Lab 2

Omar Youssif

Math 241, Week 2

```
# Put all necessary libraries here
# I got you started!
# The first time you want to install the dsbox package; then you can comment it out.
# If you have not installed the devtools package, you will need to do so first
# install.packages("devtools")
# library(devtools)

devtools::install_github("tidyverse/dsbox")
library(dsbox)
library(tidyverse)
library(viridis)
library(readr)
library(ggrepel)
```

Due: Thursday, February 8th at 8:30am

Goals of this lab

1. Practice coding to adhere to the Tidyverse Style Guide.
2. Practice creating and refining graphs with `ggplot2`.
3. Consider the strengths and weaknesses of various `geoms` and `aesthetics` for telling a data story.

Notes:

- When creating your graphs, consider context (i.e. axis labels, title, . . .)!
- If I provide partially completed code, I will put `eval = FALSE` in the chunk. Make sure to change that to `eval = TRUE` once you have completed the code in the chunk.
- Be prepared to ask for help from me, Simon, and your classmates! We scratched the surface of `ggplot2` in class. But I encourage you to really dig in and make your graphs your own (i.e. don't rely on defaults).

Problems

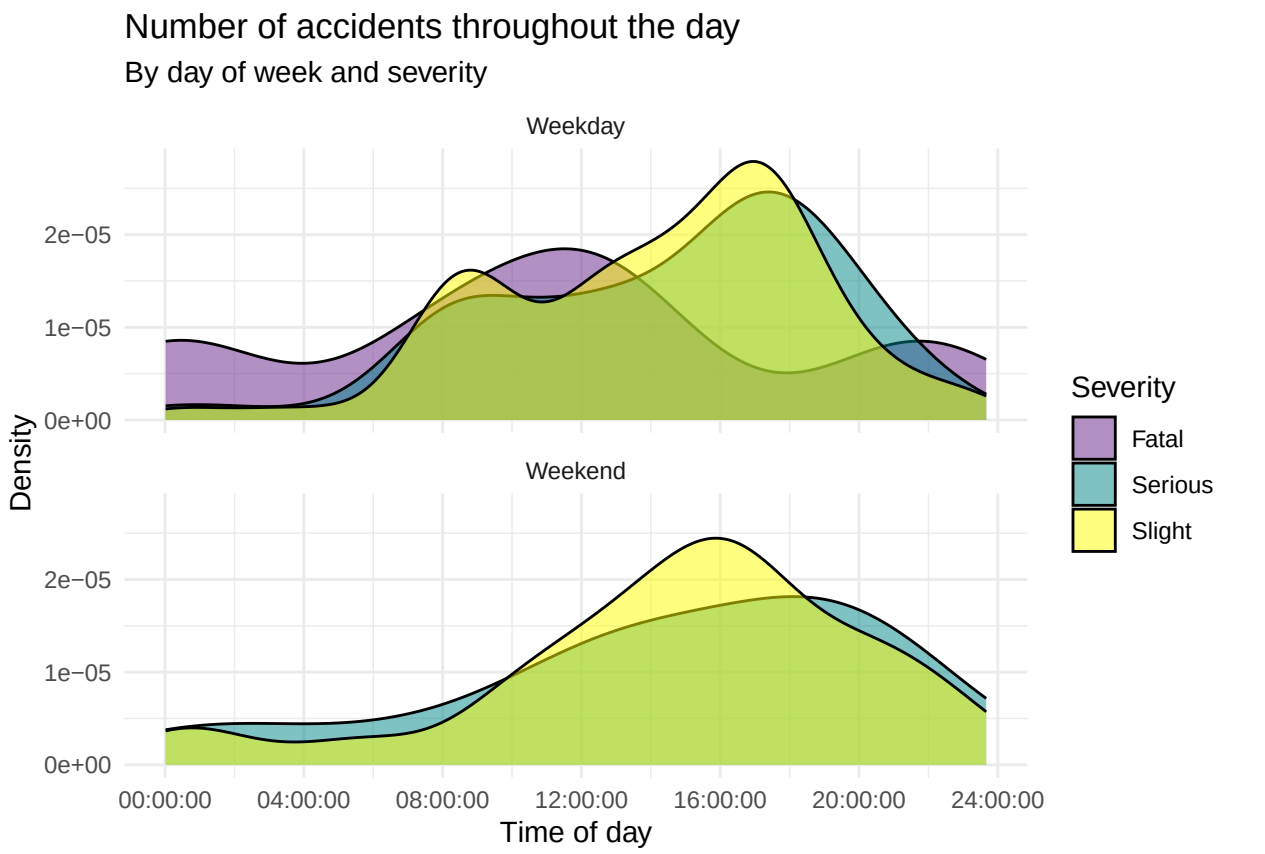
Probem 1: Road traffic injuries in Edinburgh, Scotland

The dataset can be found in the `dsbox` package, and is called `accidents`. It covers all recorded accidents in Edinburgh in 2018; compared to the dataset made available by the UK government, some of the variables were modified for the purposes of the package. You can find out more about the dataset by inspecting its documentation with `?accidents`. Recreate the following plot, and interpret the results.

```
accidents2 <- mutate(accidents, daytype = ifelse(day_of_week %in% c("Monday", "Tuesday", "Wednesday", "

ggplot(data = accidents2, mapping = aes(x = time, fill = severity)) +
  geom_density(alpha = 0.5) + scale_fill_manual(values=c("darkorchid4", "turquoise4", "yellow")) +
```

```
  labs(fill = "Severity",
       y = "Density",
       x = "Time of day",
       title = "Number of accidents throughout the day",
       subtitle = "By day of week and severity") +
  facet_wrap(~ daytype, ncol = 1) +
  theme_minimal()
```



