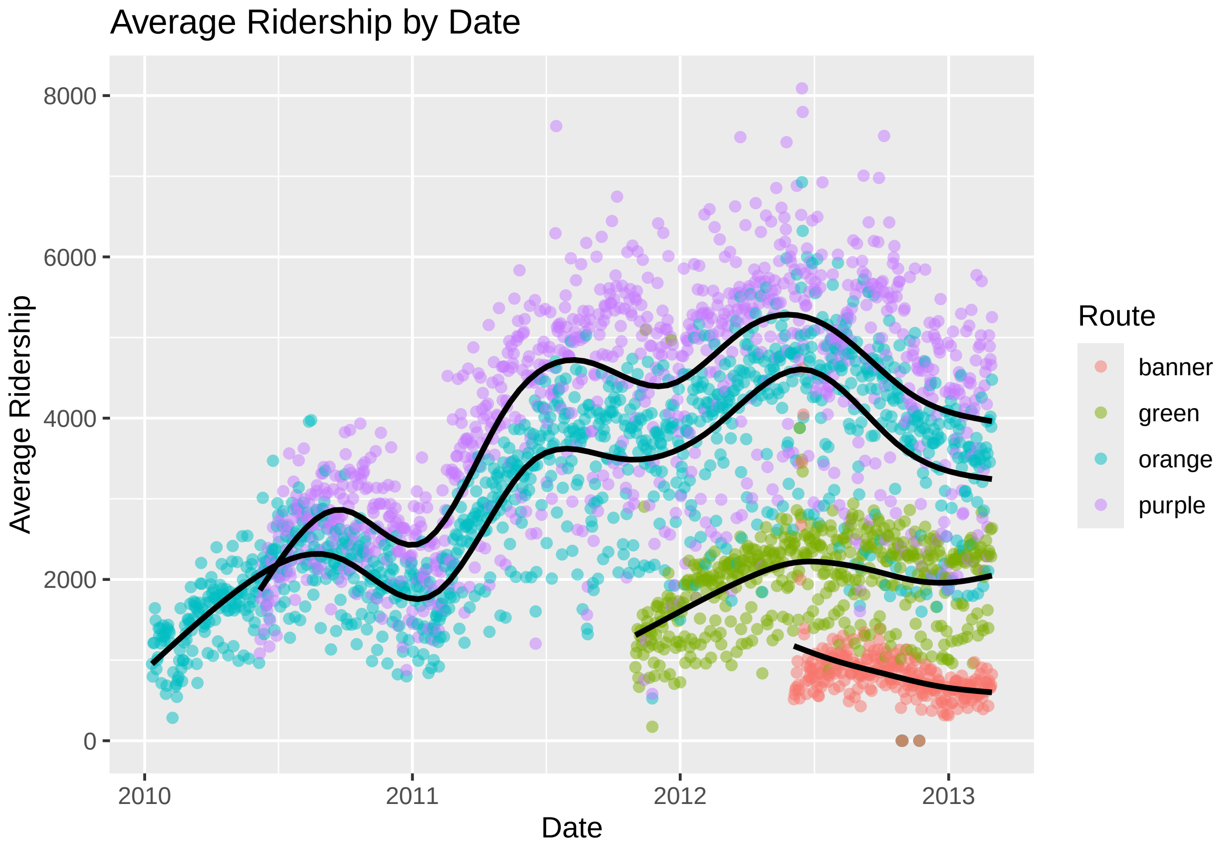
Plot average ridership by date using a scatterplot

```r
ggplot(avg, aes(x = date, y = Average, color = route)) +
  geom_point(alpha = 0.5) +
  labs(
    title = "Average Ridership by Date",
    x = "Date",
    y = "Average Ridership",
    color = "Route")
```



```
ggplot(avg, aes(x = date, y = Average, color = route)) +
  geom_point(alpha = 0.5) +
  geom_smooth(se = FALSE, color = "black", aes(group = route)) +
  labs(
    title = "Average Ridership by Date",
    x = "Date",
    y = "Average Ridership",
    color = "Route")
```

