

Lecture 3: GGLOT2

GGplot2

Spotify data:

```
library(tidyverse)
library(lubridate)
library(titanic)

theme_set(theme_minimal())
spotify <- read_csv("homework_assignments/homework_2/streaming_data.csv")
titanic <- titanic_train %>%
  as_tibble() %>%
  janitor::clean_names()
```

Let's clean the data:

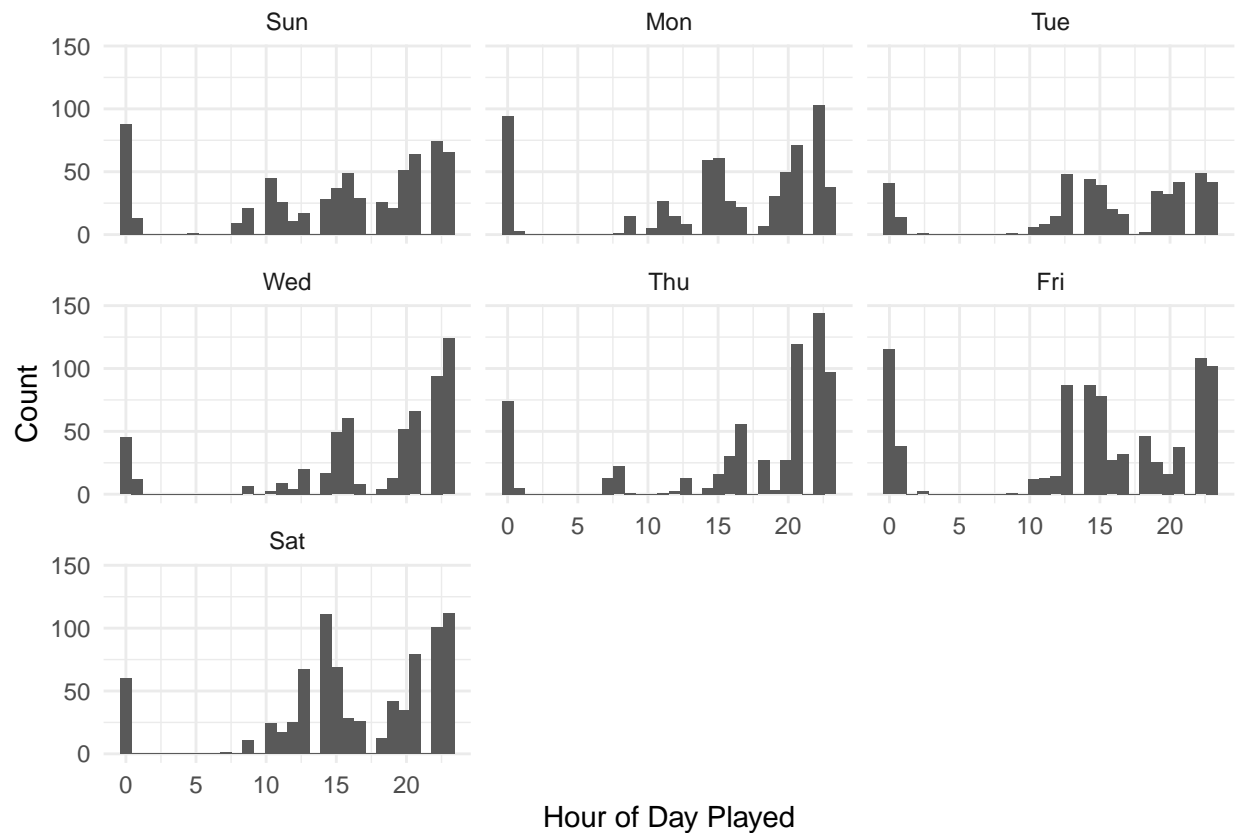
```
spotify <- spotify %>%
  mutate(seconds_played = ms_played/1000,
         minutes_played = seconds_played/60) %>%
  mutate(end_time= with_tz(end_time, tz = "America/Los_Angeles")) %>%
  mutate(time_played = hms::as_hms(end_time), .before = 1) %>%
  mutate(hour_played = hour(time_played)) %>%
  mutate(day = wday(end_time, label = T ))
```

Now let's make some graphs.