Problem 2: One Dataset, Visualized ~~25~~ 5 Ways

Inspired by Nathan Yau's One Dataset, Visualized 25 Ways, I want you to create 5 visualizations of the same data. You can use the `mpg` dataset or another dataset of your choosing, including the `accidents` dataset above. Make sure you have the data manual open for this problem!

a. Pick 3 - 4 variables you want to explore. Provide their code names here.

b. Create 5 graphs. A few things to consider:

- Like Nathan's graphs, they don't all have to contain every one of your selected variables.
- You can't use the same `geom` for all four graphs but you can use the same `geom` more than once.
- Think carefully about color, the coordinate system, and scales.
- Feel free to subset or wrangling the dataset if you want to but it isn't required.

c. Discuss the pros/cons of your graphs. What useful information can be gleaned? How do the different geoms and aesthetics impact the story?

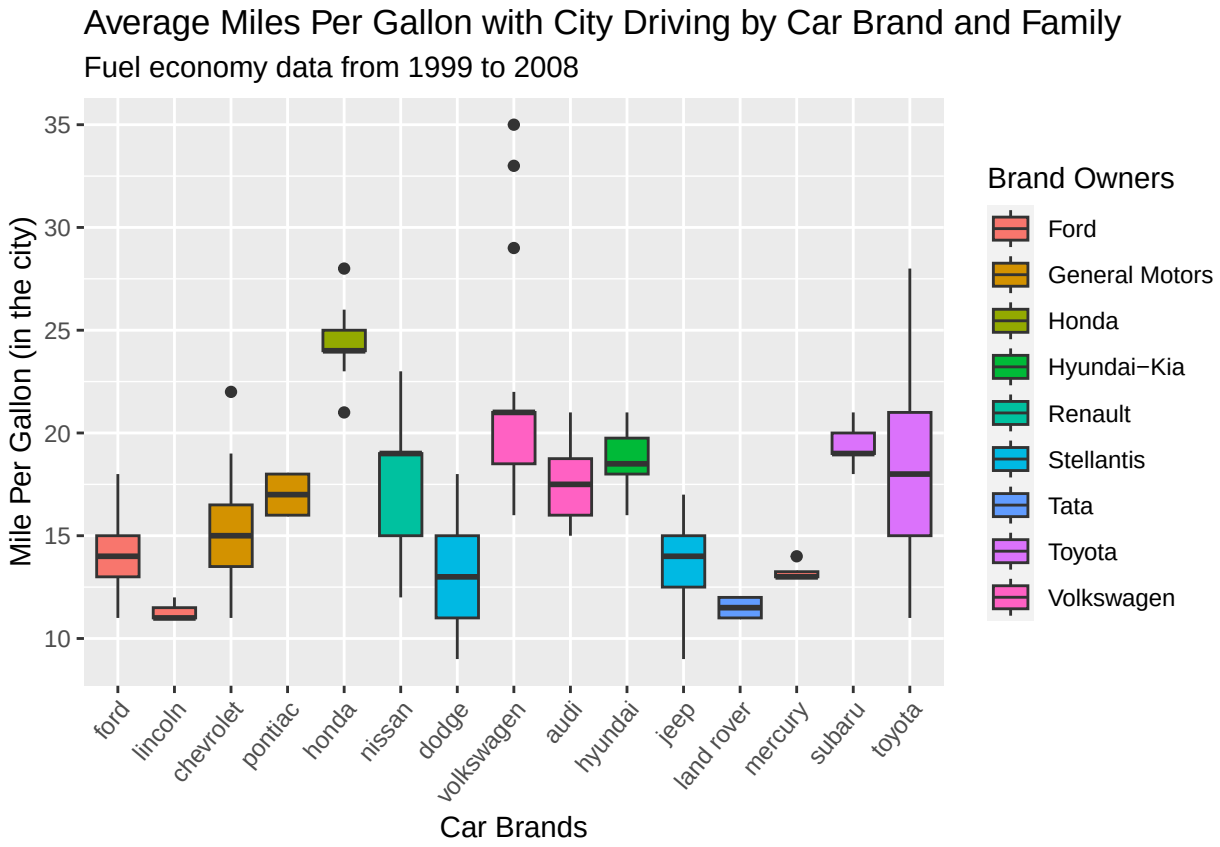
---

a. I'll be using cyl, manufacturer, cty, and drv.