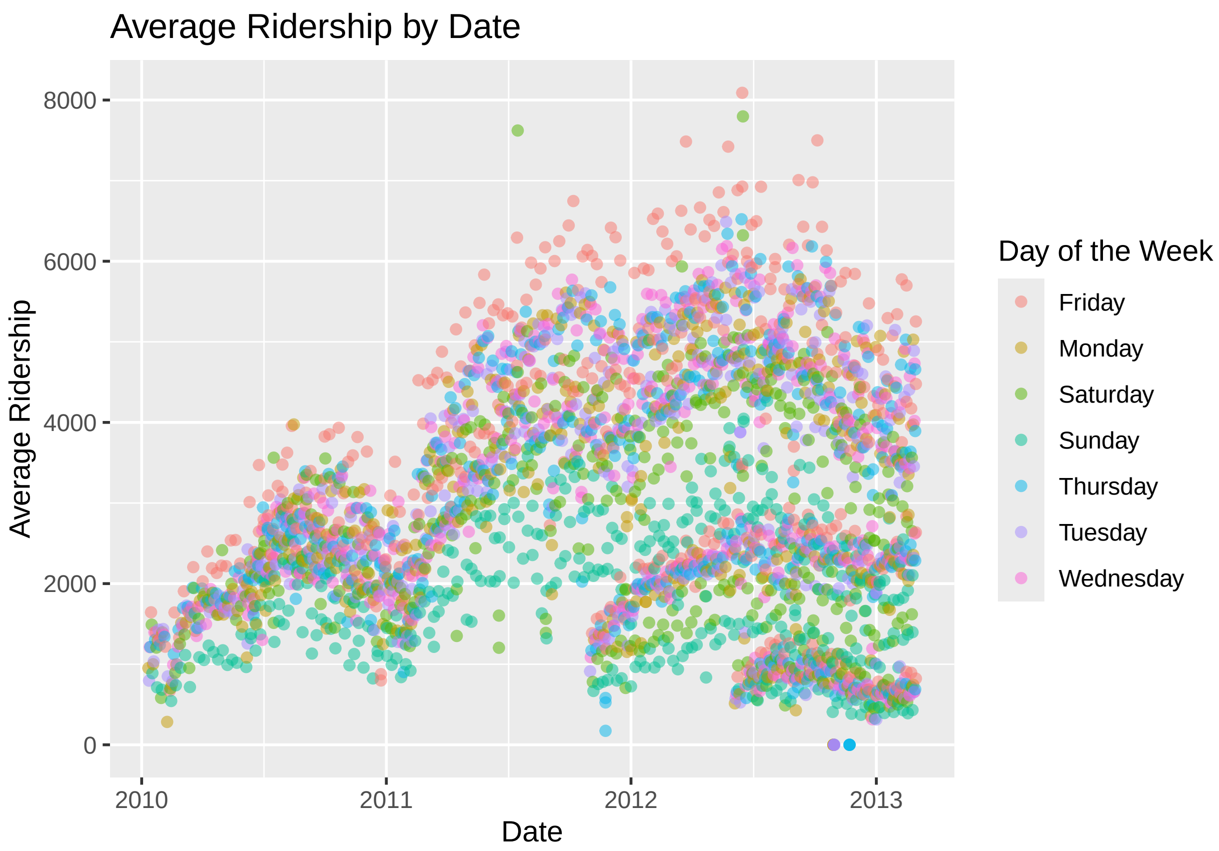
```
## ‘geom_smooth()‘ using method = ’gam’ and formula = ’y ~ s(x, bs = "cs")’
```



Problem 2

Color the points by day of the week.

