Final Project: Used Cars for Sale

Introduction

This project explores a dataset about cars for sale taken from the website TrueCar.com on September 24th, 2017. The dataset, which I downloaded from kaggle.com, includes roughly 1.2 million rows of cars' make, model, year, mileage, listing price, VIN, and the city and state in which the car was for sale.

While the original purpose of this investigation was to determine which cars were worth the price tag, it quickly turned into an exploration of determining model popularity, the trends of location, the average age, mileage, and price of models, and the distribution of electric vehicles throughout the country and state. By studying this data, consumers can inform themselves on which type of vehicle might be best fit for the area in which they live, how rare a model may be that they are looking for, finding which variation of model they like may be least expensive, give them a general idea about the average price, age, and mileage of cars they're interested in, and more. Dealerships may also be able to benefit from this data by finding which models and variations are most popular as well as what areas need more cars than others.

Load Packages

Each of the following packages allows me to easily work with the data and create graphics that display my findings in an interesting, informative, and effective way. I added the "options(scipen = 999)" to ensure that my data did not to revert to scientific notation.

```
knitr::opts_chunk$set(error = TRUE)
options(tigris_use_cache = TRUE)

library(tidyverse)
library(plotly)
library(treemapify)
library(tigris)
library(forcats)
library(maps)
library(cowplot)
options(scipen = 999)
```

Load the Data

The following datasets were loaded into RStudio to complete my project. The cars dataset is the one aforementioned, while the regions dataset includes the full name of the states in the US, their abbreviations, and which region the state is a part of (south, west, northeast, or midwest).

```
cars <- read_csv("data/tc20171021.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
```

```
##   Id = col_double(),
##   Price = col_double(),
##   Year = col_double(),
##   Mileage = col_double(),
##   City = col_character(),
##   State = col_character(),
##   Vin = col_character(),
##   Make = col_character(),
##   Model = col_character()
## )
```

```
## Warning: 16792 parsing failures.
##    row col  expected     actual                  file
## 297910  -- 9 columns 10 columns 'data/tc20171021.csv'
## 297911  -- 9 columns 10 columns 'data/tc20171021.csv'
## 297912  -- 9 columns 10 columns 'data/tc20171021.csv'
## 297913  -- 9 columns 10 columns 'data/tc20171021.csv'
## 297914  -- 9 columns 10 columns 'data/tc20171021.csv'
## ...... ... ......... .......... ......................
## See problems(...) for more details.
```

```
regions <- read_rds("data/states (1).rds")
```

Tidy the Data

Since the dataset I wanted to use was so large, I decided to limit my project to the focus only on the top ten most popular makes. In order, this turns out to be: Ford, Chevy, Toyota, Nissan, Honda, Jeep, Hyundai, Dodge, GMC, and Kia. After narrowing down my data, one of the biggest hurdles I faced was determining how to clean it up. I had many rows where the model type was grouped with a descriptor. For example, there were cars like a Nissan Altima4dr, or a Ford F-150FWD. While this information did qualify as the car's model, I wanted the model information to be split into two separate columns: model and and model variation. This way, I could search through Nissan Altimas or Ford F-150s and not worry about how many doors it had, what type of drive it was, whether it was luxury or not, etc.

To fix this issue, I used both RStudio and Excel. To create a new, tidy dataset for each make I had to manually input how many letters were in each model's name. In these new datasets I also changed the missing model variations become *NA*.

```
cars1 <- cars %>%
  filter(Make == "Ford")

cars2 <- cars1 %>%
  mutate(Model_part = str_sub(Model, 1, 7)) %>%
  group_by(Make, Model_part) %>%
  summarise(Models = str_c(unique(Model), collapse = ","))
```

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

```
write_csv(cars2, "cars2Ford.csv")

cars3 <- read_csv("data/first_word /cars2Ford_first_word.csv")
```

```
## Warning: Missing column names filled in: 'X4' [4], 'X5' [5], 'X6' [6], 'X7' [7],
## 'X8' [8]
```

```
##
```

```
## -- Column specification --------------------------------------------------------
## cols(
##   Make = col_character(),
##   Model_part = col_character(),
##   Models = col_character(),
##   X4 = col_logical(),
##   X5 = col_logical(),
##   X6 = col_logical(),
##   X7 = col_logical(),
##   X8 = col_logical(),
##   first_word = col_double()
## )
```

```
cars4 <- cars3 %>%
  separate_rows(Models, sep = ",") %>%
  select(Model= Models, first_word)

cars5 <- cars1 %>%
  left_join(cars4, by = "Model")

Ford <- cars5 %>%
  mutate(
    Model_Type = str_sub(Model, 1, first_word),
    Model_Variation = str_sub(Model, (first_word+1), -1)
  )
Ford[Ford == ""] <- NA
```

```
cars1 <- cars %>%
  filter(Make == "Chevrolet")

cars2 <- cars1 %>%
  mutate(Model_part = str_sub(Model, 1, 4)) %>%
  group_by(Make, Model_part) %>%
  summarise(Models = str_c(unique(Model), collapse = ","))
```

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

```
write_csv(cars2, "cars2Chevy.csv")

cars3 <- read_csv("data/first_word /cars2Chevy_first_word.csv")
```

```
## Warning: Missing column names filled in: 'X4' [4], 'X5' [5], 'X6' [6], 'X7' [7],
## 'X8' [8], 'X9' [9]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Make = col_character(),
##   Model_part = col_character(),
##   Models = col_character(),
##   X4 = col_logical(),
##   X5 = col_logical(),
##   X6 = col_logical(),
##   X7 = col_logical(),
##   X8 = col_logical(),
##   X9 = col_logical(),
```

```
##   first_word = col_double()
## )
```

```
cars4 <- cars3 %>%
  separate_rows(Models, sep = ",") %>%
  select(Model= Models, first_word)

cars5 <- cars1 %>%
  left_join(cars4, by = "Model")

Chevy <- cars5 %>%
  mutate(
    Model_Type = str_sub(Model, 1, first_word),
    Model_Variation = str_sub(Model, (first_word+1), -1)
  )
Chevy[Chevy == ""] <- NA
```

```
cars1 <- cars %>%
  filter(Make == "Toyota")

cars2 <- cars1 %>%
  mutate(Model_part = str_sub(Model, 1, 4)) %>%
  group_by(Make, Model_part) %>%
  summarise(Models = str_c(unique(Model), collapse = ","))
```

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

```
write_csv(cars2, "cars2Toyota.csv")

cars3 <- read_csv("data/first_word /cars2Toyota_first_word.csv")
```

```
## Warning: Missing column names filled in: 'X4' [4], 'X5' [5], 'X6' [6]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Make = col_character(),
##   Model_part = col_character(),
##   Models = col_character(),
##   X4 = col_logical(),
##   X5 = col_logical(),
##   X6 = col_logical(),
##   first_word = col_double()
## )
```

```
cars4 <- cars3 %>%
  separate_rows(Models, sep = ",") %>%
  select(Model= Models, first_word)

cars5 <- cars1 %>%
  left_join(cars4, by = "Model")

Toyota <- cars5 %>%
  mutate(
    Model_Type = str_sub(Model, 1, first_word),
    Model_Variation = str_sub(Model, (first_word+1), -1)
```

```
  )
Toyota[Toyota == ""] <- NA
```

```
cars1 <- cars %>%
  filter(Make == "Nissan")

cars2 <- cars1 %>%
  mutate(Model_part = str_sub(Model, 1, 3)) %>%
  group_by(Make, Model_part) %>%
  summarise(Models = str_c(unique(Model), collapse = ","))
```

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

```
write_csv(cars2, "cars2Nissan.csv")

cars3 <- read_csv("data/first_word /cars2TNissan_first_word.csv")
```

```
## Warning: Missing column names filled in: 'X4' [4], 'X5' [5], 'X6' [6]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Make = col_character(),
##   Model_part = col_character(),
##   Models = col_character(),
##   X4 = col_logical(),
##   X5 = col_logical(),
##   X6 = col_logical(),
##   first_word = col_double()
## )
```

```
cars4 <- cars3 %>%
  separate_rows(Models, sep = ",") %>%
  select(Model= Models, first_word)

cars5 <- cars1 %>%
  left_join(cars4, by = "Model")

Nissan <- cars5 %>%
  mutate(
    Model_Type = str_sub(Model, 1, first_word),
    Model_Variation = str_sub(Model, (first_word+1), -1)
  )
Nissan[Nissan == ""] <- NA
```

```
cars1 <- cars %>%
  filter(Make == "Honda")

cars2 <- cars1 %>%
  mutate(Model_part = str_sub(Model, 1, 3)) %>%
  group_by(Make, Model_part) %>%
  summarise(Models = str_c(unique(Model), collapse = ","))
```

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

```
write_csv(cars2, "cars2Honda.csv")
```

```
cars3 <- read_csv("data/first_word /cars2Honda_first_word.csv")
```

```
## Warning: Missing column names filled in: 'X4' [4], 'X5' [5]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Make = col_character(),
##   Model_part = col_character(),
##   Models = col_character(),
##   X4 = col_logical(),
##   X5 = col_logical(),
##   first_word = col_double()
## )
```

```
cars4 <- cars3 %>%
  separate_rows(Models, sep = ",") %>%
  select(Model= Models, first_word)

cars5 <- cars1 %>%
  left_join(cars4, by = "Model")

Honda <- cars5 %>%
  mutate(
    Model_Type = str_sub(Model, 1, first_word),
    Model_Variation = str_sub(Model, (first_word+1), -1)
  )
Honda[Honda == ""] <- NA
```

```
cars1 <- cars %>%
  filter(Make == "Jeep")

cars2 <- cars1 %>%
  mutate(Model_part = str_sub(Model, 1, 4)) %>%
  group_by(Make, Model_part) %>%
  summarise(Models = str_c(unique(Model), collapse = ","))
```

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

```
write_csv(cars2, "cars2Jeep.csv")

cars3 <- read_csv("data/first_word /cars2Jeep_first_word.csv")
```

```
## Warning: Missing column names filled in: 'X4' [4]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Make = col_character(),
##   Model_part = col_character(),
##   Models = col_character(),
##   X4 = col_logical(),
##   first_word = col_double()
## )
```

```
cars4 <- cars3 %>%
  separate_rows(Models, sep = ",") %>%
```

```
  select(Model= Models, first_word)

cars5 <- cars1 %>%
  left_join(cars4, by = "Model")

Jeep <- cars5 %>%
  mutate(
    Model_Type = str_sub(Model, 1, first_word),
    Model_Variation = str_sub(Model, (first_word+1), -1)
  )
Jeep[Jeep == ""] <- NA
```

```
cars1 <- cars %>%
  filter(Make == "Hyundai")

cars2 <- cars1 %>%
  mutate(Model_part = str_sub(Model, 1, 3)) %>%
  group_by(Make, Model_part) %>%
  summarise(Models = str_c(unique(Model), collapse = ","))
```