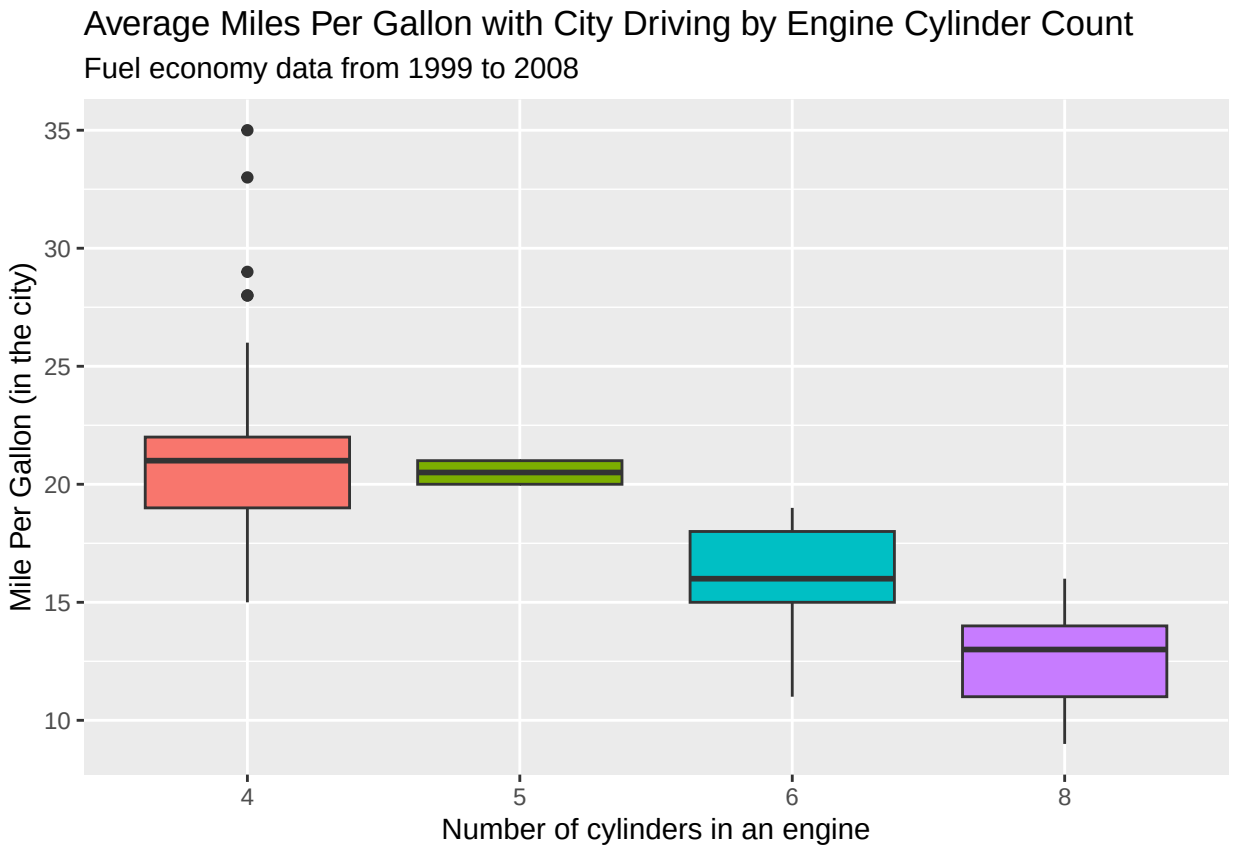
b.

```
mpg2 <- mutate(mpg, family = case_when(manufacturer %in% c("volkswagen", "audi") ~ "Volkswagen",
                                       manufacturer %in% c("dodge", "jeep") ~ "Stellantis",
                                       manufacturer %in% c("subaru", "lexus", "toyota") ~ "Toyota",
                                       manufacturer %in% c("lincoln", "ford", "mercury") ~ "Ford",
                                       manufacturer %in% c("land rover") ~ "Tata",
                                       manufacturer %in% c("nissan") ~ "Renault",
                                       manufacturer %in% c("hyundai", "kia") ~ "Hyundai-Kia",
                                       manufacturer %in% c("chevrolet", "pontiac") ~ "General Motors",
                                       manufacturer %in% c("honda") ~ "Honda"))
```

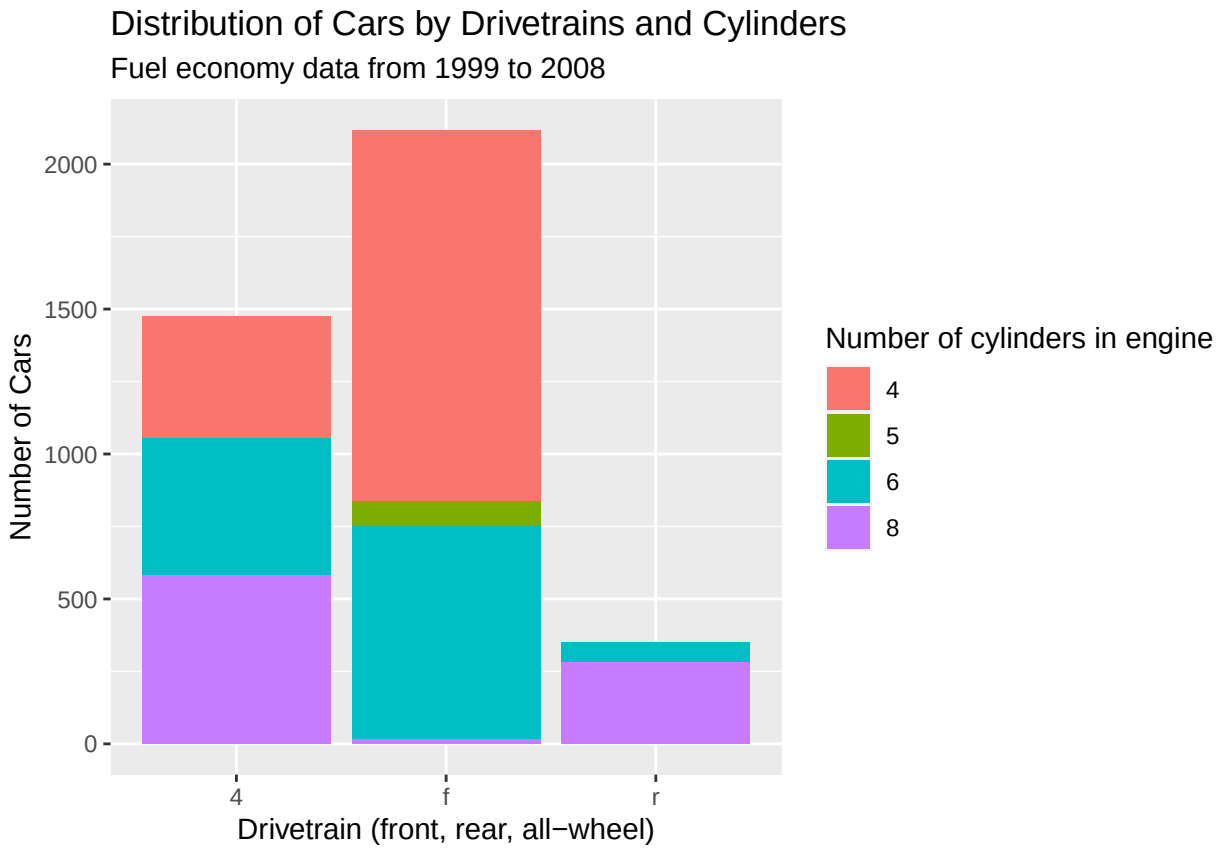
```
ggplot(data = mpg2, mapping = aes(x=fct_reorder(manufacturer, family), y = cty, fill = family)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 50, vjust = 1, hjust= 1)) +
  labs(title = "Average Miles Per Gallon with City Driving by Car Brand and Family",
       subtitle = "Fuel economy data from 1999 to 2008",
       x = "Car Brands",
       y = "Mile Per Gallon (in the city)",
       fill = "Brand Owners")
```