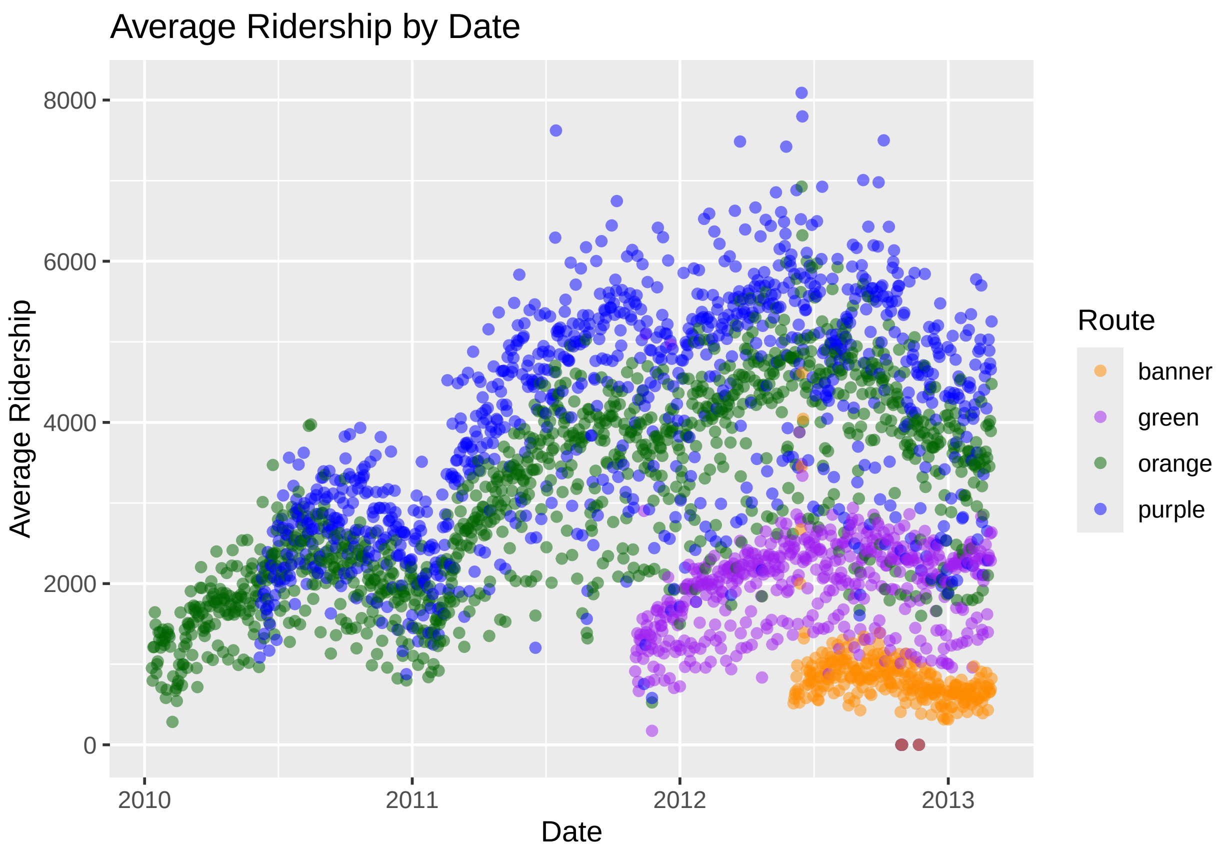
```r
ggplot(avg, aes(x = date, y = Average, color = day)) +
  geom_point(alpha = 0.5) +
  labs(
    title = "Average Ridership by Date",
    x = "Date",
    y = "Average Ridership",
    color = "Day of the Week")
```



Problem 3

Color the points by route, using the following palette: pal <- c("darkorange", "purple", "darkgreen", "blue")