Density/Histogram

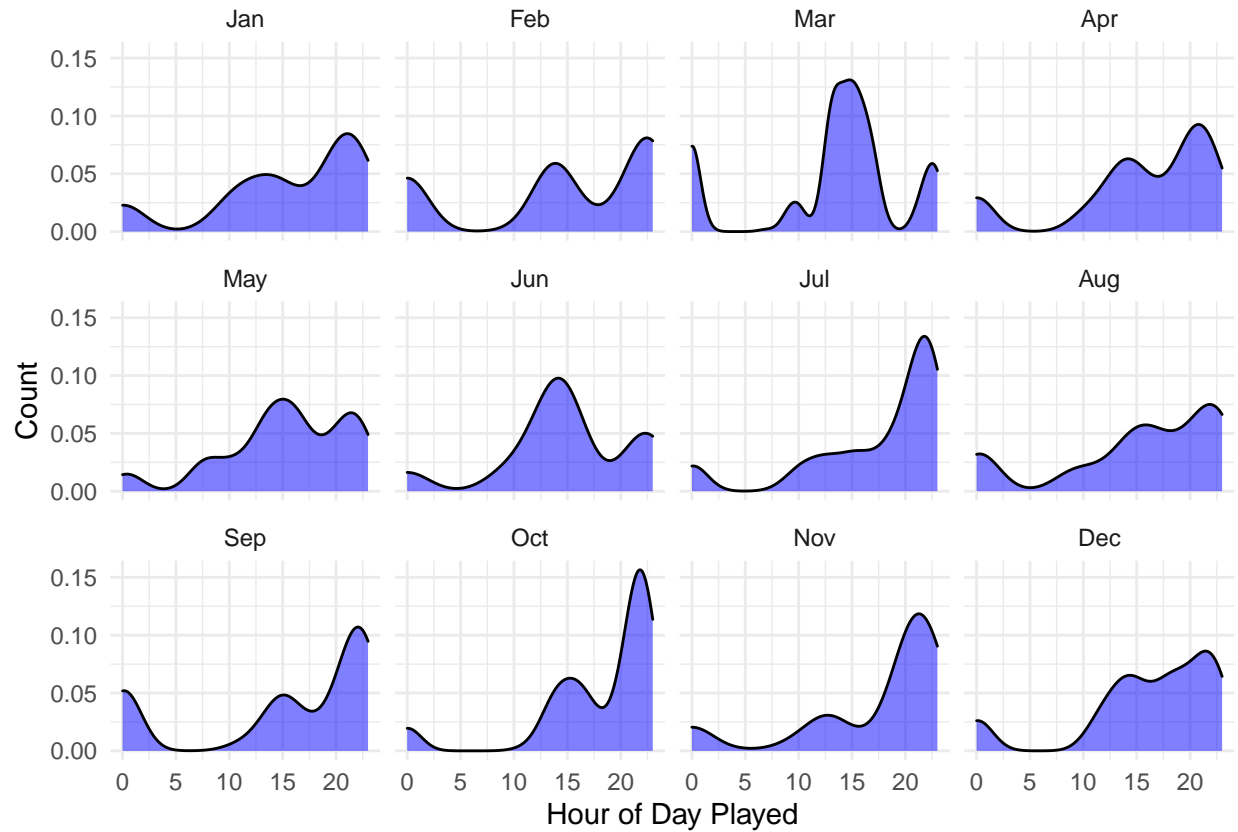
This first graph will show the power of ggplot2 and switching between layers

```
spotify %>%
  filter(seconds_played > 5) %>%
  mutate(month = month(end_time)) %>%
  ggplot(aes(hour_played)) +
  geom_histogram() +
  facet_wrap(~day) +
  labs(y = "Count", x = "Hour of Day Played")
```