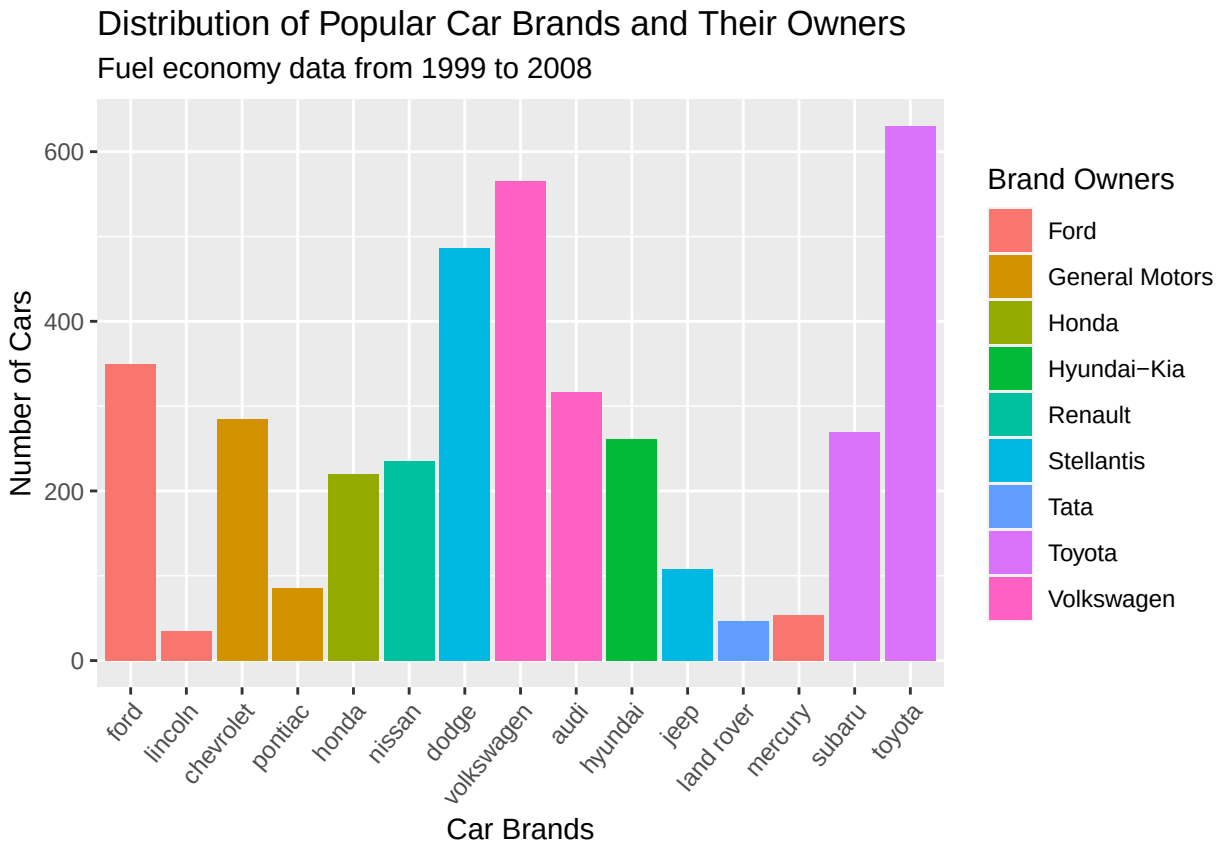

```
ggplot(data = mpg2, mapping = aes(x=as.character(cyl), y = cty, fill = as.character(cyl))) +
  geom_boxplot() +
  theme(legend.position="none") +
  labs(title = "Average Miles Per Gallon with City Driving by Engine Cylinder Count",
       subtitle = "Fuel economy data from 1999 to 2008",
       x = "Number of cylinders in an engine",
       y = "Mile Per Gallon (in the city)")
```