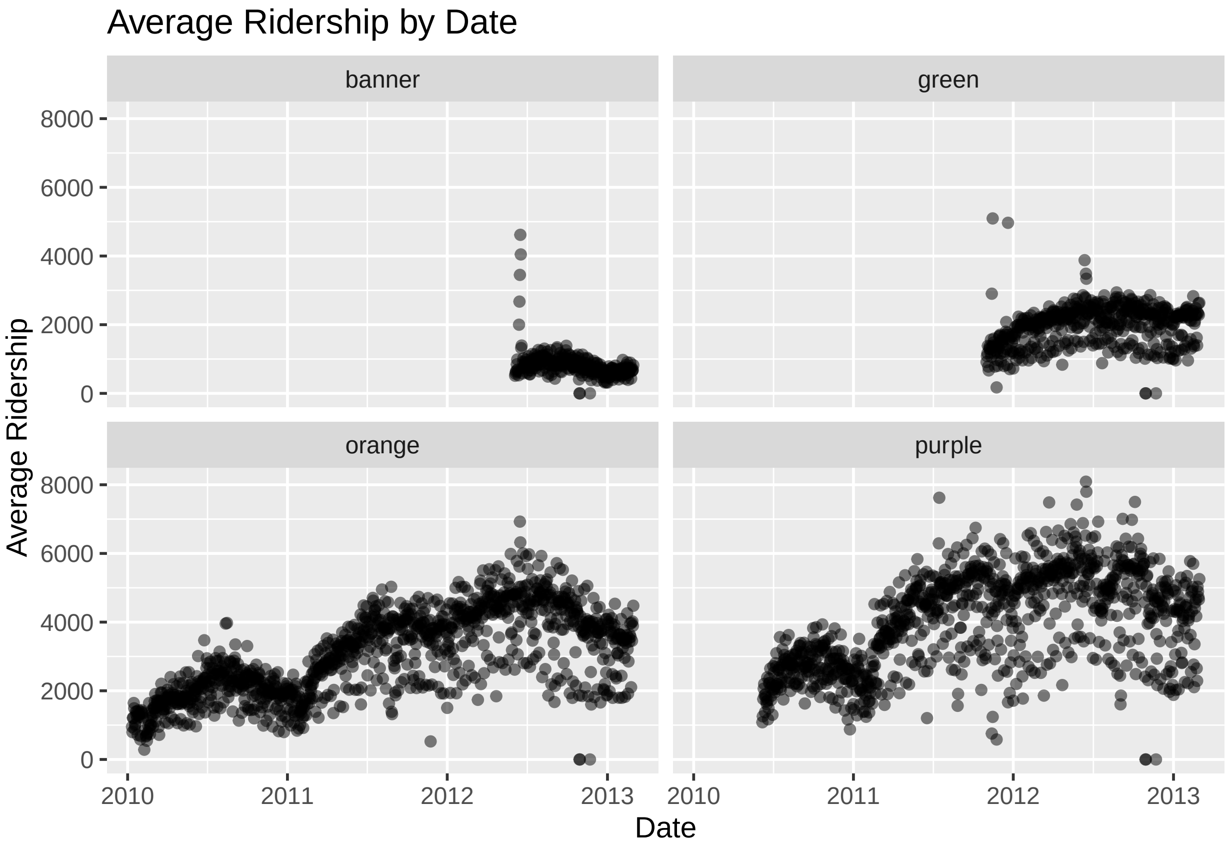
```
pal <- c("darkorange", "purple", "darkgreen", "blue")

ggplot(avg, aes(x = date, y = Average, color = route)) +
  geom_point(alpha = 0.5) +
  scale_color_manual(values = pal) +
  labs(
    title = "Average Ridership by Date",
    x = "Date",
    y = "Average Ridership",
    color = "Route")
```



Problem 4

Plot average ridership by date in separate panels by route.

