Lecture 3: GGPLOT2

GGplot2

Spotify data:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'readr' was built under R version 4.0.5
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

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

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

```
library(titanic)

theme_set(theme_minimal())
spotify <- read_csv("homework_assignments/homework_2/streaming_data.csv")
```

```
## Rows: 5159 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): artist_name, track_name
## dbl  (1): ms_played
## dttm (1): end_time
##
## 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.
```

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))
```

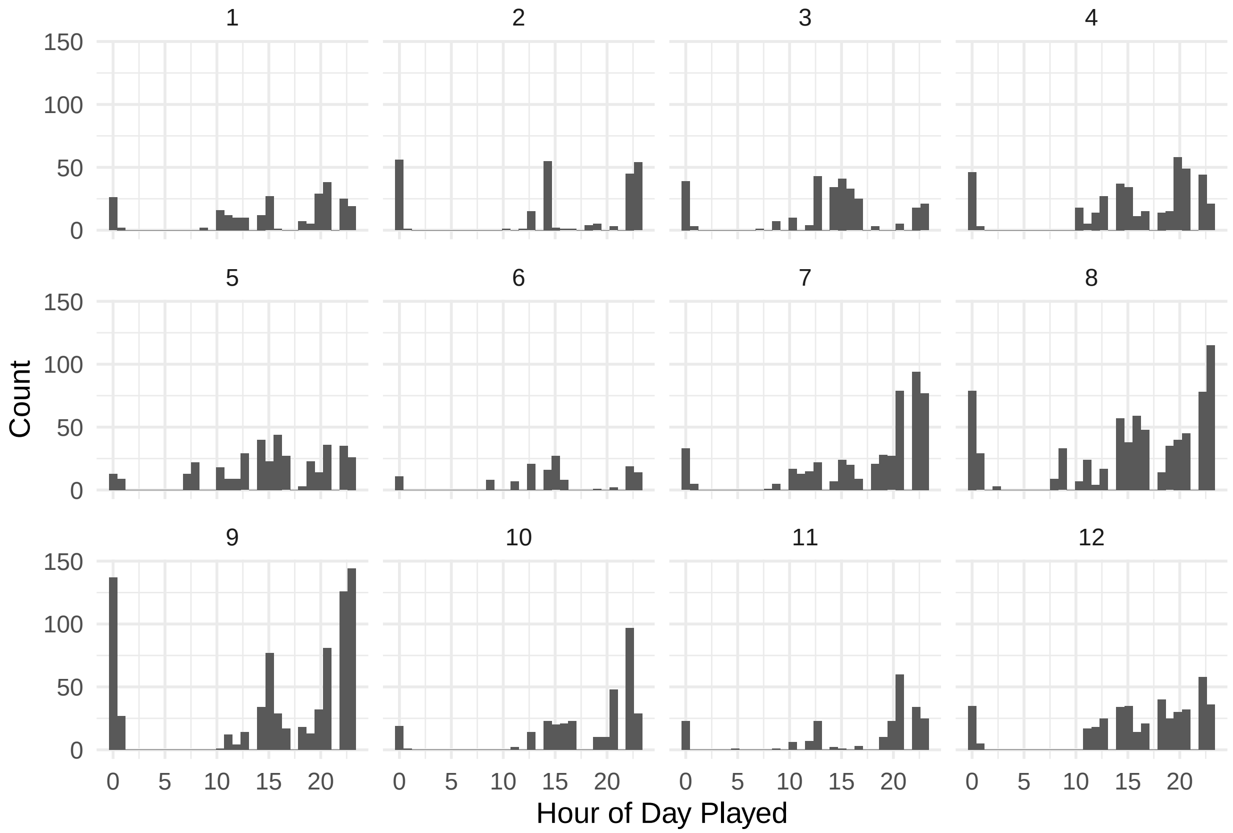
Now let’s make some graphs.

Density/Histogram

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

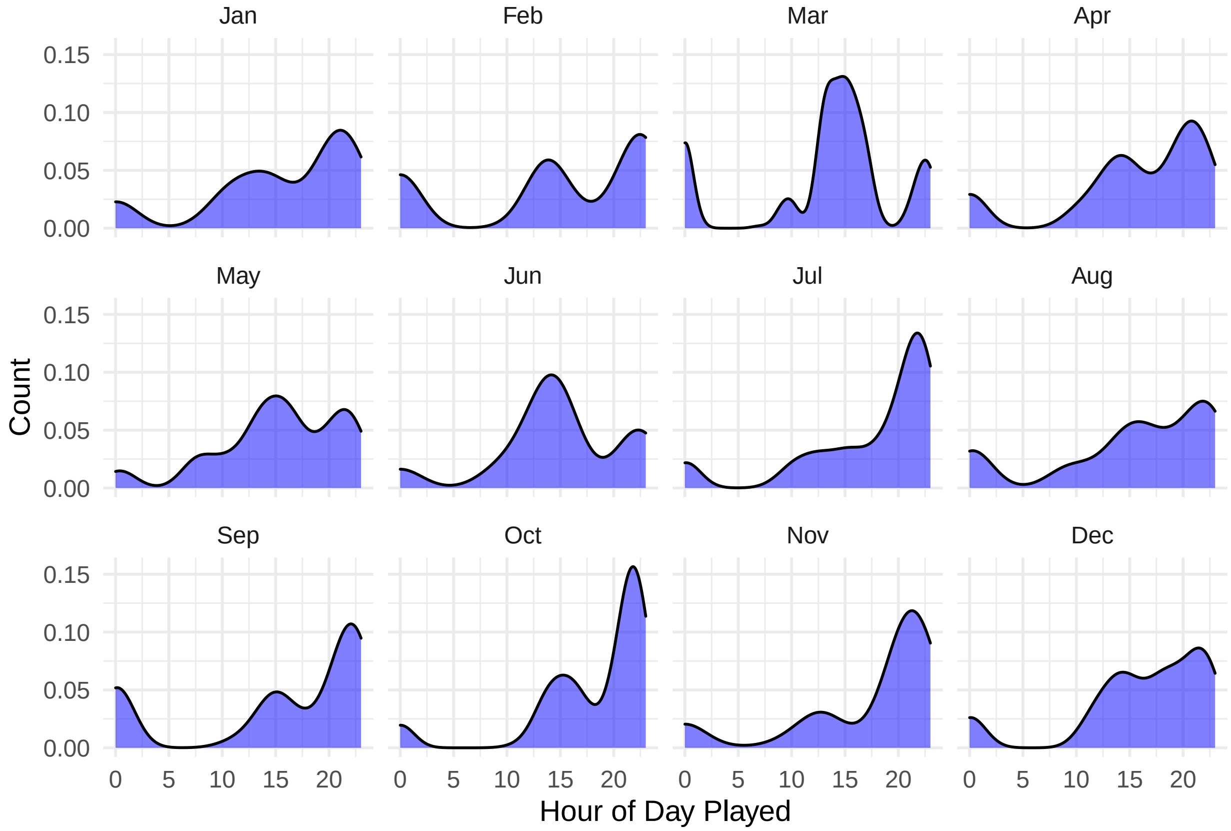
```
spotify %>%
  filter(seconds_played > 5) %>%
  mutate(month = month(end_time)) %>%
  ggplot(aes(hour_played)) +
  geom_histogram() +
  facet_wrap(~month) +
  labs(y = "Count", x = "Hour of Day Played")
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