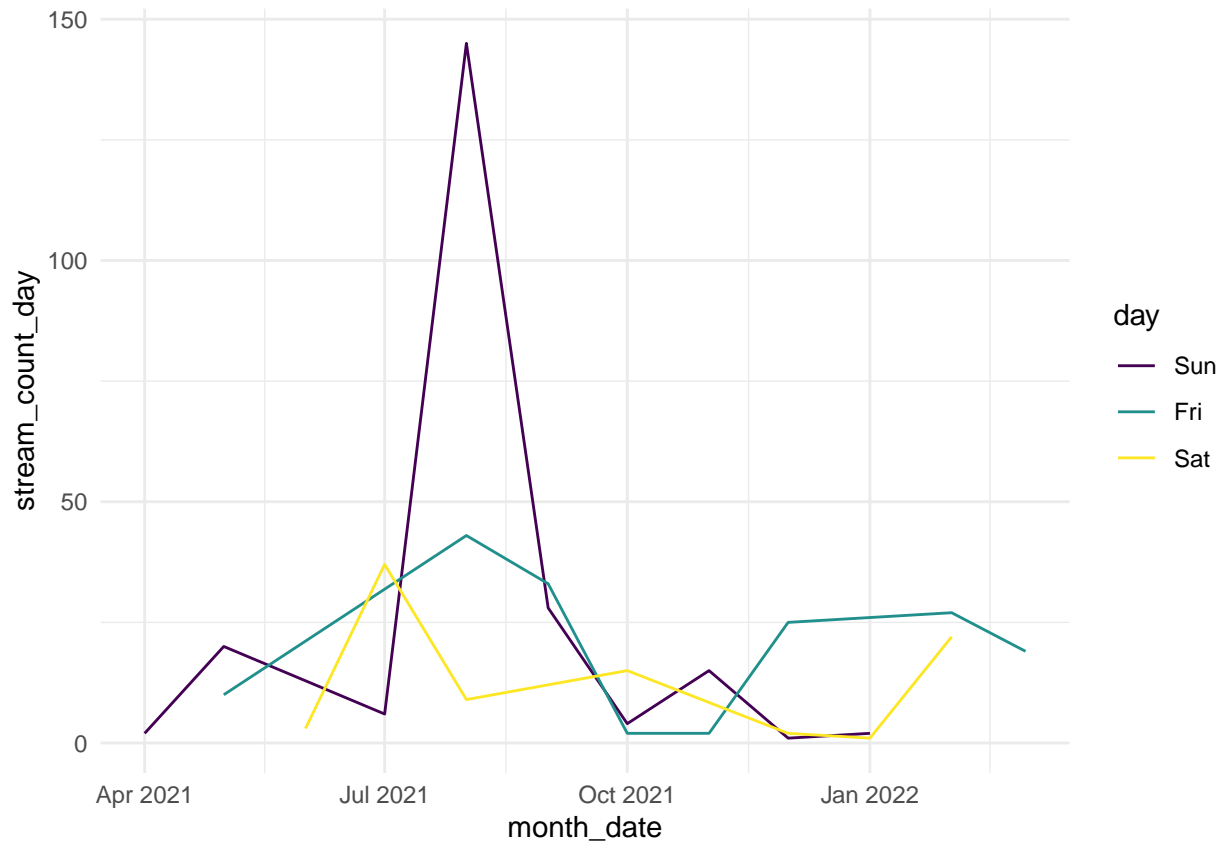
Start with just doing the hour played histogram, then add facet, then filter, then labs, then change to a density, then add a fill:

```
spotify %>%
  filter(seconds_played > 5) %>%
  mutate(month = month(end_time, label = T)) %>%
  ggplot(aes(hour_played)) +
  geom_density(fill = "blue", alpha = 0.5) +
  facet_wrap(~month) +
  labs(y = "Count", x = "Hour of Day Played")
```