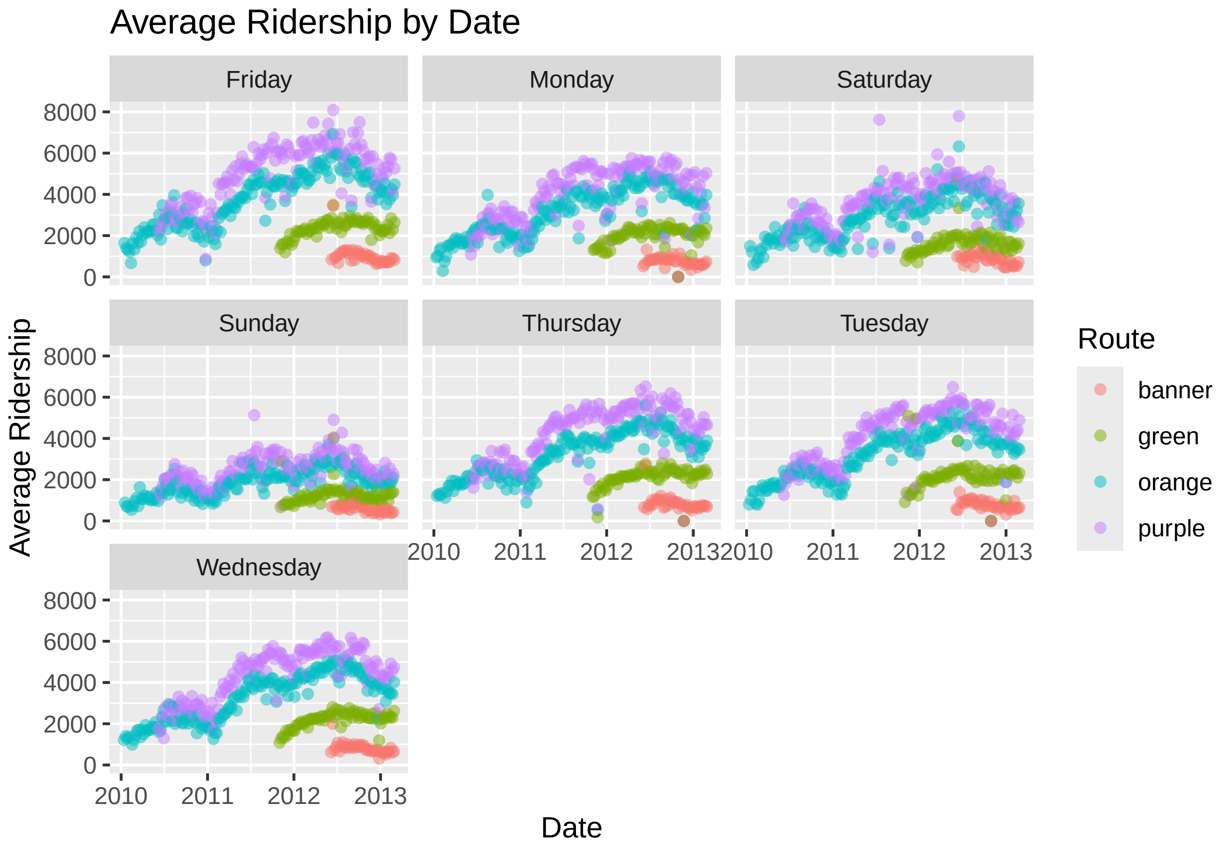
```r
ggplot(avg, aes(x = date, y = Average)) +
  geom_point(alpha = 0.5) +
  facet_wrap(~ route) +
  labs(
    title = "Average Ridership by Date",
    x = "Date",
    y = "Average Ridership")
```



Problem 5

Plot average ridership by date in separate panels by day of the week. Color the points by route.