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

```
write_csv(cars2, "cars2Hyundai.csv")

cars3 <- read_csv("data/first_word /cars2Hyundai_first_word.csv")
```

```
## Warning: Missing column names filled in: 'X4' [4]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Make = col_character(),
##   Model_part = col_character(),
##   Models = col_character(),
##   X4 = col_logical(),
##   first_word = col_double()
## )
```

```
cars4 <- cars3 %>%
  separate_rows(Models, sep = ",") %>%
  select(Model= Models, first_word)

cars5 <- cars1 %>%
  left_join(cars4, by = "Model")

Hyundai <- cars5 %>%
  mutate(
    Model_Type = str_sub(Model, 1, first_word),
    Model_Variation = str_sub(Model, (first_word+1), -1)
  )
Hyundai[Hyundai == ""] <- NA
```

```
cars1 <- cars %>%
  filter(Make == "Dodge")

cars2 <- cars1 %>%
```

```
  mutate(Model_part = str_sub(Model, 1, 4)) %>%
  group_by(Make, Model_part) %>%
  summarise(Models = str_c(unique(Model), collapse = ","))
```

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

```
write_csv(cars2, "cars2Dodge.csv")

cars3 <- read_csv("data/first_word /cars2Dodge_first_word.csv")
```

```
## Warning: Missing column names filled in: 'X4' [4], 'X5' [5], 'X6' [6]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Make = col_character(),
##   Model_part = col_character(),
##   Models = col_character(),
##   X4 = col_logical(),
##   X5 = col_logical(),
##   X6 = col_logical(),
##   first_word = col_double()
## )
```

```
cars4 <- cars3 %>%
  separate_rows(Models, sep = ",") %>%
  select(Model= Models, first_word)

cars5 <- cars1 %>%
  left_join(cars4, by = "Model")

Dodge <- cars5 %>%
  mutate(
    Model_Type = str_sub(Model, 1, first_word),
    Model_Variation = str_sub(Model, (first_word+1), -1)
  )
Dodge[Dodge == ""] <- NA
```

```
cars1 <- cars %>%
  filter(Make == "GMC")

cars2 <- cars1 %>%
  mutate(Model_part = str_sub(Model, 1, 3)) %>%
  group_by(Make, Model_part) %>%
  summarise(Models = str_c(unique(Model), collapse = ","))
```

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

```
write_csv(cars2, "cars2GMC.csv")

cars3 <- read_csv("data/first_word /cars2GMC_first_word.csv")
```

```
## Warning: Missing column names filled in: 'X4' [4]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
```

```
##   Make = col_character(),
##   Model_part = col_character(),
##   Models = col_character(),
##   X4 = col_logical(),
##   first_word = col_double()
## )
```

```
cars4 <- cars3 %>%
  separate_rows(Models, sep = ",") %>%
  select(Model= Models, first_word)

cars5 <- cars1 %>%
  left_join(cars4, by = "Model")

GMC <- cars5 %>%
  mutate(
    Model_Type = str_sub(Model, 1, first_word),
    Model_Variation = str_sub(Model, (first_word+1), -1)
  )
GMC[GMC == ""] <- NA
```

```
cars1 <- cars %>%
  filter(Make == "Kia")

cars2 <- cars1 %>%
  mutate(Model_part = str_sub(Model, 1, 3)) %>%
  group_by(Make, Model_part) %>%
  summarise(Models = str_c(unique(Model), collapse = ","))
```

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

```
write_csv(cars2, "cars2Kia.csv")

cars3 <- read_csv("data/first_word /cars2Kia_first_word.csv")
```

```
## Warning: Missing column names filled in: 'X4' [4], 'X5' [5]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Make = col_character(),
##   Model_part = col_character(),
##   Models = col_character(),
##   X4 = col_logical(),
##   X5 = col_logical(),
##   first_word = col_double()
## )
```

```
cars4 <- cars3 %>%
  separate_rows(Models, sep = ",") %>%
  select(Model= Models, first_word)

cars5 <- cars1 %>%
  left_join(cars4, by = "Model")

Kia <- cars5 %>%
  mutate(
```

```
    Model_Type = str_sub(Model, 1, first_word),
    Model_Variation = str_sub(Model, (first_word+1), -1)
  )
Kia[Kia == ""] <- NA
```

Once this process was completed for each of the top ten most popular makes, I wanted to combine all of the tidied datasets to create a comprehensive one called "pretty_cars". I also wanted this pretty_cars dataset to have the region in which the car's posting was, so I joined pretty_cars to the regions dataset. There was also an issue in the inconsistency of the capitalization of the state abbreviations, so I changed the new dataset to have all uppercase state abbreviations.

```
pretty_cars <- rbind.data.frame(Ford, Chevy, Toyota, Nissan, Honda,
                                Jeep, Hyundai, Dodge, GMC, Kia)

pretty_cars <- pretty_cars %>%
  left_join(regions, by = c("State" = "state"))

pretty_cars <- pretty_cars %>%
  mutate(State = str_to_upper(State))

head(pretty_cars)
```

```
## # A tibble: 6 x 14
##      Id Price  Year Mileage City  State Vin   Make  Model first_word Model_Type
##   <dbl> <dbl> <dbl>   <dbl> <chr> <chr> <chr> <chr> <chr>      <dbl> <chr>
## 1  1200  9799  2012   37419 San ~ TX    3FAH~ Ford  Fusi~          6 Fusion
## 2  1201 13585  2016   35825 Oswe~ NY    3FA6~ Ford  Fusi~          6 Fusion
## 3  1202 11810  2015   46872 Alex~ VA    3FA6~ Ford  Fusi~          6 Fusion
## 4  1203 10925  2015   19279 Rale~ NC    1FAD~ Ford  Focu~          5 Focus
## 5  1204 12926  2016   12055 Gait~ MD    1FAD~ Ford  Focu~          5 Focus
## 6  1205 11675  2016   34392 Live~ CA    1FAD~ Ford  Focu~          5 Focus
## # ... with 3 more variables: Model_Variation <chr>, fullname <chr>,
## #   region <chr>
```

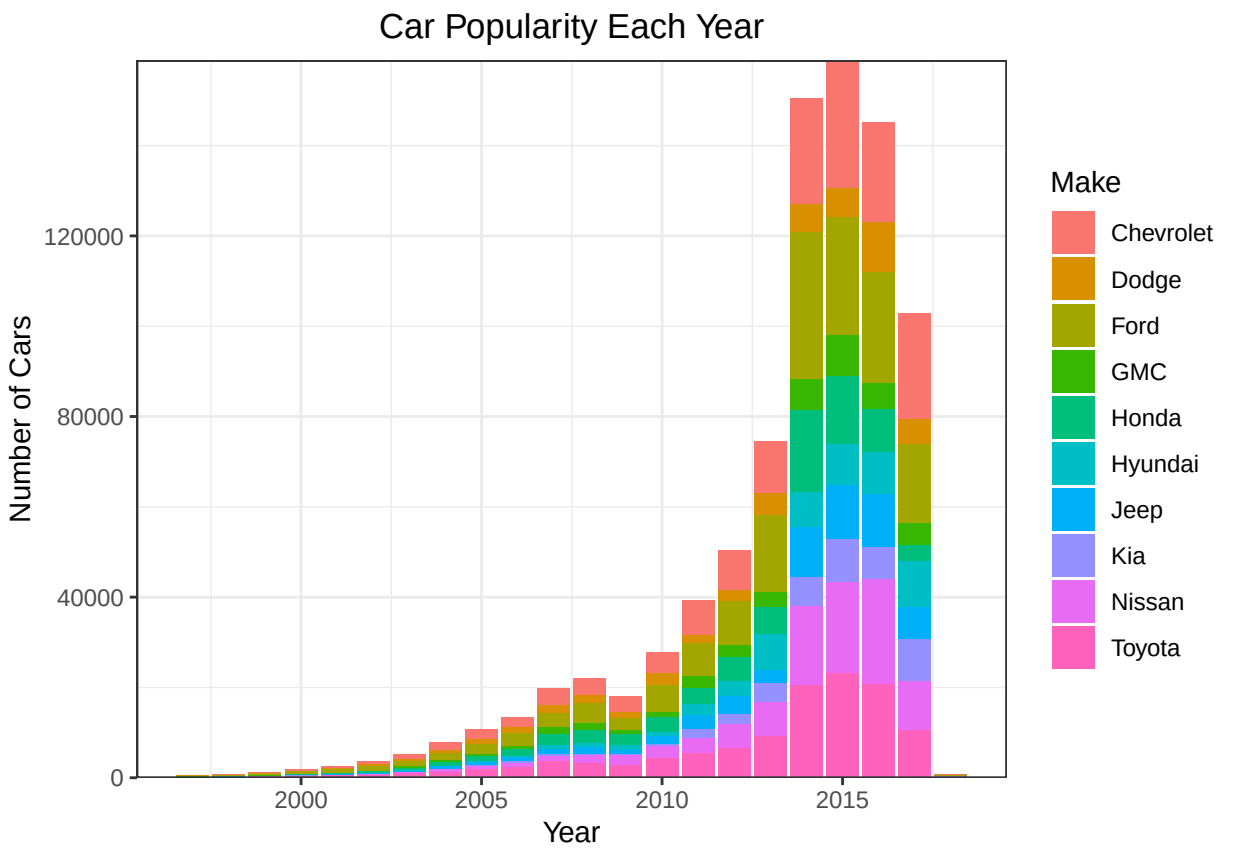
The pretty_cars dataset now has the original "Model" column as well as the new "Model_Type" and "Model_Variation". You can also see the "first_word" column represents how many letters are in the model name and how the regional information has been added.