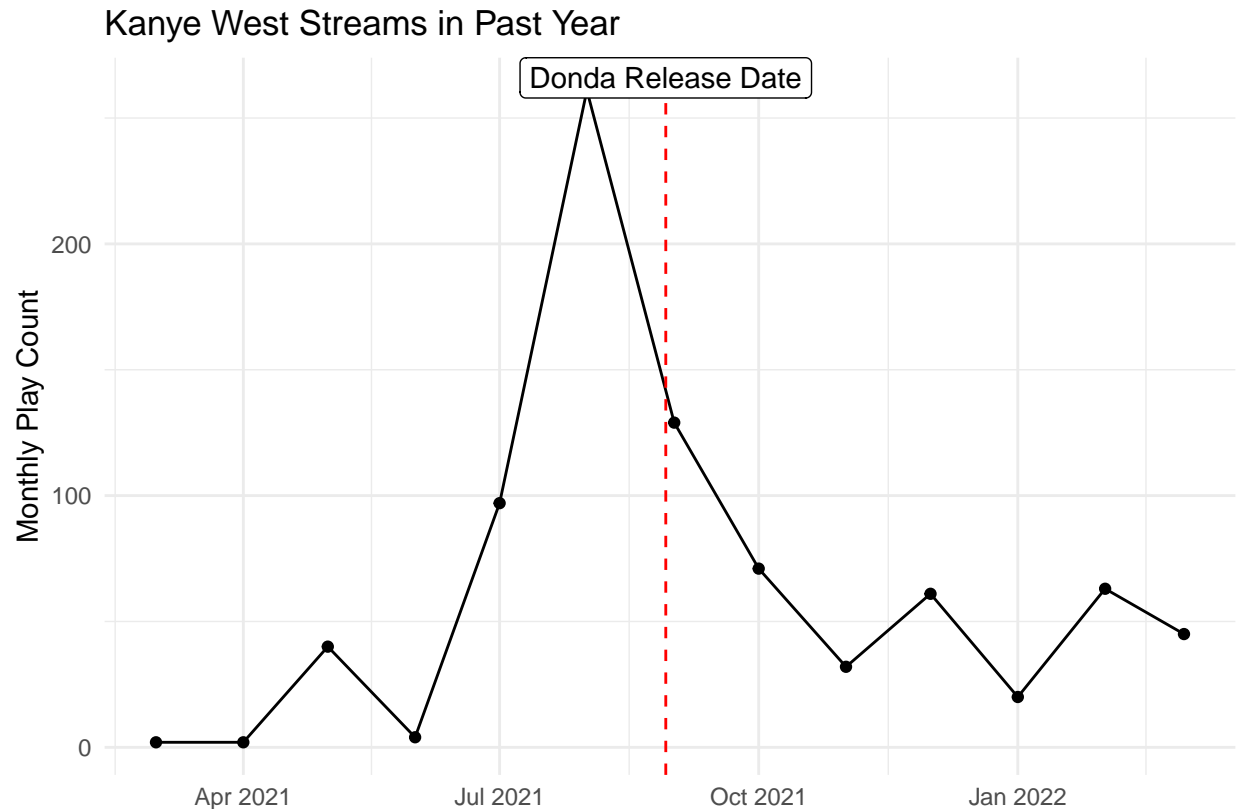


Time Plot

```
spotify %>%
  filter(artist_name == "Kanye West") %>%
  filter(day == "Fri" | day == "Sat" | day == "Sun") %>%
  mutate(month = month(end_time), year = year(end_time)) %>%
  mutate(month_date = ymd(paste0(year, "-", month, "-1"))) %>%
  group_by(day, month_date) %>%
  summarize(stream_count_day = n()) %>%
  ggplot(aes(month_date, stream_count_day, color = day)) +
  geom_line()
```



```
spotify %>%
  filter(artist_name == "Kanye West") %>%
  mutate(month = month(end_time), year = year(end_time)) %>%
  mutate(month_date = ymd(paste0(year, "-", month, "-1"))) %>%
  group_by(month_date) %>%
  summarize(month_count = n()) %>%
  ggplot(aes(month_date, month_count)) +
  geom_line() +
  geom_point() +
  geom_vline(xintercept = ymd("2021-08-29"), linetype = "dashed", color = "red") +
  annotate(x = ymd("2021-08-29"), y = Inf, label = 'Donda Release Date', geom = "label",
           vjust = 1) +
  labs(x = " ", y = "Monthly Play Count", title = "Kanye West Streams in Past Year")
```