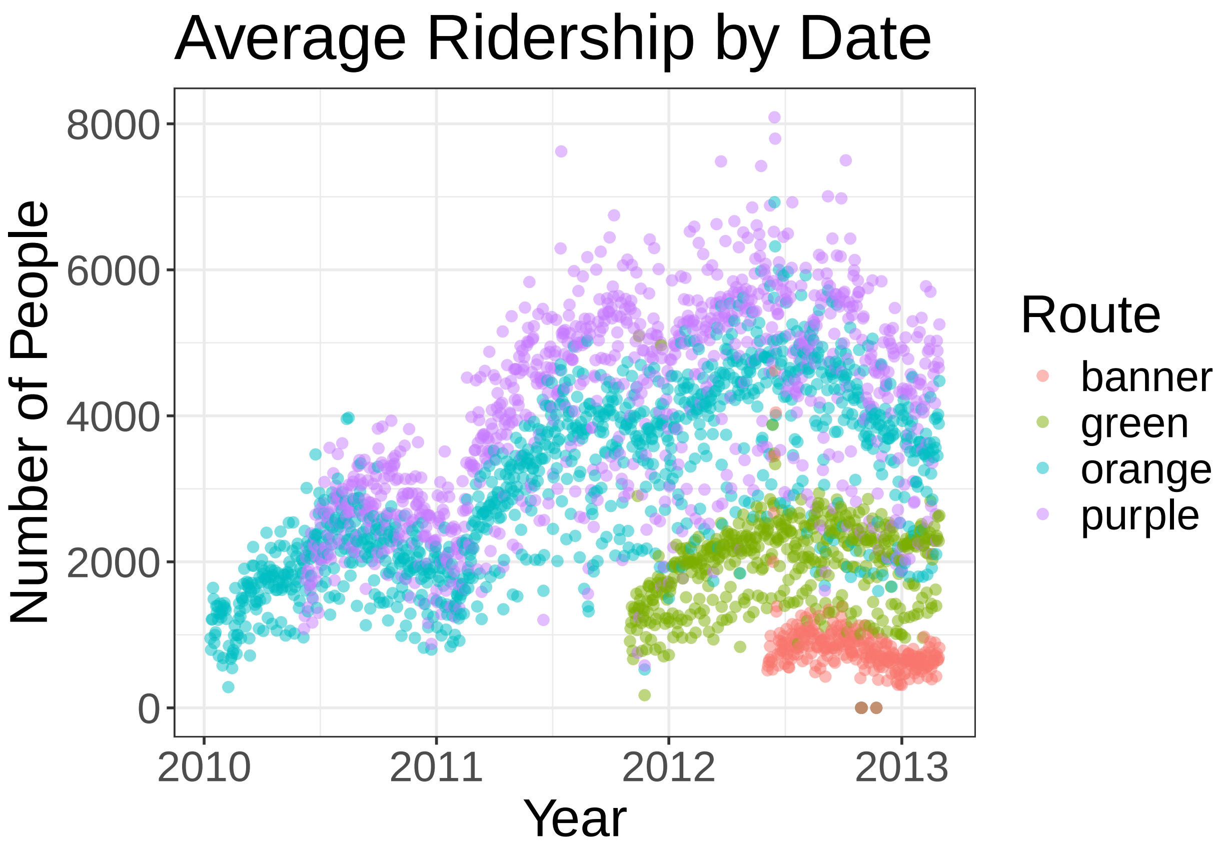
```
ggplot(avg, aes(x = date, y = Average, color = route)) +
  geom_point(alpha = 0.5) +
  facet_wrap(~ day) +
  labs(
    title = "Average Ridership by Date",
    x = "Date",
    y = "Average Ridership",
    color = "Route")
```



Problem 6

Plot average ridership by date. Color the points by route. Customize the plot: Change the x-axis label to “Year”. Change the y-axis label to “Number of People”. Use the black-and-white theme: theme_bw(). Set the text size using theme(text = element_text(size = 20)).

```
ggplot(avg, aes(x = date, y = Average, color = route)) +
  geom_point(alpha = 0.5) +
  labs(
    title = "Average Ridership by Date",
    x = "Year",
    y = "Number of People",
    color = "Route") +
  theme_bw() +
  theme(text = element_text(size = 20))
```