Analyze the Data

Popularity

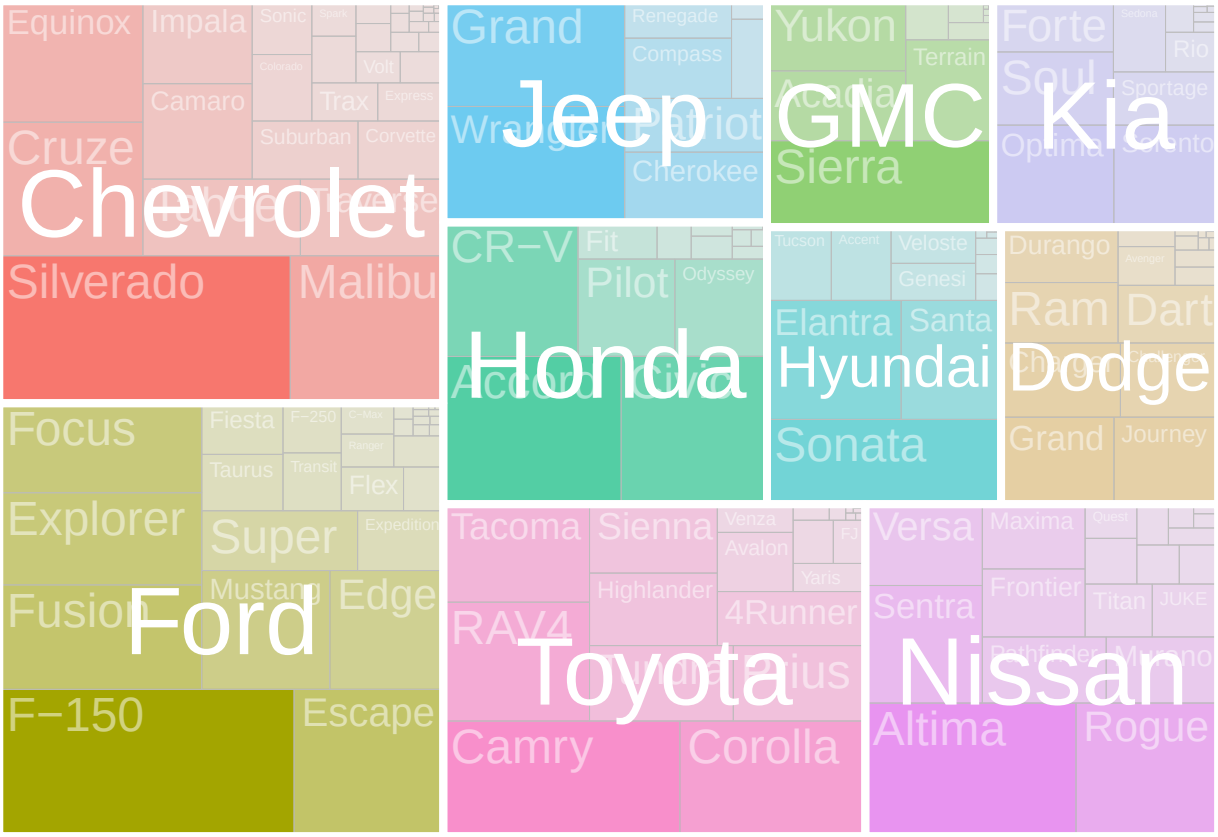
As previously mentioned, I also determined the top ten most popular makes so that I could narrow down the data I would be working with for the rest of the report. Once I established the "pretty_cars" dataset, I wanted to get an initial understanding of which makes were most popular and approximately how many cars where made in each year, so I put together this graph:

```
pretty_cars %>%
  ggplot(aes(Year, fill = Make))+
  geom_bar()+
  theme_bw()+
  scale_y_continuous(expand = c(0,0))+
  labs(title = "Car Popularity Each Year", y = "Number of Cars")+
  theme(plot.title = element_text(hjust = .5))
```



From this, we begin to see that when the dataset was taken from TrueCar.com in 2017, the ages of the cars ranged from 20 years old to brand new (since year is anywhere between 1997 and 2018). Expanding upon that idea, I was interested to find which models were most popular within each make. To show this, I created a tree map where the size of the boxes represents the number of cars in a make/model.

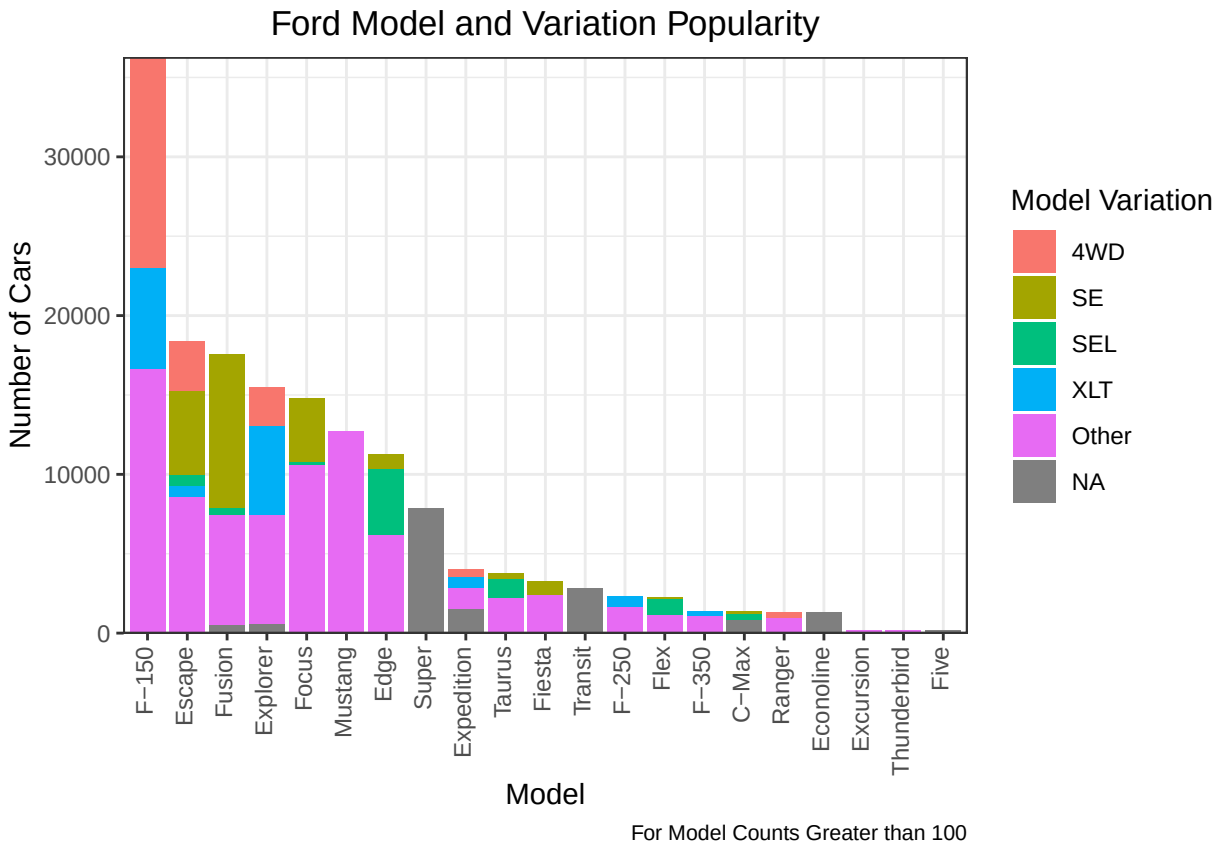
```
pretty_cars %>%
  count(Make, Model_Type) %>%
  ggplot(aes(area = n, fill = Make, subgroup = Make))+
  geom_treemap(aes(alpha = n/100))+
  geom_treemap_text(color = "white", alpha = .5, aes(label = Model_Type))+
  geom_treemap_subgroup_text(place = "center", color = "white")+
  geom_treemap_subgroup_border(color = "white")+
  theme(legend.position = "none")
```



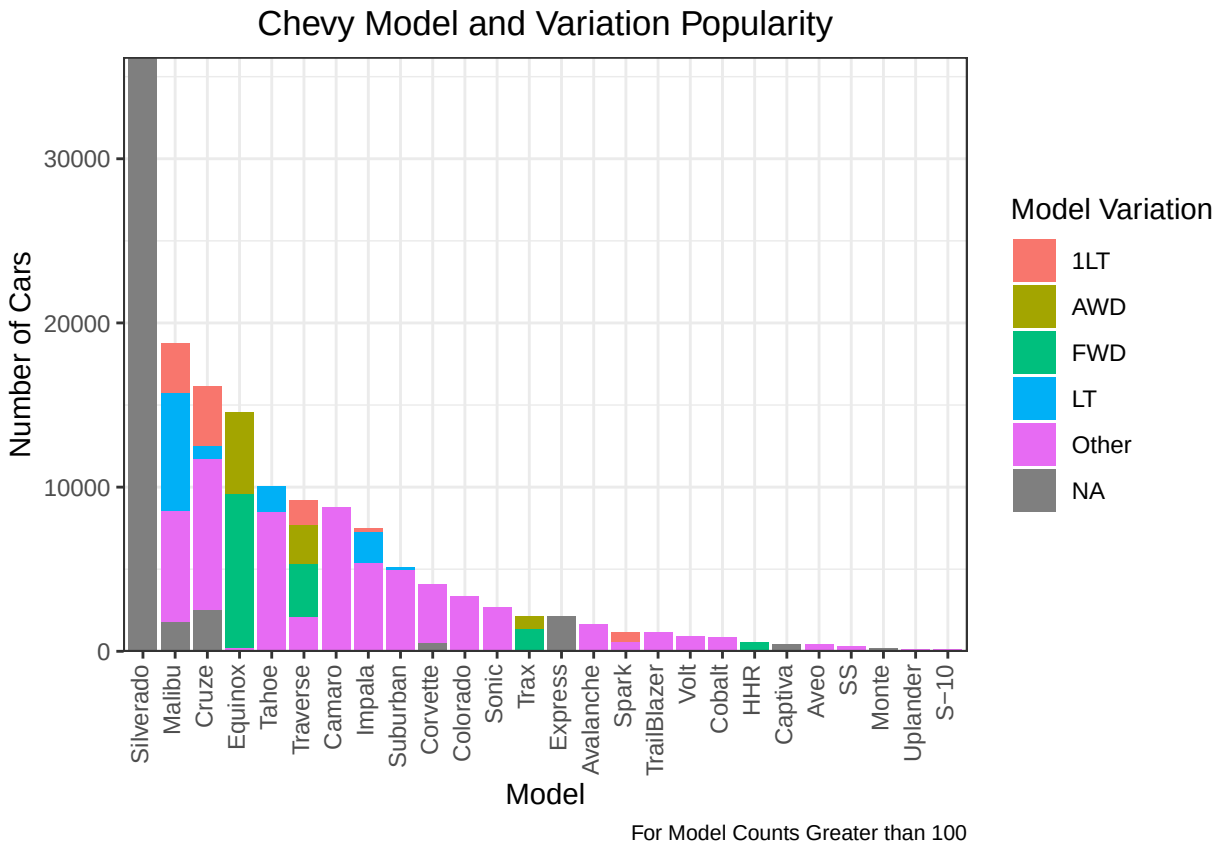
From these graphs, it was interesting to see that Ford, Chevy, and Toyota accounted for nearly half of the dataset themselves. Within the Ford and Chevy makes, F-150s and Silverado were by far the most popular models. The graphs also showed that Mustangs, Corvettes, and Camaros were decently popular models for Ford and Chevy, which is not something I expected to see.