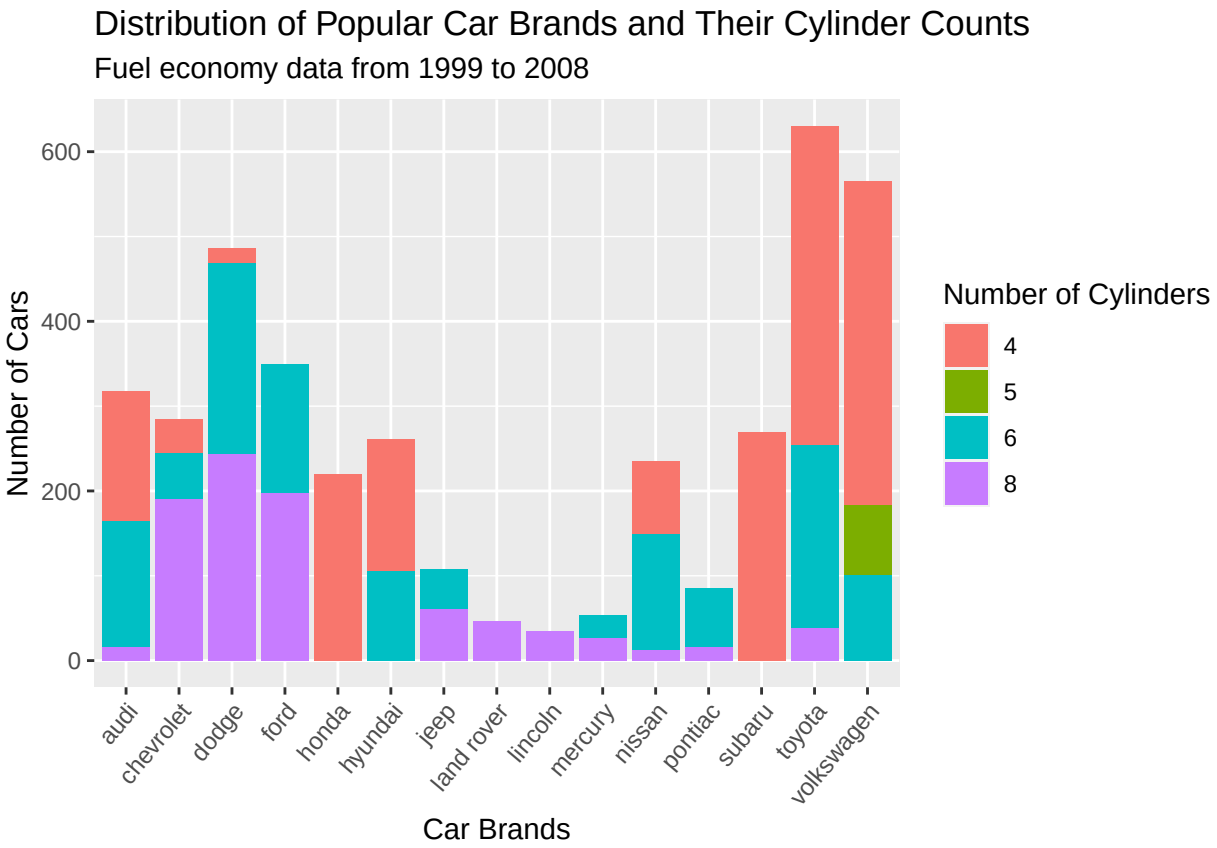

```
ggplot(data = mpg2, mapping = aes(x=drv, y = cty, fill = as.character(cyl))) +
  geom_col() +
  labs(title = "Distribution of Cars by Drivetrains and Cylinders",
       subtitle = "Fuel economy data from 1999 to 2008",
       x = "Drivetrain (front, rear, all-wheel)",
       y = "Number of Cars",
       fill = "Number of cylinders in engine")
```




```
ggplot(data = mpg2, mapping = aes(x=fct_reorder(manufacturer, family), y = cty, fill = family)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 50, vjust = 1, hjust= 1)) +
  labs(title = "Distribution of Popular Car Brands and Their Owners",
       subtitle = "Fuel economy data from 1999 to 2008",
       x = "Car Brands",
       y = "Number of Cars",
       fill = "Brand Owners")
```




```
ggplot(data = mpg2, mapping = aes(x=manufacturer, y = cty, fill = as.character(cyl))) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 50, vjust = 1, hjust= 1)) +
  labs(title = "Distribution of Popular Car Brands and Their Cylinder Counts",
       subtitle = "Fuel economy data from 1999 to 2008",
       x = "Car Brands",
       y = "Number of Cars",
       fill = "Number of Cylinders")
```



c. My graphs are able to show some relationships between some variables (like car brand and mpg) and allows the reader to make conclusions about the relationships on their own. The colors and how it is displayed also allows for clear distinctions between the different variables. However, the graphs are not telling an interestingly different story in some of them, even though it's the same data set (perhaps it's harder due to how smaller this data set is?).

Problem 3: Style This Code!