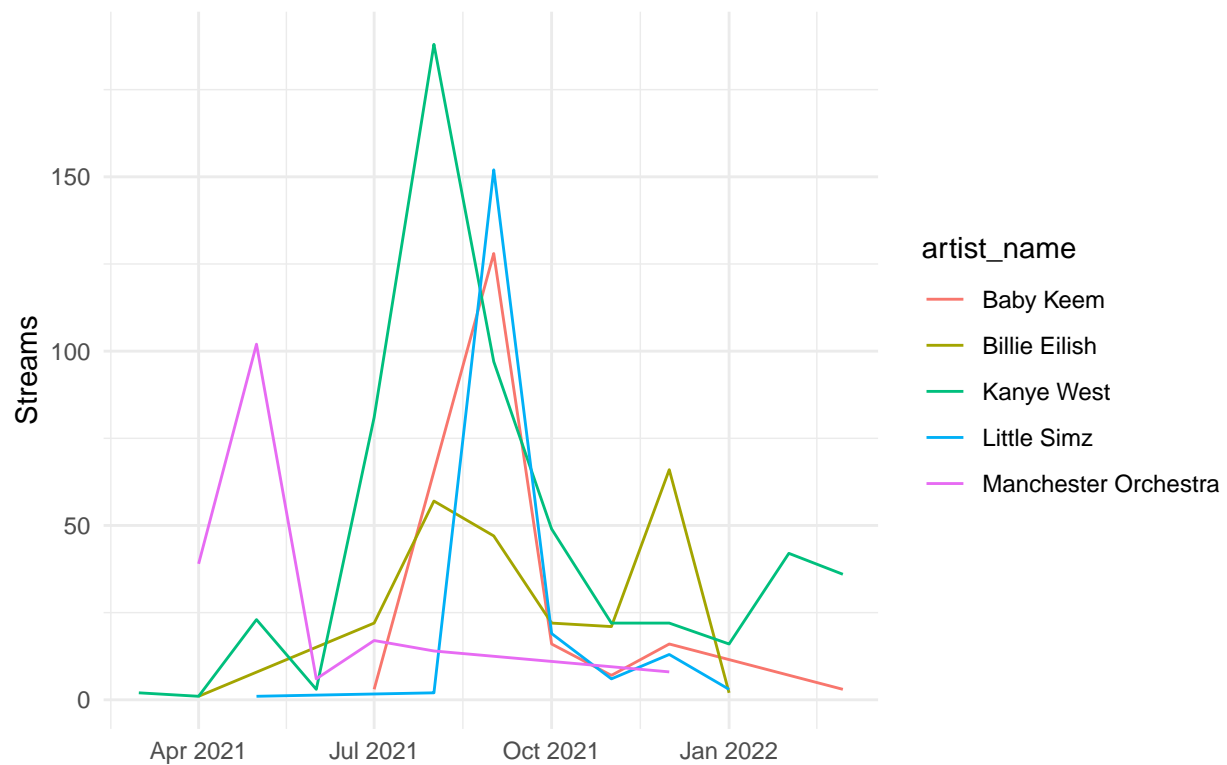


Can put multiple lines on the same graph.

Favorite artist streams. This plot is good for the following reasons: 1. You get to understand the color argument. 2. You get to understand more how the labs argument works with color. 3. You can understand why this is a bad graph.

```
spotify %>%
  filter(artist_name == "Kanye West" |
         artist_name == "Billie Eilish" |
         artist_name == "Baby Keem" |
         artist_name == "Little Simz" |
         artist_name == "Manchester Orchestra") %>%
  mutate(month = month(end_time), year = year(end_time)) %>%
  mutate(month_date = ymd(paste0(year, "-", month, "-1"))) %>%
  group_by(month_date, artist_name) %>%
  filter(seconds_played > 60) %>%
  summarize(streams_per_day = n()) %>%
  arrange(month_date) %>%
  ggplot(aes(month_date, streams_per_day, color = artist_name)) +
  geom_path() +
  labs(title = "Streams Per-Month of Favorite Artists", x = " ", y = "Streams")
```