I also wanted to visualize what variations were most popular within each model for Fords, Chevys, and Toyotas. I found that there was an excessive amount of model variation options, so I limited the graphs to show only the top 4 most popular variations and an "Other" category. Since there was a lot of different model types within each make, I also filtered the graph to only show the models which had more than 100 vehicles in the data.

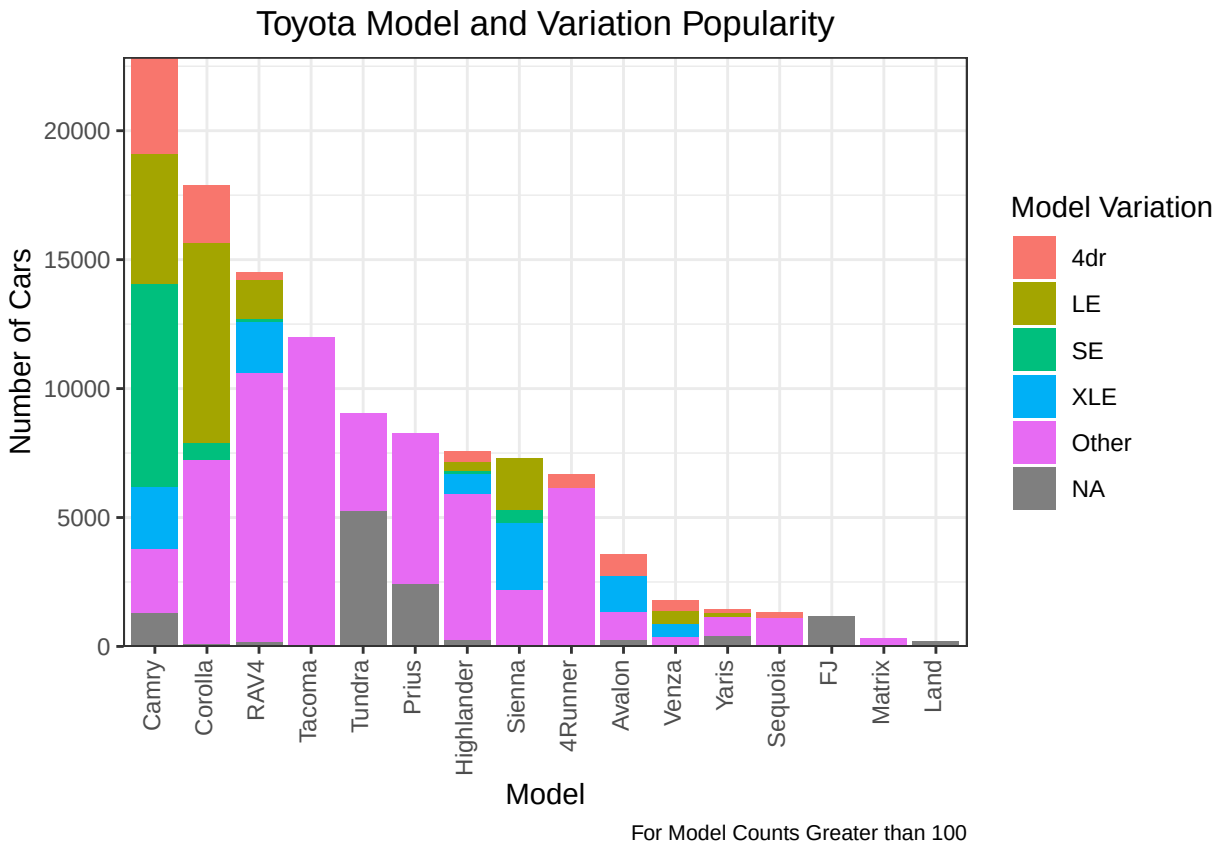
```
pretty_cars %>%
  filter(Make == "Ford") %>%
  group_by(Make) %>%
  mutate(adjusted_model_variation = fct_lump_n(Model_Variation, 4)) %>%
  count(Model, Model_Type, Model_Variation, adjusted_model_variation) %>%
  filter(n > 100) %>%
  ggplot(aes( x = reorder(Model_Type, -n, sum), y = n,  fill = adjusted_model_variation))+
  geom_col()+
  theme_bw()+
  scale_y_continuous(expand = c(0,0))+
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = .3),
        plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5),
        plot.caption = element_text(size = 8))+
  labs(title = "Ford Model and Variation Popularity", x = "Model", y = "Number of Cars",
       fill = "Model Variation", caption = "For Model Counts Greater than 100")
```




```
pretty_cars %>%
  filter(Make == "Chevrolet")%>%
  group_by(Make) %>%
  mutate(adjusted_model_variation = fct_lump_n(Model_Variation, 4)) %>%
  count(Model, Model_Type, Model_Variation, adjusted_model_variation) %>%
  filter(n > 100) %>%
  ggplot(aes( x = reorder(Model_Type, -n, sum), y = n,  fill = adjusted_model_variation))+
  geom_col()+
  theme_bw()+
  scale_y_continuous(expand = c(0,0))+
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = .3),
        plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5),
        plot.caption = element_text(size = 8))+
  labs(title = "Chevy Model and Variation Popularity", x = "Model", y = "Number of Cars",
       fill = "Model Variation", caption = "For Model Counts Greater than 100")
```




```
pretty_cars %>%
  filter(Make == "Toyota") %>%
  group_by(Make) %>%
  mutate(adjusted_model_variation = fct_lump_n(Model_Variation, 4)) %>%
  count(Model, Model_Type, Model_Variation, adjusted_model_variation) %>%
  filter(n > 100) %>%
  ggplot(aes( x = reorder(Model_Type, -n, sum), y = n,  fill = adjusted_model_variation))+
  geom_col()+
  theme_bw()+
  scale_y_continuous(expand = c(0,0))+
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = .3),
        plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5),
        plot.caption = element_text(size = 8))+
  labs(title = "Toyota Model and Variation Popularity", x = "Model", y = "Number of Cars",
       fill = "Model Variation", caption = "For Model Counts Greater than 100")
```



I also found out that while Ford F-150s had many different kinds of model variations, the Chevy Silverados had none. I did continue to make these graphs for the remaining 7 makes and found that Honda Accords, Jeep Wranglers, Jeep Grand Cherokees, Hyundai Santa Fes, Dodge Grand Caravan, Dodge Rams, and GMC Sierras had nearly no variations within the models. There was also a pattern that the type of drive (4WD, 2WD, AWD, FWD), the number of doors (4dr, 2dr), and the luxury level of the vehicles (SE, LT, LX, sport, etc.) accounted for most of the model variations. As a whole, it seemed as though larger vehicles, like trucks and SUVs, are more popular than smaller vehicles like traditional cars.

Location

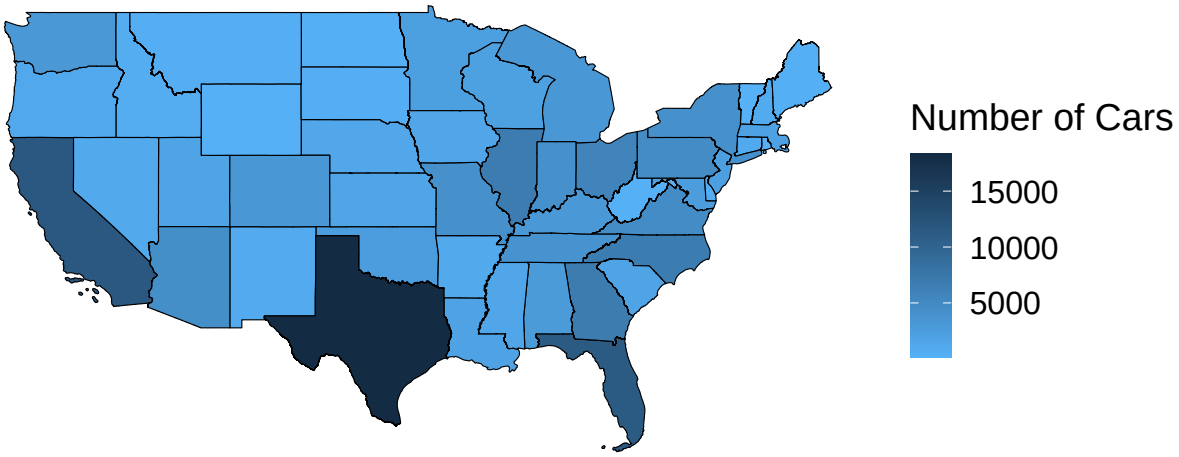
Location wasn't a factor that I was originally concerned with, but it became more prevalent as I began finding trends within the data. Because of this, I wanted to see the distribution of models across the US and the most popular models in each state and region. Since Fords, Chevys, and Toyotas were the most popular makes, I created to following graphs to show their presence in the United States. I also made a graph that displayed which model was most popular for each state, and another that showed the top 5 most popular models in each region.

```
ford_state_data <- pretty_cars %>%
  filter(Make == "Ford") %>%
  filter(!(State == "DC")) %>%
  group_by(State) %>%
  count(State)

us_states <- states(year = 2019, resolution = "20m")

ford_us_state_cont <- us_states %>%
```

```
  filter(!STATEFP %in% c("15", "78", "69", "66", "02", "60", "72")) %>%
  left_join(ford_state_data, by = c("STUSPS" = "State"))

ggplot()+
  geom_sf(
    data = ford_us_state_cont,
    aes(fill = n),
    color = "black",
    size = .2)+
  theme_map()+
  labs(title = "Popularity of Fords in the United States", fill = "Number of Cars")+
  theme(plot.title = element_text(hjust = .5))+
  scale_fill_gradient(low = "#56B1F7", high = "#132B43")
```