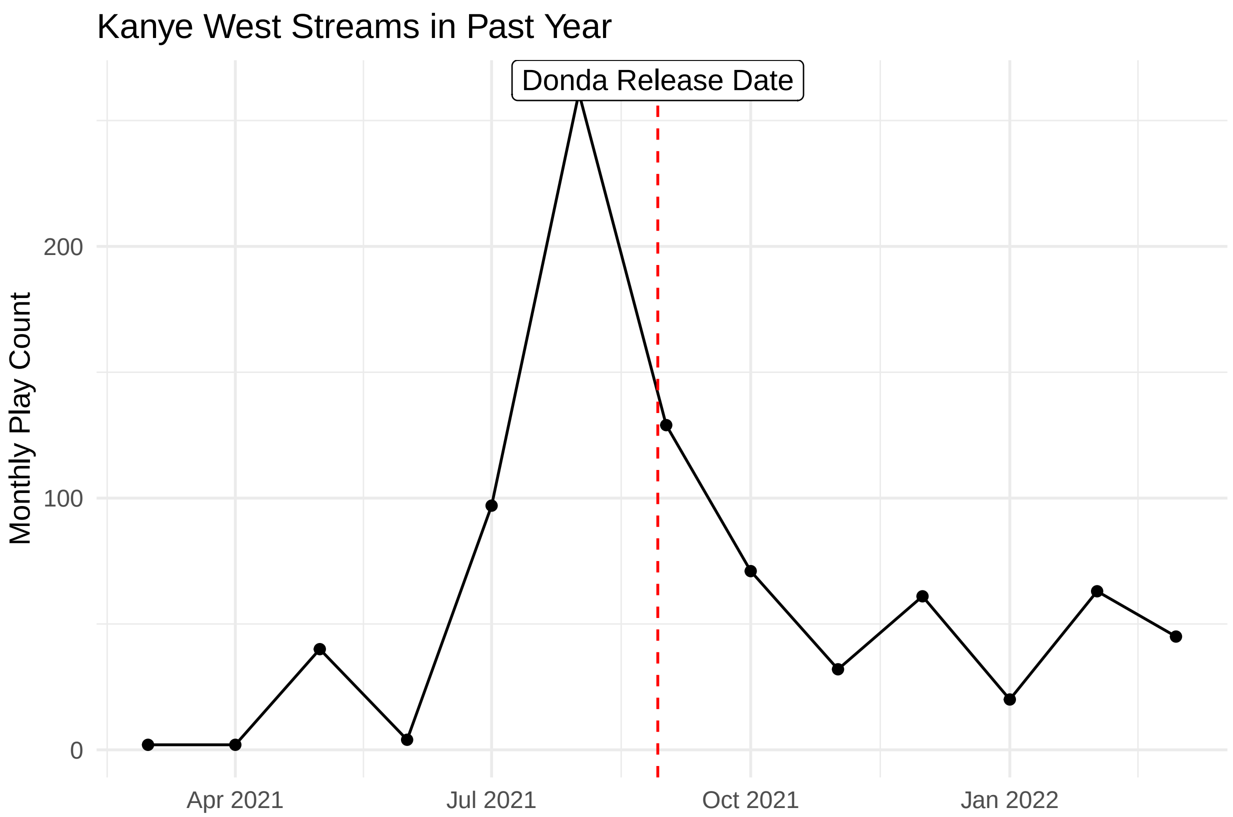
Start with just doing the hour played histogram, then add facet, then filter, then labs, then change to a density, then add a fill:

```
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") %>%
  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")
```