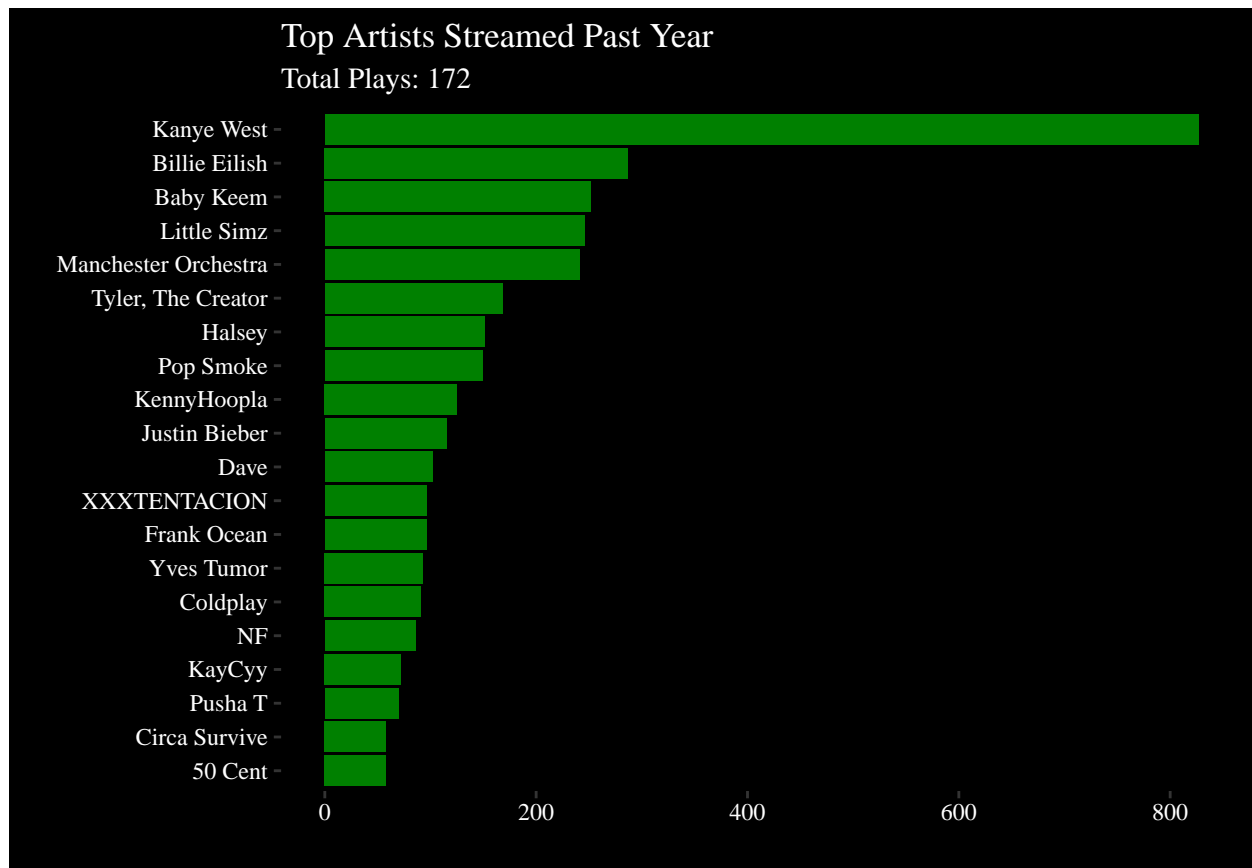
Streams Per-Month of Favorite Artists



Change this to a facet wrap.