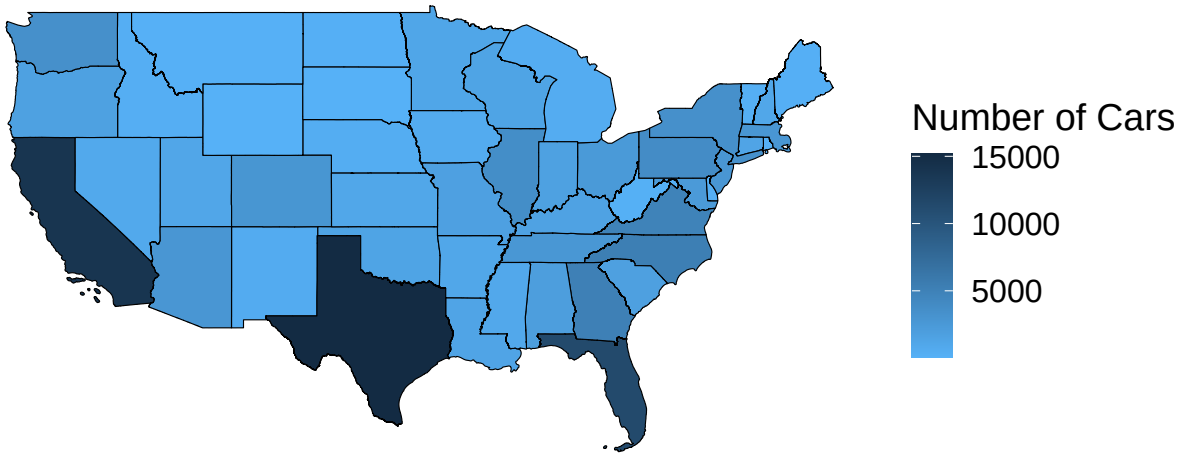

```
chevy_state_data <- pretty_cars %>%
  filter(Make == "Chevrolet") %>%
  filter(!(State == "DC")) %>%
  group_by(State) %>%
  count(State)

us_states <- states(year = 2019, resolution = "20m")

chevy_us_state_cont <- us_states %>%
  filter(!STATEFP %in% c("15", "78", "69", "66", "02", "60", "72")) %>%
  left_join(chevy_state_data, by = c("STUSPS" = "State"))

ggplot()+
  geom_sf(
    data = chevy_us_state_cont,
    aes(fill = n),
    color = "black",
    size = .2)+
  theme_map()+
  labs(title = "Popularity of Chevys in the United States", fill = "Number of Cars")+
  theme(plot.title = element_text(hjust = .5))+
  scale_fill_gradient(low = "#56B1F7", high = "#132B43")
```

Popularity of Chevys in the United States



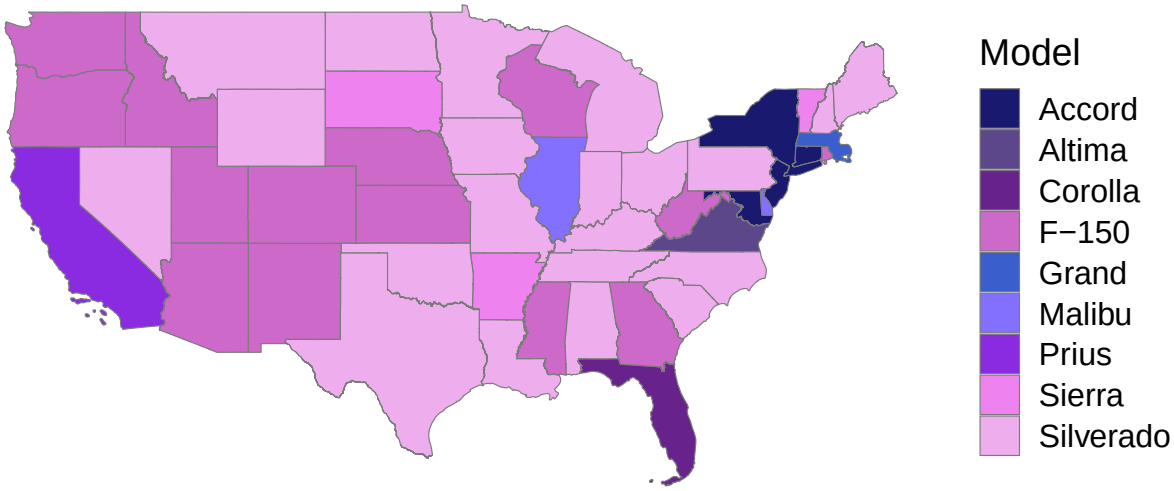

```
toyota_state_data <- pretty_cars %>%
  filter(Make == "Toyota") %>%
  filter(!(State == "DC")) %>%
  group_by(State) %>%
  count(State)

us_states <- states(year = 2019, resolution = "20m")

toyota_us_state_cont <- us_states %>%
  filter(!STATEFP %in% c("15", "78", "69", "66", "02", "60", "72")) %>%
  left_join(toyota_state_data, by = c("STUSPS" = "State"))

ggplot()+
  geom_sf(
    data = toyota_us_state_cont,
    aes(fill = n),
    color = "black",
    size = .2)+
  theme_map()+
  labs(title = "Popularity of Toyotas in the United States", fill = "Number of Cars")+
  theme(plot.title = element_text(hjust = .5))+
  scale_fill_gradient(low = "#56B1F7", high = "#132B43")
```

Popularity of Toyotas in the United States

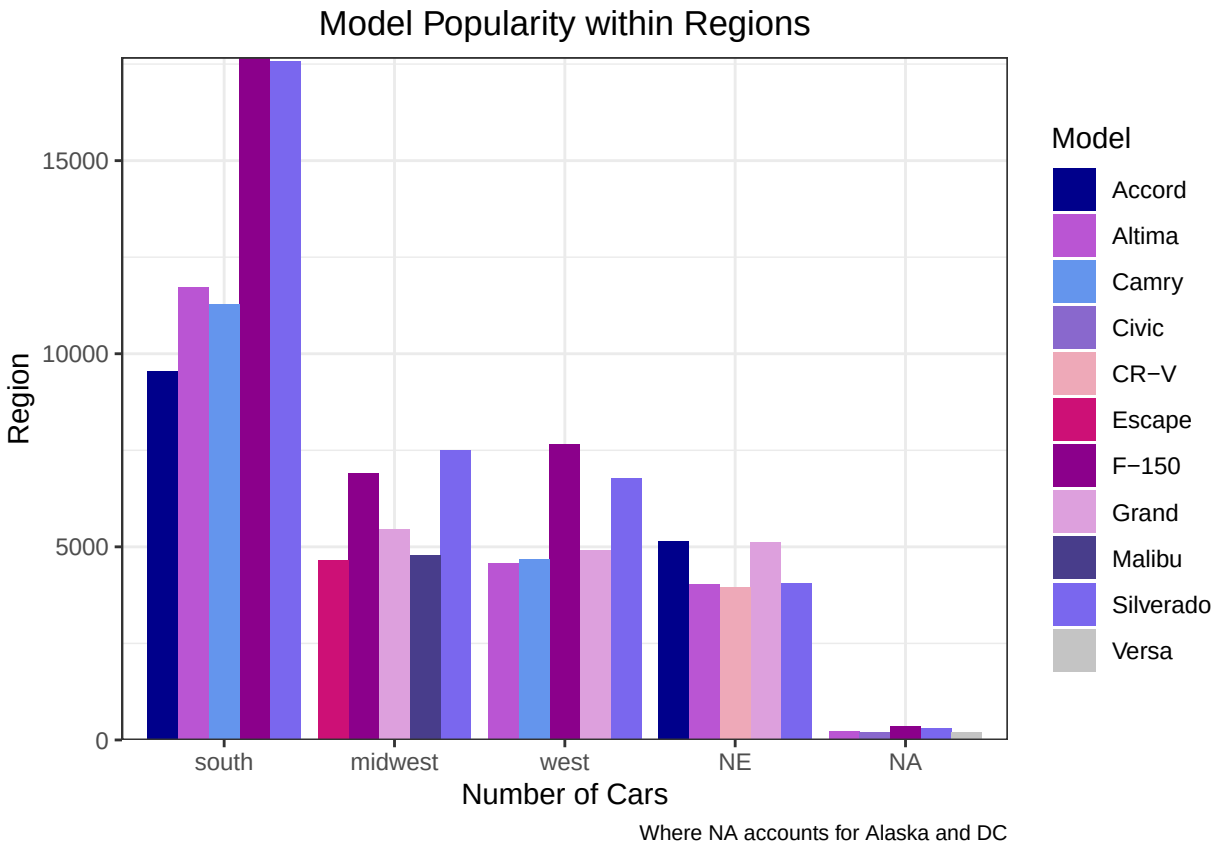



```
state_data <- pretty_cars %>%
  filter(!(State == "DC")) %>%
  group_by(State) %>%
  count(Model_Type, Make, sort = TRUE) %>%
  slice_max(n)

us_states <- states(year = 2019, resolution = "20m")

us_state_cont <- us_states %>%
  filter(!STATEFP %in% c("15", "78", "69", "66", "02", "60", "72")) %>%
  left_join(state_data, by = c("STUSPS" = "State")) %>%
  filter(!(is.na(Model_Type)))

ggplot()+
  geom_sf(
    data = us_state_cont,
    aes(fill = Model_Type),
    color = "grey48",
    size = .2)+
  theme_map()+
  scale_fill_manual(values = c( "midnightblue", "mediumpurple4", "darkorchid4","orchid3",
                                "royalblue3", "lightslateblue","blueviolet", "violet",
                                "plum2"),name = "Model")+
  labs(title = "Most Popular Models in the United States")+
  theme(plot.title = element_text(hjust = .5))
```

Most Popular Models in the United States




```
region_popularity_colors <- c("Accord" = "blue4", "Civic" = "mediumpurple3",
                              "CR-V" = "pink2", "Altima" = "mediumorchid",
                              "Versa" = "grey77","Camry" = "cornflowerblue",
                              "Escape" = "deeppink3", "F-150" = "magenta4",
                              "Grand" = "plum", "Malibu" = "darkslateblue",
                              "Silverado" = "slateblue2")

pretty_cars %>%
  group_by(region) %>%
  count(Model_Type, sort = TRUE) %>%
  slice(1:5) %>%
  ggplot(aes( x = reorder(region, -n), y = n, fill = Model_Type))+
  geom_bar(position = "dodge", stat = "identity")+
  scale_y_continuous(expand = c(0,0))+
  labs(title = "Model Popularity within Regions", x = "Number of Cars", y = "Region",
       fill = "Model", caption = "Where NA accounts for Alaska and DC")+
  scale_fill_manual(values = region_popularity_colors)+
  theme_bw()+
  theme(plot.title = element_text(hjust = .5), plot.caption = element_text(size = 8))
```



Evidently, the distribution of Fords, Chevys, and Toyotas show that there is an abundance of cars in Texas, California, and Florida. Each of the makes also seem to be equivalent in their distribution across the country. The regional graph reassures that there are more cars in the south than any other region. Since the regional bar graph accounts for the top 5 most popular models in each region, we can see that interestingly enough, F-150s and Silverados are the most popular types of vehicles in every region except the northeast. In the states from Pennsylvania up to Maine, Honda Accords and Jeep Grand Cherokees are the most popular vehicles. In fact, Ford F-150s aren't even in the top 5 models for the northeast region.