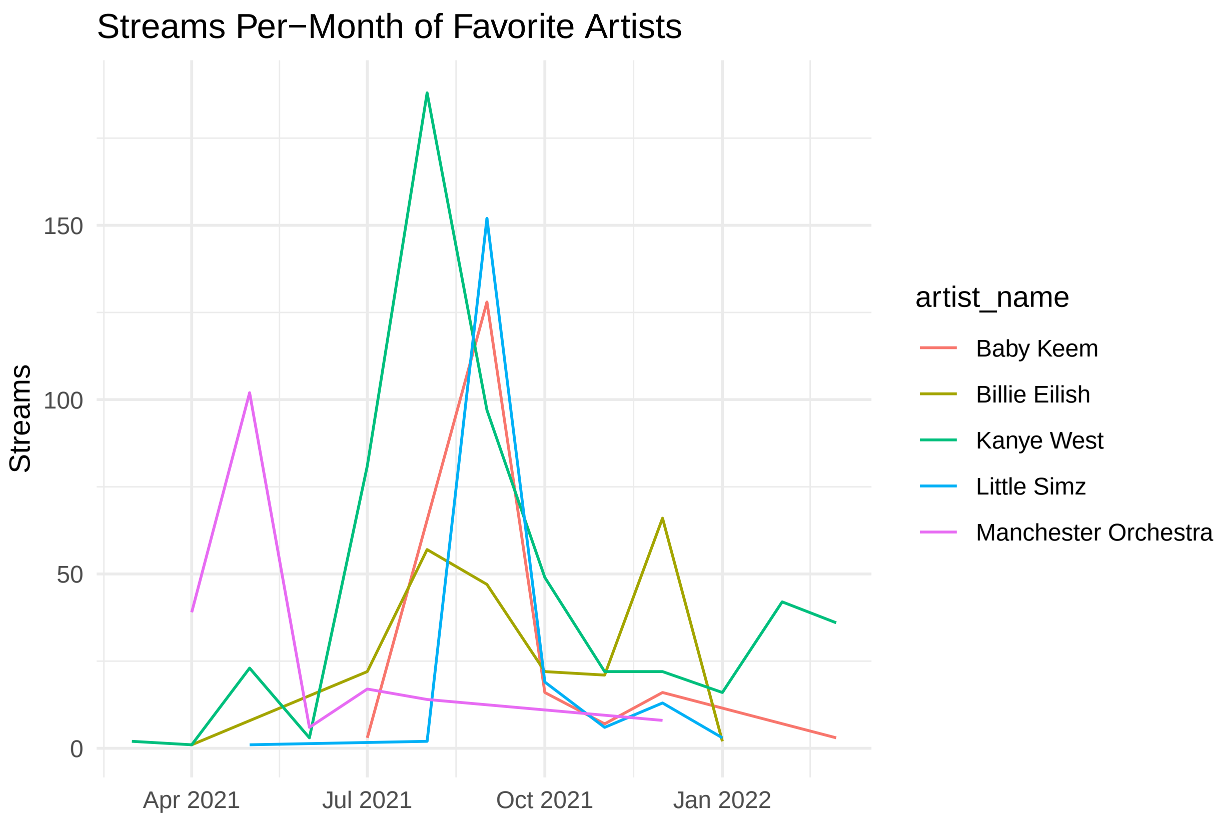


Can put multiple lines on the same graph.

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")
```

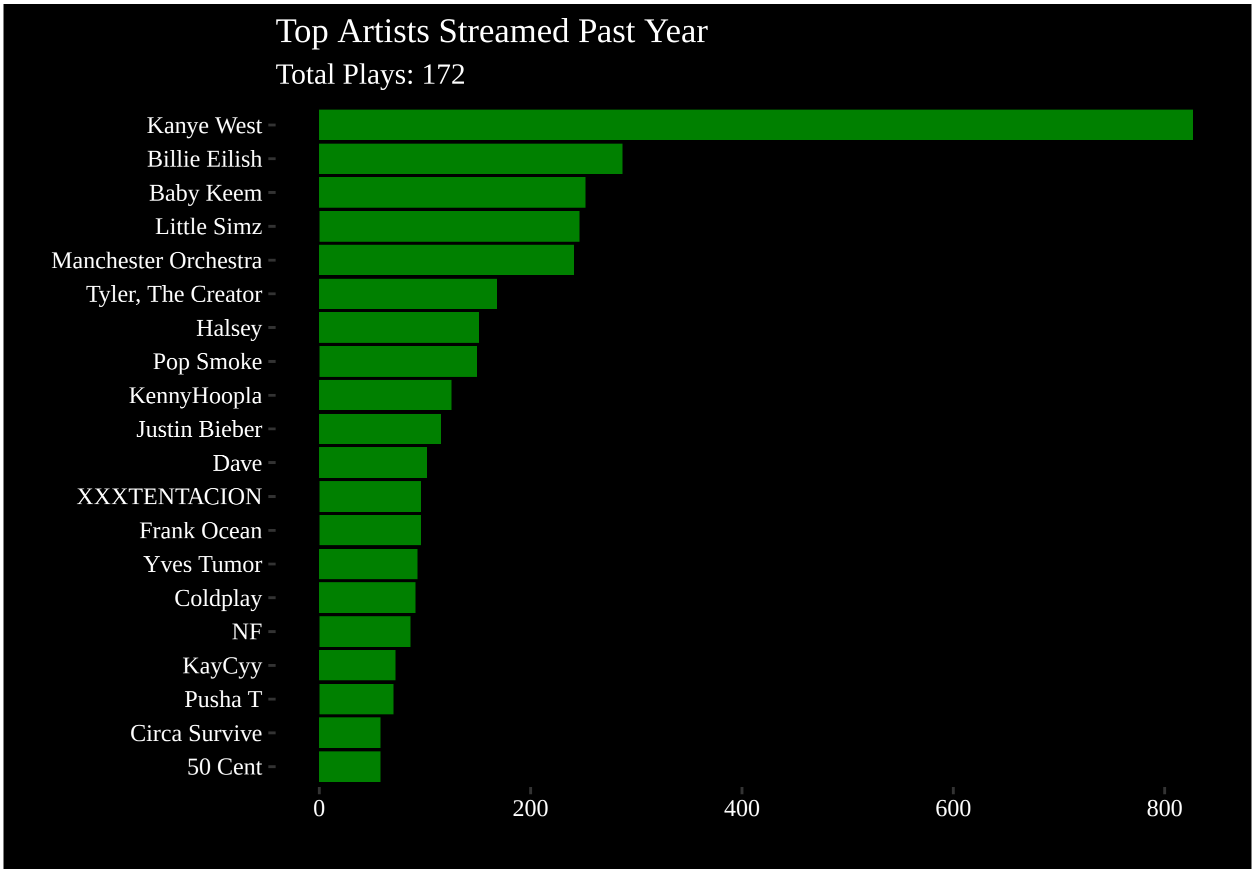
```
## `summarise()` has grouped output by 'month_date'. You can override using the
## `.groups` argument.
```



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