Take the following code and don't change its functionality but DO change its style. Use the Tidyverse Style Guide!

```
animal_weight = data.frame(weight = c(runif(3),NA), y = c("cat","mouse","dog","rat"))
median(animal_weight$weight, TRUE);
```

```
## [1] 0.3035805
```

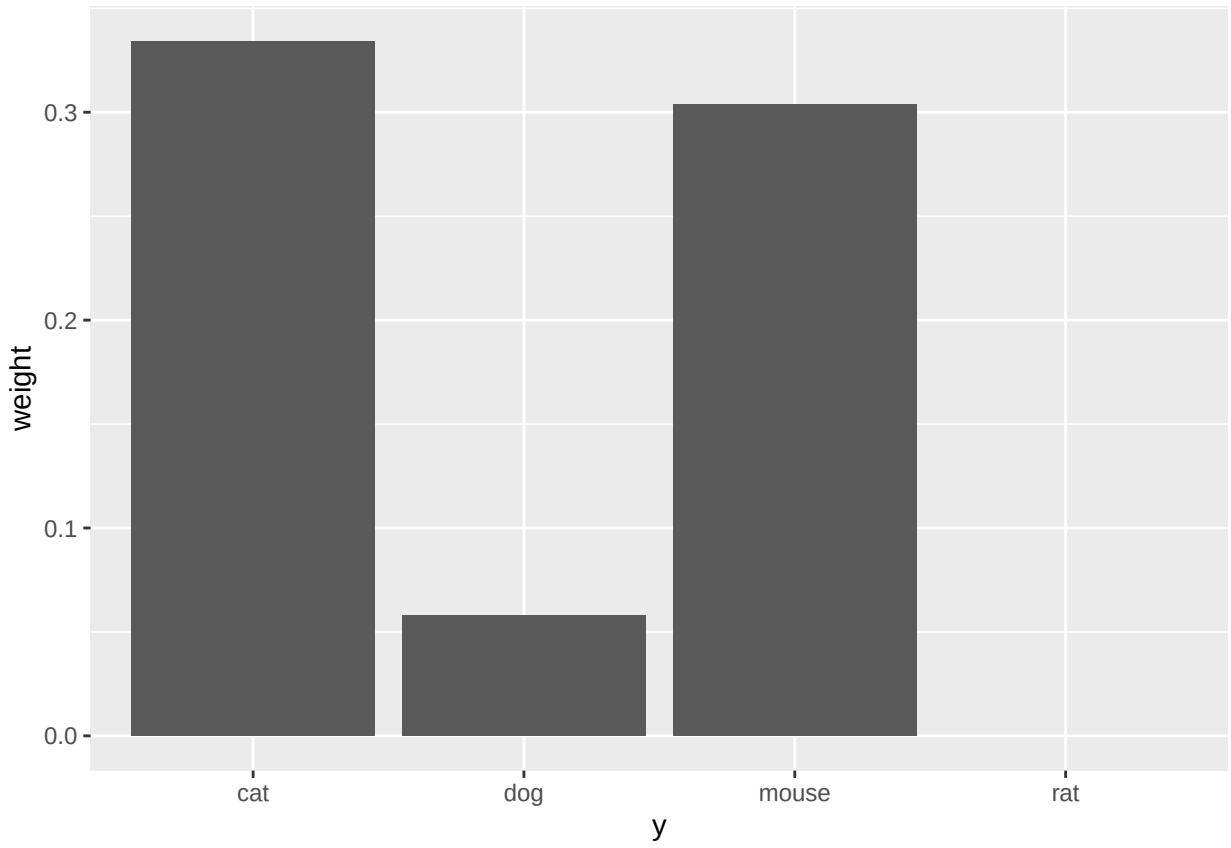
```
mean(animal_weight$weight, 0 , TRUE);
```

```
## [1] 0.231998
```

```
var(animal_weight$weight, NULL, TRUE)
```

```
## [1] 0.02290125
```

```
ggplot(animal_weight, aes(y = weight, x = y)) +
  geom_col() +
  scale_y_continuous()
```