The map of the most popular models in each state show again that F-150s and Silverados are the most popular model in a majority of the states. In the northeast region, you can also verify that Accords are the most popular. California, Florida, Virginia, Delaware, Massachusetts, and Illinois seem to be outliers in the fact that their most popular vehicle is different than every other state. In these states, the most popular vehicle is a car while the non-outlier states have SUVs and trucks as their most popular vehicles.

Mileage, Age, and Price

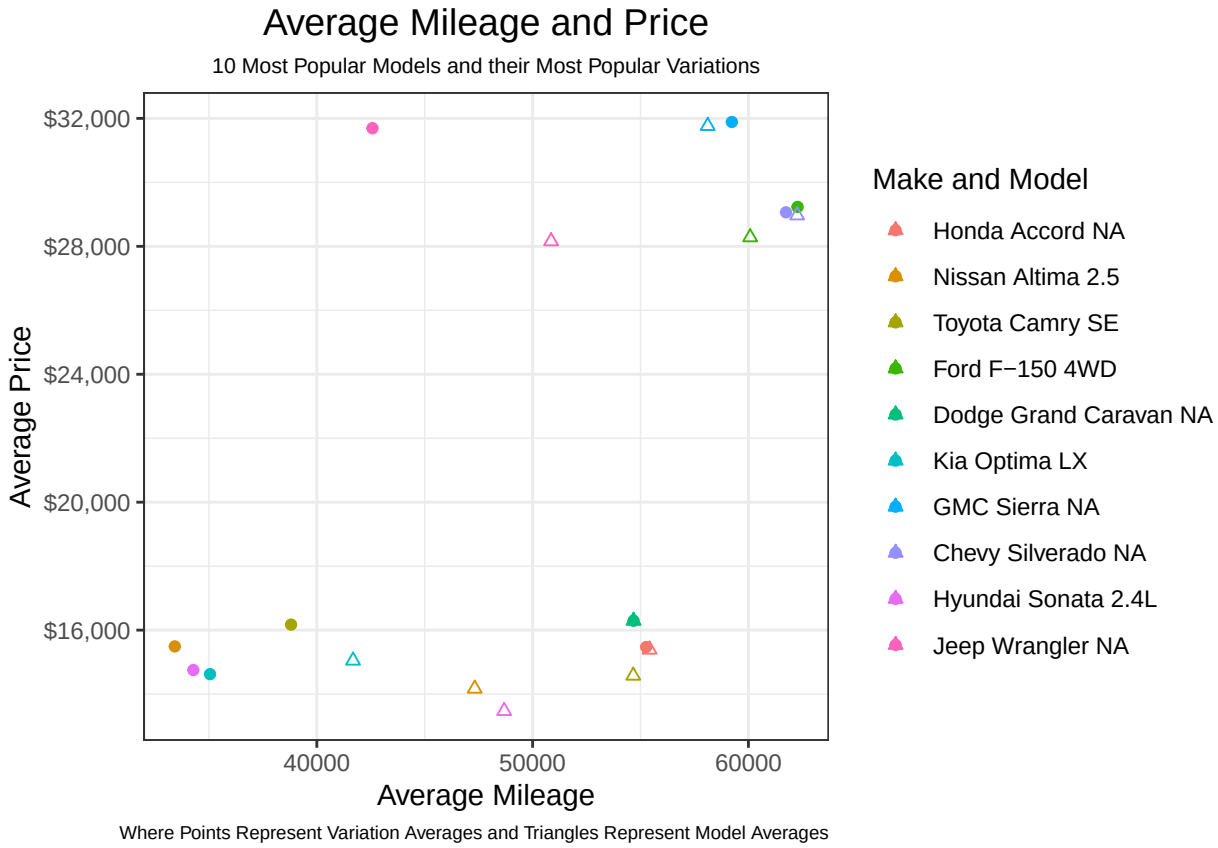
One of the main details that I wanted to explore within the dataset was the trends among age, mileage, and price. In addition to finding the averages of these factors for each model, I also wanted to compare this information with the averages of the most popular variation within each of these models.

```
most_popular_model_averages <- pretty_cars %>%
  filter(Model_Type == c("F-150", "Silverado", "Camry", "Altima", "Accord", "Sonata",
                         "Sierra", "Optima", "Wrangler")|
           Make == "Dodge" & Model_Type == "Grand") %>%
  group_by(Model_Type) %>%
  mutate(average_year = mean(Year)) %>%
  mutate(average_mileage = mean(Mileage)) %>%
```

```
  mutate(average_price = mean(Price)) %>%
  slice(1)
```

```
## Warning in Model_Type == c("F-150", "Silverado", "Camry", "Altima", "Accord", :
## longer object length is not a multiple of shorter object length
```

```
pretty_cars %>%
  filter(Model_Type == "F-150" & Model_Variation == "4WD" |
           Model_Type == "Silverado" & is.na(Model_Variation) |
           Model_Type == "Camry" & Model_Variation == "SE"|
           Model_Type == "Altima" & Model_Variation == "2.5"|
           Model_Type == "Accord" & is.na(Model_Variation) |
           Model_Type == "Wrangler" & is.na(Model_Variation) |
           Model_Type == "Sonata" & Model_Variation == "2.4L" |
           Make == "Dodge" & Model_Type == "Grand" & is.na(Model_Variation) |
           Model_Type == "Sierra" & is.na(Model_Variation) |
           Model_Type == "Optima" & Model_Variation == "LX") %>%
  group_by(Model_Type) %>%
  mutate(average_mileage = mean(Mileage)) %>%
  mutate(average_price = mean(Price)) %>%
  slice(1) %>%
  select(Make, Model_Type, Model_Variation, average_mileage, average_price) %>%
  arrange(desc(average_mileage)) %>%
  ggplot(aes(average_mileage, average_price,color = Model_Type))+
  geom_point()+
  scale_color_discrete(labels = c("Honda Accord NA", "Nissan Altima 2.5", "Toyota Camry SE",
                                  "Ford F-150 4WD", "Dodge Grand Caravan NA", "Kia Optima LX",
                                  "GMC Sierra NA", "Chevy Silverado NA", "Hyundai Sonata 2.4L",
                                  "Jeep Wrangler NA"))+
  theme_bw()+
  labs(title = "Average Mileage and Price",
       subtitle = "10 Most Popular Models and their Most Popular Variations",
       y = "Average Price", x = "Average Mileage",
       caption = "Where Points Represent Variation Averages and Triangles Represent Model Averages",
       color = "Make and Model")+
  scale_y_continuous(labels = scales::dollar_format())+
  theme(plot.title = element_text(hjust = .5, size = 14),
        plot.subtitle = element_text(hjust = .5, size = 8),
        plot.caption = element_text(size = 7))+
  geom_point(data = most_popular_model_averages, aes(average_mileage,average_price), shape = 2)
```



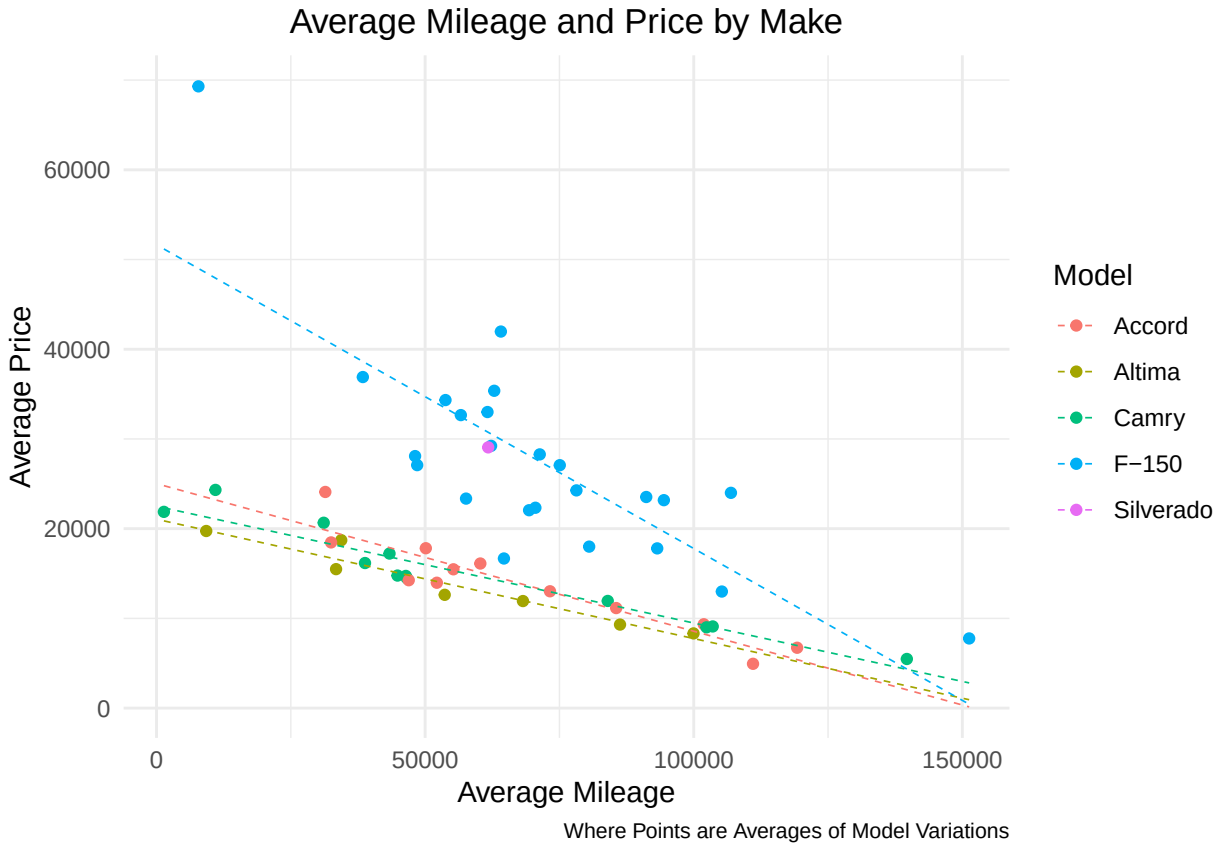
It is in this graph that we can see how the most popular variation of a model compares to the model as a whole. Specifically, Nissan Altima 2.5s tend to have less mileage on them than Nissan Altimas as a whole. The same is true for Kia Optimas, Hyundai Sonatas, Toyota Camrys, and Jeep Wranglers. You can also notice that the average mileage and price of Honda Accord NAs and Dodge Grand Caravan NAs are close to the averages of all Honda Accords and all Dodge Grand Caravans, respectively. Another trend this graph shows is how Ford F-150s, Chevy Silverados, GMC Sierras, and Jeep Wranglers have high prices for their mileage compared to the rest of the vehicles in the graph.