Stacking Bar Plot

```
spotify %>%
  count(artist_name, sort = T) %>%
  head(20) %>%
  mutate(artist_name = fct_reorder(artist_name, n)) %>%
  ggplot(aes(x = n, y = artist_name)) +
  geom_col(fill = "green", alpha = 0.5) +
  geom_text(aes(label = n), hjust = -0.1) +
  labs(x = "Number of Streams", y = " ", title = "Top Artists Streamed Past Year",
       subtitle = "Total Plays: 172") +
  ggthemes::theme_tufte() +
  theme(plot.background = element_rect(fill = "black"),
        axis.text.x = element_text(color = "white"),
        axis.text.y = element_text(color = "white"),
        plot.title = element_text(color = "white"),
        plot.subtitle = element_text(color = "white"))
```



Titanic Data

1. Make a graph that scatters ticket fare on age and then do a color argument.
2. Do a fill argument

Assignment:

Replicate Figure 1



Figure 1: Boxplot of Titanic Survivors

Hints:

1. Start with a normal boxplot for pclass and age
2. Use the `aes()` argument and facet wraps.
3. mutate your variables to clean up the graph.

Assignment 2: Replicate Figure 2.

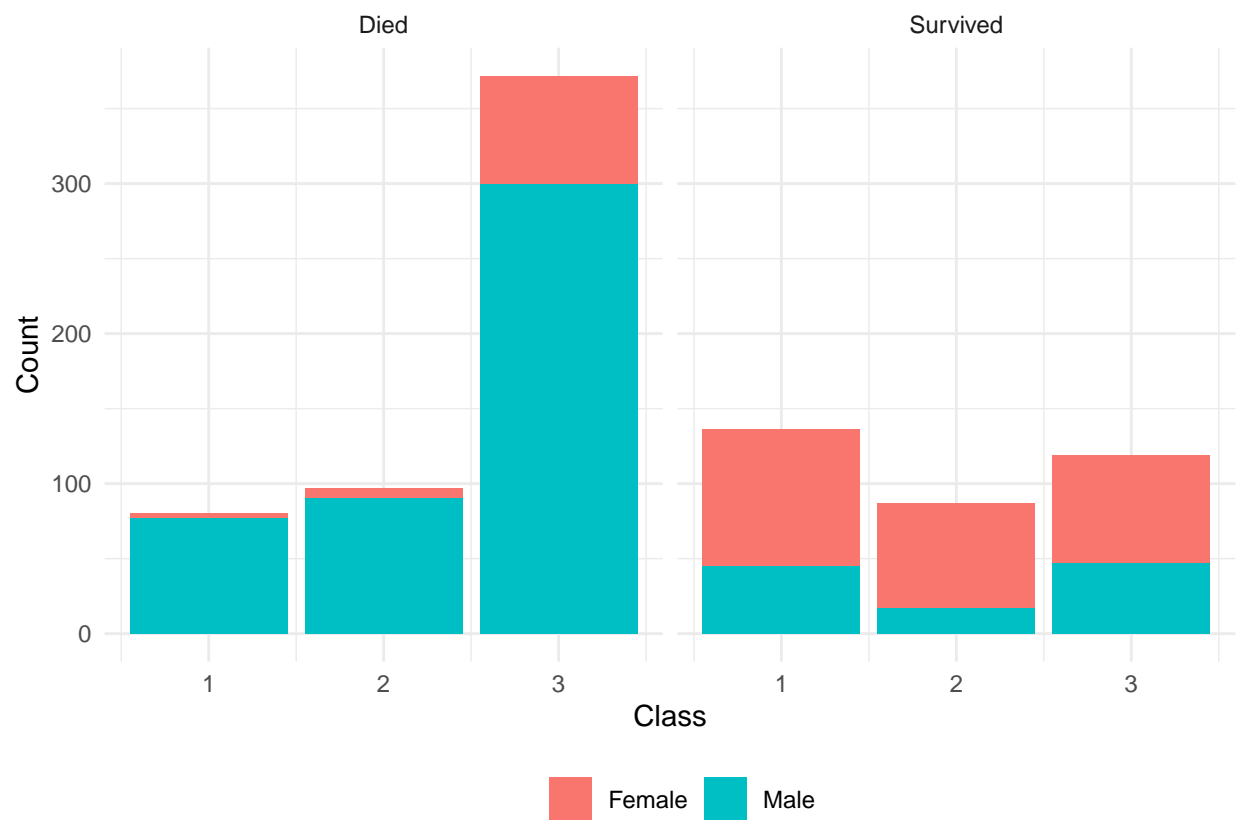


Figure 2: Graph of Classes and Number of Survivors by Sex