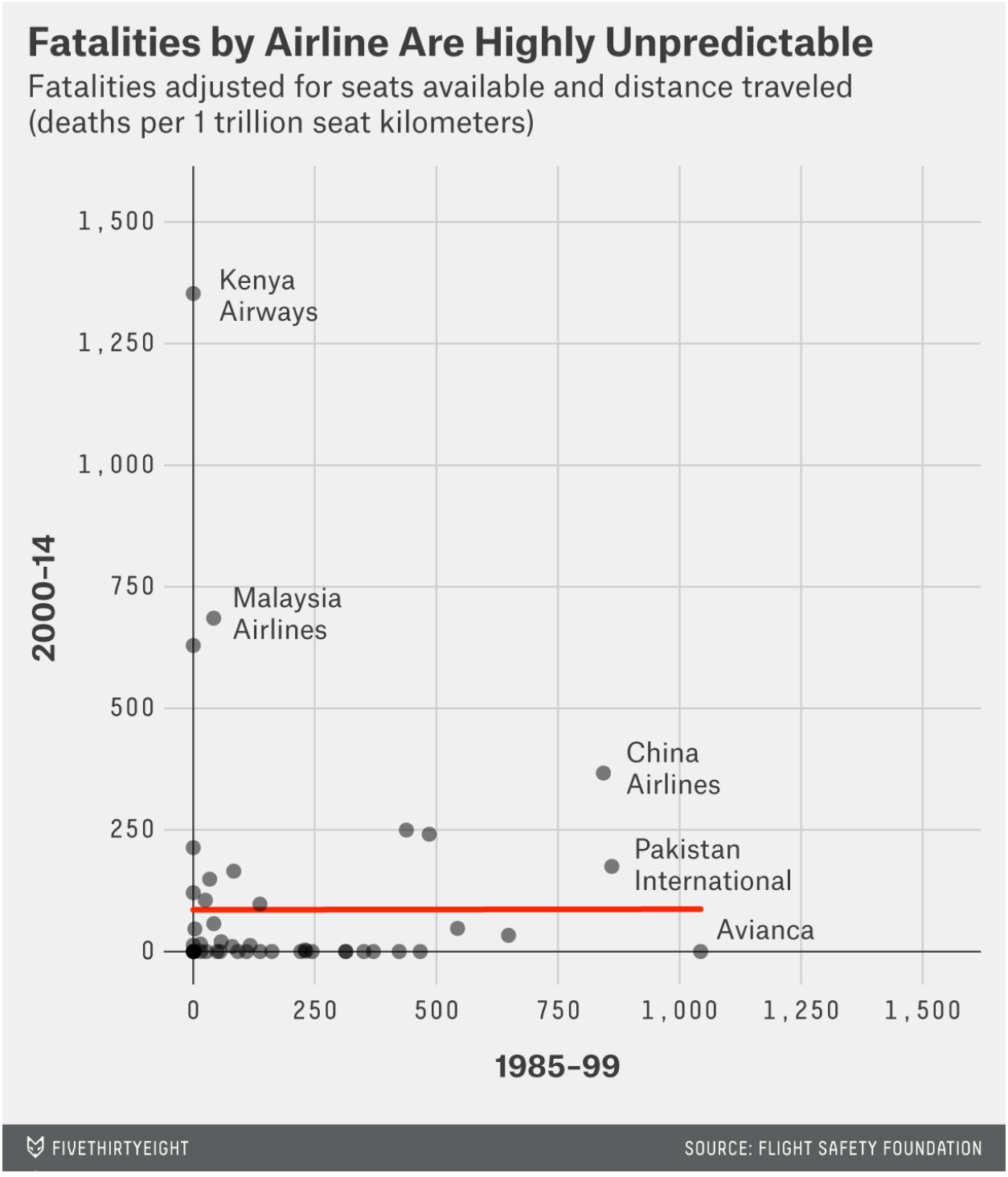


Problem 4: Imitation is the Sincerest Form of Flattery

For this problem, I want you to try to recreate a FiveThirtyEight.com graphic. Awesomely, they share their data with the world here. (Note: You don't need to recreate all their branding/background color scheme.)

a. Take a screenshot of the graph, upload it to the same folder on the server where you have saved your lab, and insert the file name below. Then change the `eval = FALSE` to `eval = TRUE`.

```
knitr::include_graphics("~/Github/math241/labs/lab2pic.png")
```