To continue in the exploration of model types and their average mileage, price, and year, I had the following graph display the averages of the model variations (for the most popular models of the top 5 makes) along with an average line for the makes as a whole.

```
pretty_cars %>%
  filter(Model_Type == "Silverado" |
           Model_Type == "Camry" |
           Model_Type == "F-150" |
           Model_Type == "Altima" |
           Model_Type == "Accord") %>%
  group_by(Make, Model_Type, Model_Variation) %>%
  summarise(average_variation_price = mean(Price), average_year = mean(Year),
            average_mileage = mean(Mileage)) %>%
  arrange(desc(average_variation_price)) %>%
  ggplot(aes(average_mileage, average_variation_price, color = Model_Type))+
  geom_point()+
  geom_smooth(method = lm, linetype = "dashed", fullrange = TRUE, se = FALSE, size = .3)+
  theme_minimal()+
  theme(plot.caption = element_text(size = 8),
        plot.title = element_text(hjust = .5))+
```

```
  labs(x = "Average Mileage", y = "Average Price", title = "Average Mileage and Price by Make",
       caption = "Where Points are Averages of Model Variations", color = "Model")
```

```
## `summarise()` has grouped output by 'Make', 'Model_Type'. You can override using the `.groups` argume
## `geom_smooth()` using formula 'y ~ x'
```



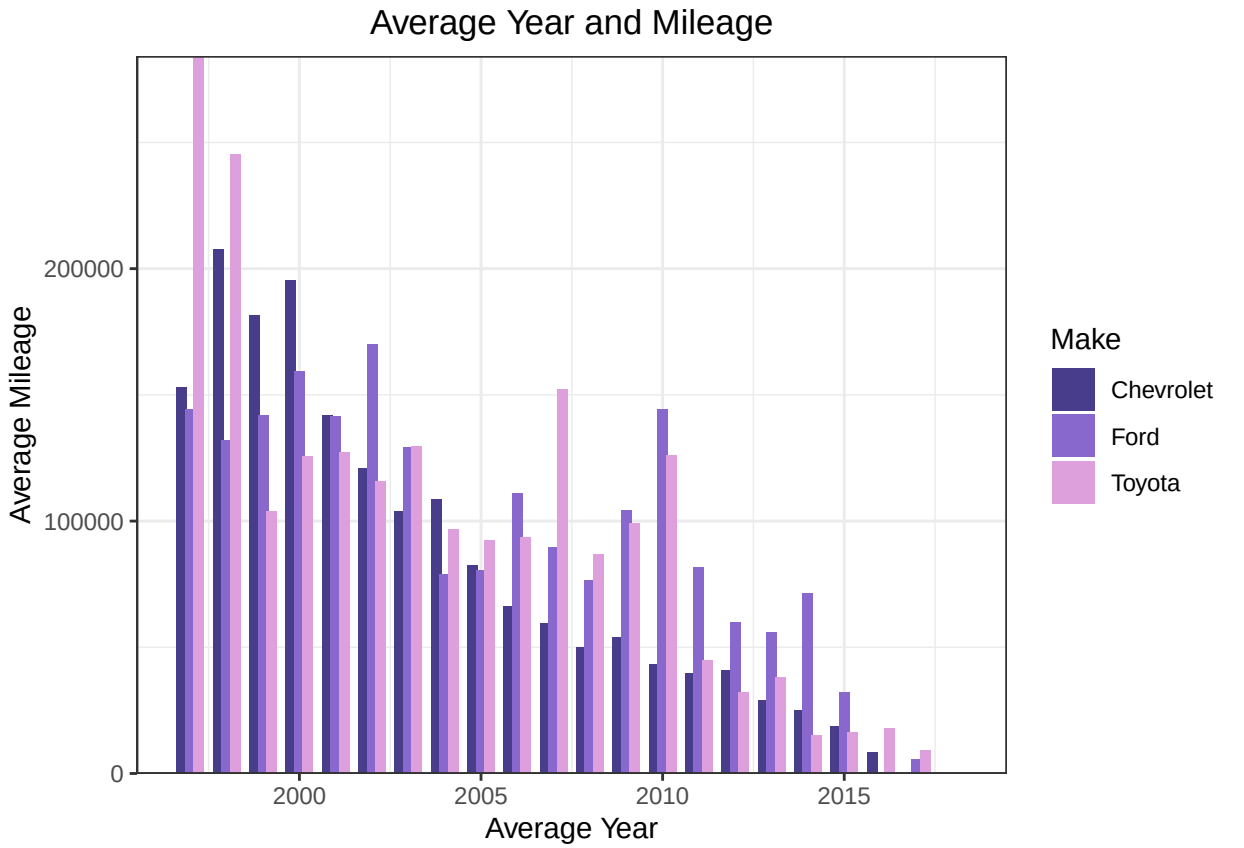
One important thing to notice in this graph is that Chevy Silverados don't have an average line because there is only one variation (and therefore not enough information to make a trendline). Here we can also see that the average price of Ford F-150s per mile is much greater than the other models. There does seem to be an intersection point of all but one of the models when mileage is 150,000 miles and price is almost $0. Out of the 3 car models (meaning Accord, Altima, and Camry), Toyota Camrys tend to maintain the greatest average price as mileage increases. The cars also seem to have less outliers than the trucks do.

Anyone who is devoted to finding a good deal on a car knows to pay attention to its mileage. In addition to its variation or age, mileage can indicated how much longer the car will run.To gauge how the age of Fords, Chevys, and Toyotas impacts their mileage, I created a graph that would show me the trend of mileage over time.

```
pretty_cars %>%
  filter(Make == "Ford" | Make == "Chevrolet"|Make == "Toyota") %>%
  group_by(Make, Model_Type, Year, Mileage) %>%
  summarise(average_price = mean(Price), average_mileage = mean(Mileage),
            average_year = mean(Year)) %>%
  slice(1) %>%
  ggplot(aes(average_year, average_mileage, fill = Make))+
  geom_col(position = position_dodge(width = .7))+
  scale_fill_manual(values = c("darkslateblue", "mediumpurple3", "plum"))+
```

```
  scale_y_continuous(expand = c(0,0))+
  theme_bw()+
  labs(title = "Average Year and Mileage", x = "Average Year", y = "Average Mileage")+
  theme(plot.title = element_text(hjust = .5))
```

```
## `summarise()` has grouped output by 'Make', 'Model_Type', 'Year'. You can override using the `.groups
```



Here we can see that Toyotas tend to have the highest mileage over time, followed by Fords then Chevys. While the mileage of each make tends to decline as time goes on, the trend of Toyota mileage seems to be nearly bimodal with a second peak (although not as large as the first) occuring near the year 2007. This being said, it seems that Toyotas can run longer than Chevys or Fords, on average.

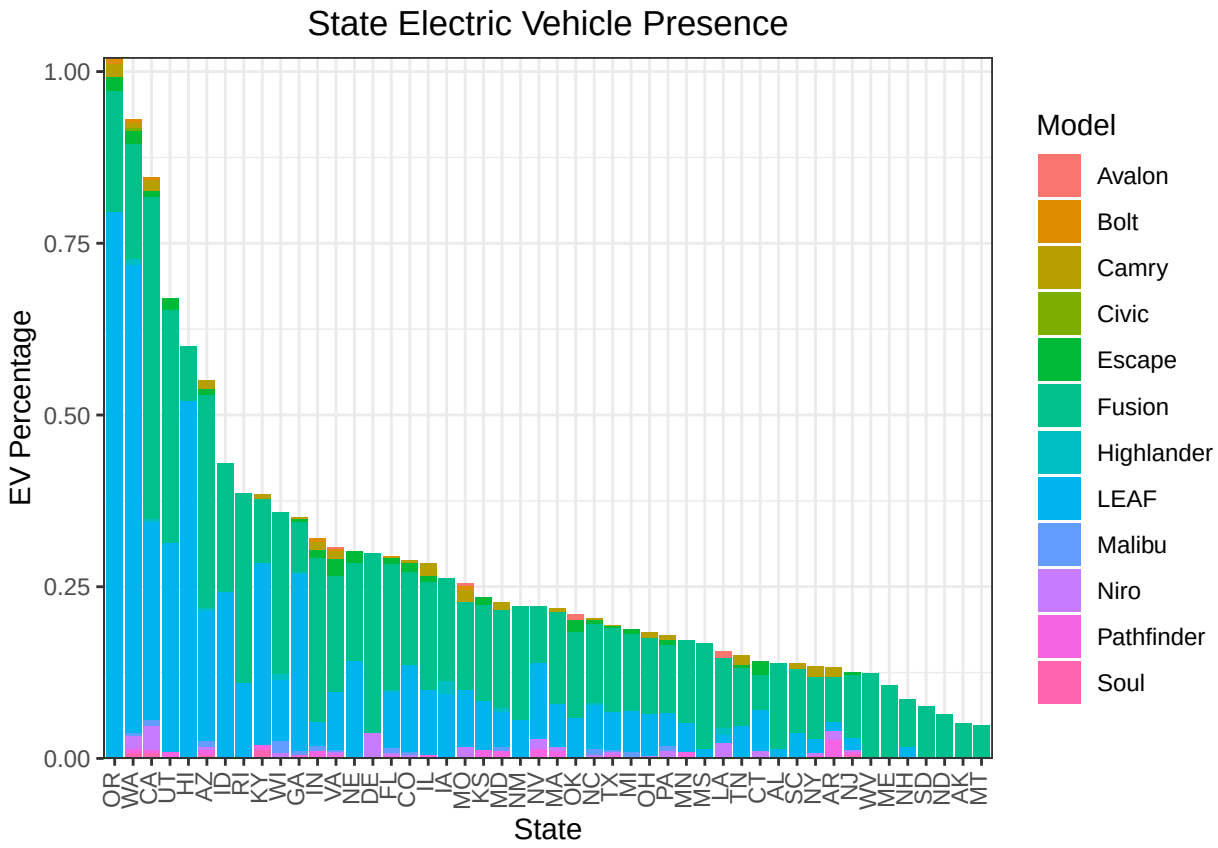
Electric Vehicles

The presence of electric vehicles within the country and state was also something I wanted to investigate. I had expected to see a greater amount of EVs in cities, but I was unsure about their popularity in suburban or rural settings. Because of this, I created a graph that would compare which state had the most EVs and another that would show which city in PA had the most EVs.