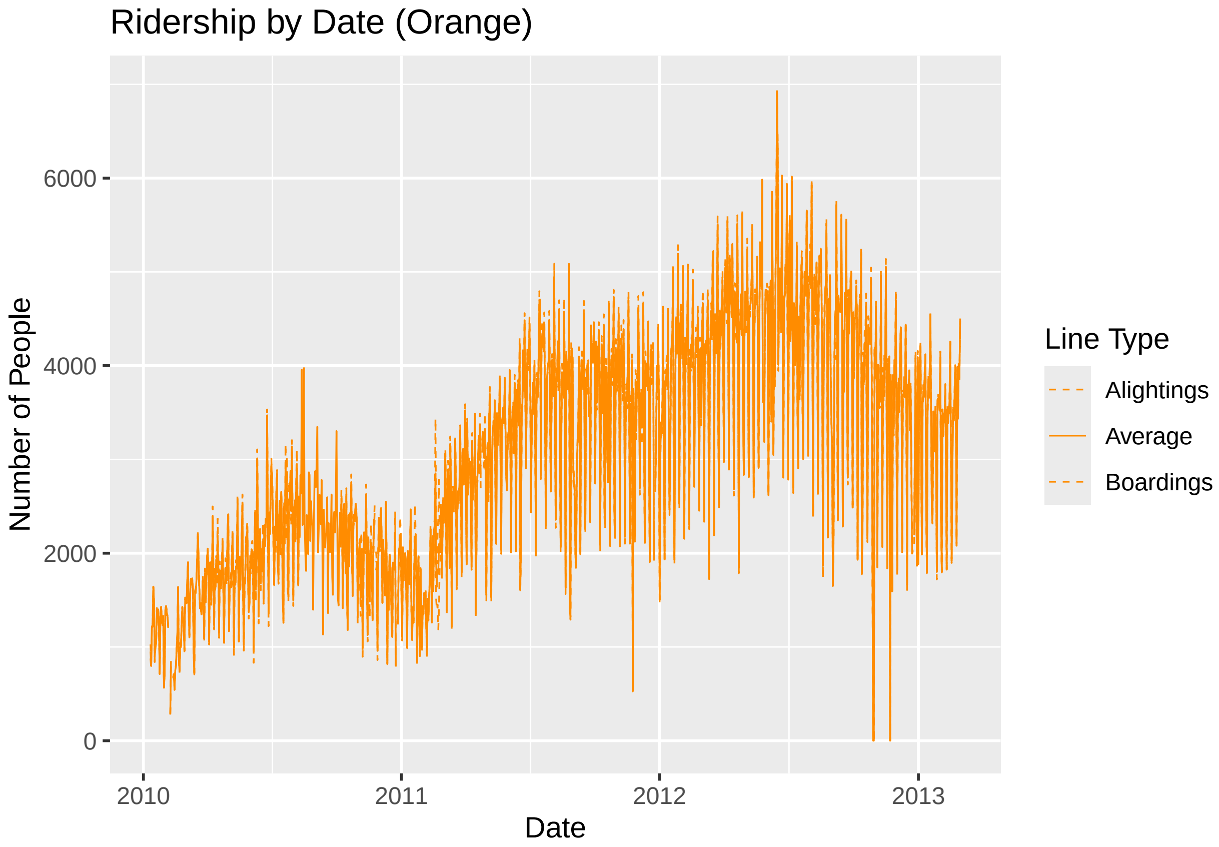


Problem 7

Plot average ridership on the orange route vs. date as a solid line. Add dashed lines for Boardings and Alightings. Use "darkorange" for all line colors. Use linetype as an aesthetic for lines. Customize line types using scale_linetype_manual() with: values = c(Average = "solid", Boardings = "dashed", Alightings = "dashed")

```r
ggplot(filter(long, route == "orange"), aes(x = date, y = number, linetype = type)) +
  geom_line(color = "darkorange", size = 0.3) +
  scale_linetype_manual(values = c("Average" = "solid", "Boardings" = "dashed", "Alightings" = "dashed"
  labs(
    title = "Ridership by Date (Orange)",
    x = "Date",
    y = "Number of People",
    linetype = "Line Type")
```

```
## 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.
```



```
# Assess whether the boardings and alightings differ from the average values.
```

The dashed lines for Boardings and Alightings are hard to see because the solid line overrides them. So, this means there isn't a big difference between the boardings and alightings from the average values.