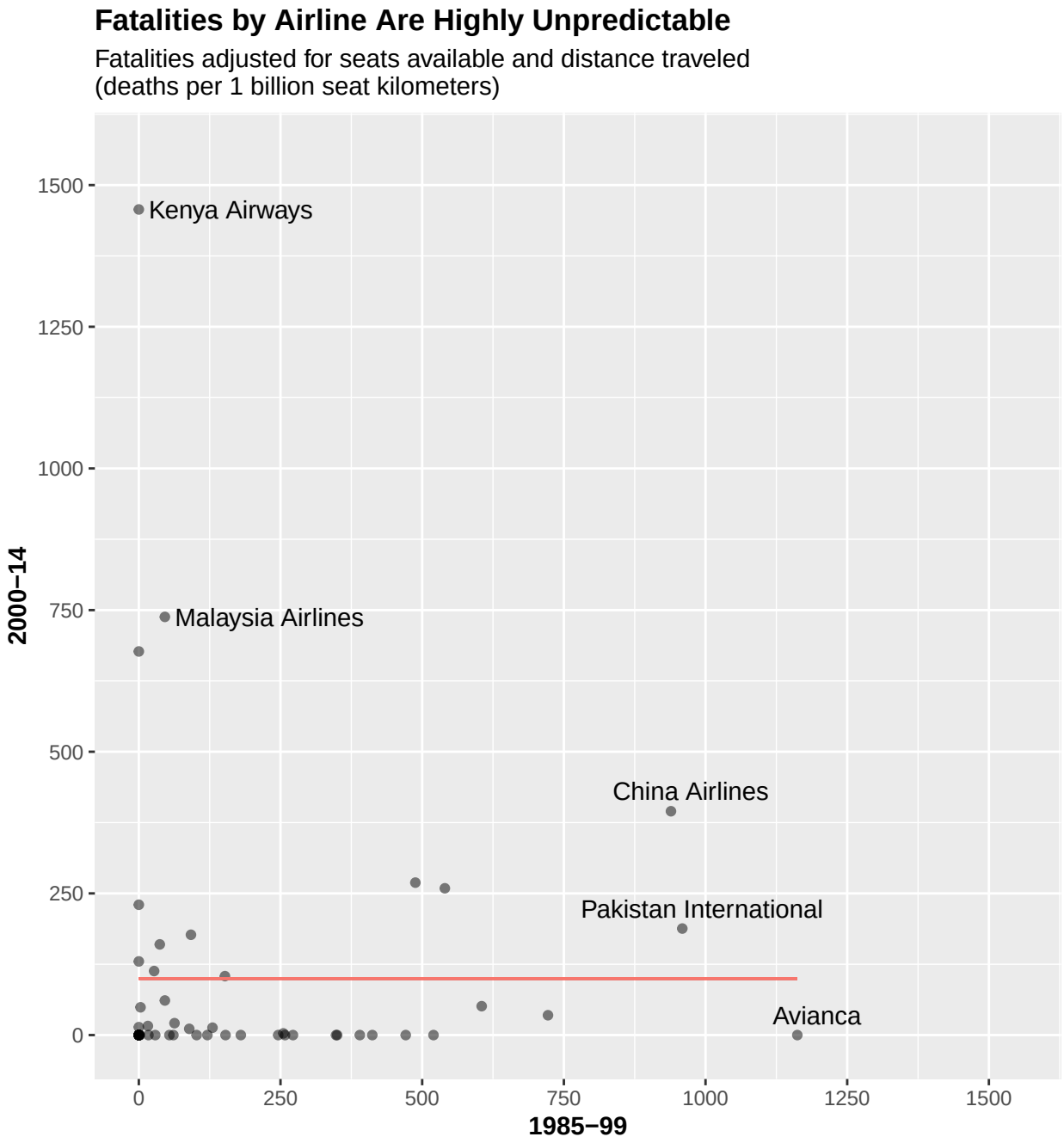


b. Load the data and recreate the graph as best as you can.

```
airline_safety <- read_csv("~/Github/math241/labs/airline-safety.csv")

airline_safety <- mutate(airline_safety,
                         fatalities_per_1t_85_99 = as.integer((fatalities_85_99 / (avail_seat_km_per_we
                         fatalities_per_1t_00_14 = as.integer((fatalities_00_14 / (avail_seat_km_per_we
```

```r
ggplot(data = airline_safety, mapping = aes(x = fatalities_per_1t_85_99, y = fatalities_per_1t_00_14, la
  geom_point(alpha = 0.5) +
  geom_text_repel(data = subset(airline_safety, airline %in% c("Kenya Airways", "Malaysia Airlines", "Cl
                  aes(label= airline)) +
  geom_segment(aes(x = 0, y = 100, xend = 1162, yend = 100, color = "red")) +
  scale_x_continuous(name="1985-99", limits=c(0, 1550), breaks=c(0, 250, 500, 750, 1000, 1250, 1500)) +
  scale_y_continuous(name="2000-14", limits=c(0, 1550), breaks = c(0, 250, 500, 750, 1000, 1250, 1500))
  labs(title = "Fatalities by Airline Are Highly Unpredictable",
       subtitle = "Fatalities adjusted for seats available and distance traveled\n(deaths per 1 billion
  theme(aspect.ratio = 1,
        plot.title = element_text(hjust = 0, face = "bold"),
        plot.subtitle = element_text(hjust = 0),
        axis.title = element_text(face = "bold"),
        legend.position="none")
```



c. Now make the graph better somehow.

I made the data a little more accurate, and also made the names listed on the graph not nudge the point, like it does in the graph. I also corrected a mistake they did where they said it was deaths per 1 trillion seat kilometers, when in reality it was billion not trillion.

d. Justify why your rendition of this `FiveThirtyEight.com` graph is more effective at telling the data story than the original.

It's more effective due to slightly more accurate points due to not nudging points and also has a more accurate subtext that would lessen the dramatization of the insuiuated safety. I'm not sure whether it was intentional or not, but they had a big mistake where they said the unit was in trillions, while it was in billions, and my rendition is correct.

Problem 5: Rental apartments in SF

The data for this exercise comes from `TidyTuesday`, and is on rental prices in San Francisco. You can find out more about the dataset by inspecting its documentation here. The dataset you'll be using is called `rent`. Create a visualization that will help you compare the distribution of rental prices (`price`) per bedroom (`beds`) across neighborhoods (`nhood`) in the city of San Francisco (`city == "san francisco"`), over time.

Limit your analysis to rentals where the full unit is available, i.e. (`room_in_apt == 0`). You have the flexibility to choose which years and which neighborhoods. Note that you should have a maximum of 8 neighborhoods on your visualization, but one or more of them can be a combination of many (e.g., an "other" category). Your visualization should also display some measure of the variability in your data. You get to decide what type of visualization to create and there is more than one correct answer! In your answer, include a brief description of why you made the choices you made as well as an interpretation of the findings of how rental prices vary over time and neighborhoods in San Francisco.

```
# Get the Data

# Read in with tidytuesdayR package
# Install from CRAN via: install.packages("tidytuesdayR")
# This loads the readme and all the datasets for the week of interest

library(tidytuesdayR)
tuesdata <- tidytuesdayR::tt_load('2022-07-05') # this could take a minute
```

```
##
##  Downloading file 1 of 3: `rent.csv`
##  Downloading file 2 of 3: `sf_permits.csv`
##  Downloading file 3 of 3: `new_construction.csv`
```

```
rent <- tuesdata$rent
```

```
rent_data <- filter(tuesdata$rent, city == "san francisco", room_in_apt == 0)
rent_data <- filter(rent_data, nhood %in% (rent_data %>% group_by(nhood) %>% summarize(n = n()) %>% top

ggplot(data = rent_data, mapping = aes(color = as.character(beds), x = price, shape = as.character(year
  geom_point(position = "jitter", alpha = 0.5) +
  scale_x_continuous(trans = "log10") + labs(
    title = "Distribution of housing by beds, price, top neighborhoods\nover time in San Francisco",
    color = "Number of Beds",
    shape = "Year",
    x = "Price (USD)",
    y = "Neighboorhood"
  )
```

I have made things look visible in a way that shows trends, rather than individual points. With this many data points, I believe the focus should be that someone can gather some information about housing trends, which is clear with the price and number of beds relationship, and also the shapes that are visible should give a sense of what year these numbers are from. We can see that within this era, not much has changed, but most look pretty expensive regardless.