The easiest way for me to create the percentages was to add a column to the pretty_cars dataset called "cars_per_state". This "cars_per_state" column counts how many cars were listed for sale in each state. With this column, I could calculate the percentage of EVs rather than just the count. I appied the same method to create the second graph but instead of finding the number of cars per state, I found the number of "cars_per_city".

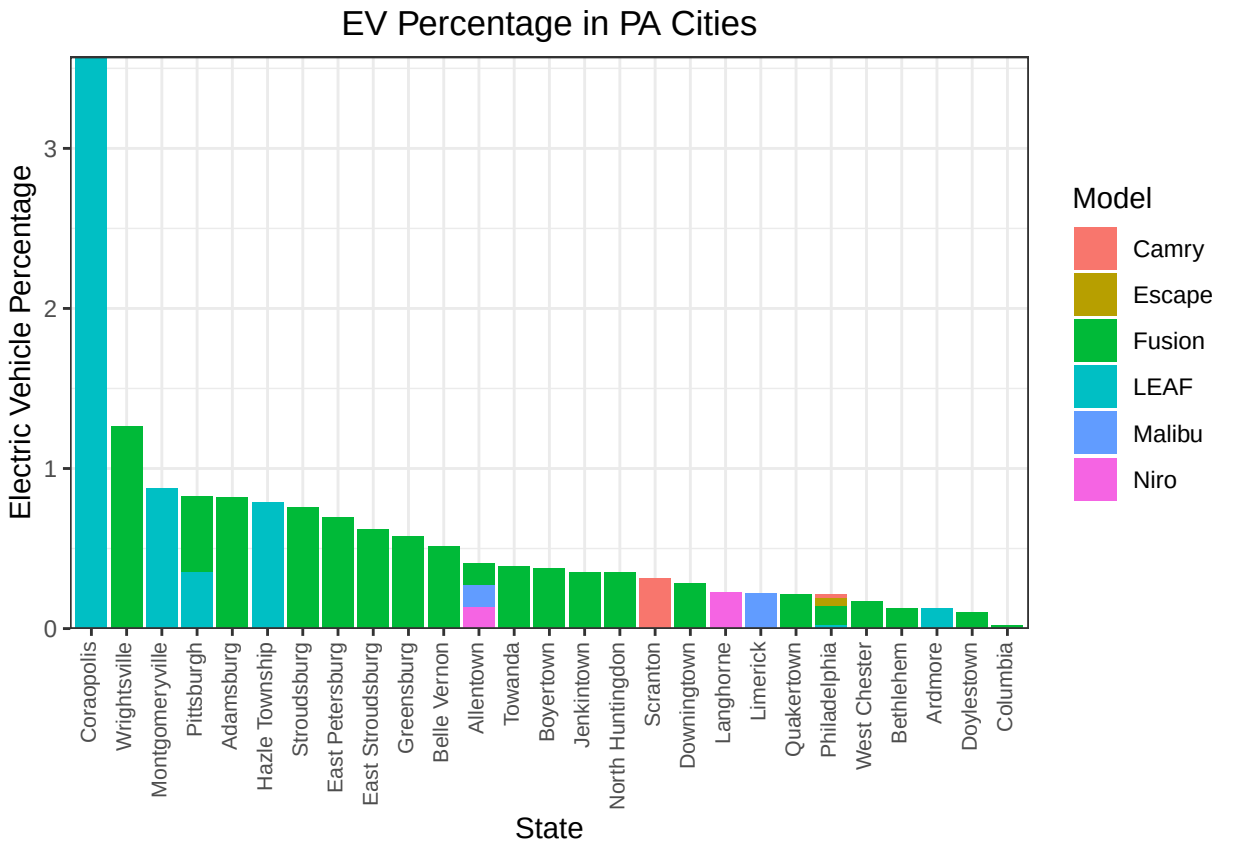
```
cars_per_state <- pretty_cars %>%
  count(State) %>%
```

```
  mutate(cars_per_state = n) %>%
  select(State, cars_per_state)

pretty_cars <- pretty_cars %>%
  full_join(cars_per_state, by = "State")

pretty_cars %>%
  filter(Model_Variation == "Hybrid" |
           Model_Type == "Bolt" |
           Model_Type == "Niro" |
           Model_Type == "LEAF" |
           Model_Variation == "EV") %>%
  group_by(State) %>%
  count(Model_Type, Make, City, cars_per_state) %>%
  ungroup() %>%
  mutate(ev_percentage = 100*(n/cars_per_state)) %>%
  arrange(desc(ev_percentage)) %>%
  ggplot(aes(x = reorder(State, -ev_percentage, sum),
             y = ev_percentage,
             fill = Model_Type))+
  geom_col()+
  theme_bw()+
  scale_y_continuous(expand = c(0,0))+
  labs(x = "State", y = "EV Percentage", fill = "Model",
       title = "State Electric Vehicle Presence")+
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = .3),
        plot.title = element_text(hjust = .5))
```




```
cars_per_city <- pretty_cars %>%
  count(City) %>%
  mutate(cars_per_city = n) %>%
  select(City, cars_per_city)

pretty_cars <- pretty_cars %>%
  full_join(cars_per_city, by = "City")

pretty_cars %>%
  filter(Model_Variation == "Hybrid" |
           Model_Type == "Bolt" |
           Model_Type == "Niro" |
           Model_Type == "LEAF" |
           Model_Variation == "EV",
         State == "PA") %>%
  group_by(City) %>%
  count(Model_Type, Make, City, cars_per_city) %>%
  mutate(ev_percentage = 100*(n/cars_per_city)) %>%
  arrange(desc(ev_percentage)) %>%
  ggplot(aes(x = fct_reorder(City, -ev_percentage, sum), y = ev_percentage,
             fill = Model_Type))+
  geom_col()+
  theme_bw()+
  scale_y_continuous(expand = c(0,0))+
  labs(x = "State", y = "Electric Vehicle Percentage", fill = "Model",
       title = "EV Percentage in PA Cities")+
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = .3, size = 8),
```

```
        plot.title = element_text(hjust = .5))
```



The first thing that stands out from these graphs is that Nissan LEAFs seem to be the most popular EV in the United States while Ford Fusion Hybrids are the most popular EV in PA. Another notable part of these graphs is that the EV percentages in Oregon, Washington, and California are much more substantial than many of the other states. While EVs still only account for about 1% of the cars for sale in each of these three states, the impact of 1% in California compared to less than .1% in Alaska is impressive in relativity. I thought it was also interesting to see that PA is near the tail end of the graph when we have two large cities (Philadelphia and Pittsburg).

Just as it was interesting to see the relationship between the states, it's also interesting to see the relationship between the cities in PA. While I expected Philadelphia and Pittsburgh to have the greatest percentage of EVs, it turns out that a town I had never heard before, Coraopolis, did instead. Coraopolis seemed to be made entirely of Nissan LEAFs when most of the other cities in PA were composed of Ford Fusions or a combination of the EV models. Similar to the graph comparing the states, the PA graph shows a drastic difference between the highest and lowest percentage with a not so smooth gradient in between. I thought my hometown, Lancaster, would make this graph since it is a decent sized city but apparently its EV count is not significant enough to make an appearance.

Final Notes

Linear Regression

One of the more interesting parts of this project to me was the ability to predict price based on mileage and age. This task seemed simpler to approach when focusing on one kind of model and since I currently drive a Honda Accord, I decided to complete a linear regression using data on Accords.

```r
linear_regression_data <- pretty_cars %>%
  filter(Model_Type == "Accord")

model <- lm(Price ~ Mileage+Year, data = linear_regression_data)
summary(model)
```

```
##
## Call:
## lm(formula = Price ~ Mileage + Year, data = linear_regression_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8882  -1691   -336   1331 106353
##
## Coefficients:
##                      Estimate       Std. Error t value            Pr(>|t|)
## (Intercept) -1793769.6278398    17679.2517252 -101.46 <0.0000000000000002 ***
## Mileage           -0.0394497        0.0006765  -58.31 <0.0000000000000002 ***
## Year             899.9160329        8.7677351  102.64 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2545 on 22170 degrees of freedom
## Multiple R-squared:  0.7687, Adjusted R-squared:  0.7686
## F-statistic: 3.683e+04 on 2 and 22170 DF,  p-value: < 0.00000000000000022
```

These results tell me that the equation $Price = -1,793,769.628 + 899.916 * Year - .039 * Mileage$ is fairly accurate in predicting price and that mileage and age are important variables in this equation. If someone would like, this process could be repeated for other models to determine what kind of deal they would be getting when purchasing or selling a car.

Limitations

Working through this project revealed some limitations both within the dataset and my work. For example, splitting the model types and model variations overlooked types of vehicles with names that were more than one word. Models like Ford Five Hundred became Ford Five. Jeep Grand Cherokees showed up as Jeep Grand or Jeep Cherokee. Dodge Grand was supposed to be Dodge Caravan, and Toyota Land and Toyota FJ were meant to have Cruiser as their second word. Additionaly, Hyundai Santas are really Hyundai Sante Fes.

There is more explanation due in the model variation category. When manually entering the number of letters in the model name, I incorrecly entered Hyundai Genesis to be Hyundai Genesi. This mistake is not too difficult to correct, but it is a multi-step process. Some variations were also confusing in their naming. The most popular variation of Nissan Altimas was 2.5. This 2.5 represents the number of liters the engine has and after some reasearch, I found that more liters results in a more powerful engine. Another decision I made when making the model variation category was to substitute missing variations with *NA*. The missing variation could be due to the vehicle being a baseline model or simply because the information was not available on TrueCar.com. As previously mentioned, there was a handful of models that seemed to have no variations (like the Silverado). While this may be true in the downloaded data, a Google search revealed that in fact, there are multiple variations of Chevy Silverados.

The missing variations could result in some of the information in this report being skewed. Another reason could be due to the fact that the cars listed for sale do not necessarily represent cars that are on the road. For example, the percentages of cars depends on the number of cars listed in the dataset. While

the information on TrueCar.com could be a random sample of cars being driven, I believe there is some difference between cars we see while driving and cars we see sitting on the side of the road with "For Sale" signs. The last contributor I considered when analyzing this data was where the data was taken when it was downloaded from TrueCar.com. Whenever you shop for cars online, websites tend to show you cars for sale that are within a radius of x many miles from your location. For this reason, I am slightly concerned that this may have happened unknowingly to the person that compiled this data.

Overall, the exploration of this dataset clearly revealed several different patterns that car owners can consider when shopping for a new car or selling an old one. There is a lot that this report covered and still more that it didn't (like seeing the distribution of cars in cities, repair costs, mpg, etc.), but I believe that the information contained in this write up can be useful to many people. From those that have little to no knowledge of cars to those that know much more than me, the exploration of this dataset can be useful in a multitude of ways.