EDA1

Emily Goldfarb

2023-12-06

Using fread() instead of read_csv since data is so large

```
setwd("~/Documents/Rstudio/CPSC 537/Project")
library(data.table)
library(ggplot2)
library(stringr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::between()     masks data.table::between()
## x dplyr::filter()      masks stats::filter()
## x dplyr::first()       masks data.table::first()
## x lubridate::hour()    masks data.table::hour()
## x lubridate::isoweek() masks data.table::isoweek()
## x dplyr::lag()         masks stats::lag()
## x dplyr::last()        masks data.table::last()
## x lubridate::mday()    masks data.table::mday()
## x lubridate::minute()  masks data.table::minute()
## x lubridate::month()   masks data.table::month()
## x lubridate::quarter() masks data.table::quarter()
## x lubridate::second()  masks data.table::second()
## x purrr::transpose()   masks data.table::transpose()
## x lubridate::wday()    masks data.table::wday()
## x lubridate::week()    masks data.table::week()
## x lubridate::yday()    masks data.table::yday()
## x lubridate::year()    masks data.table::year()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

EDA for Dataset1 used_cars_data.csv cols to import chosen based off of EDA from python notebook Data1 preprocessing

```
d1_cols <- c('vin',  'body_type',
  'city', 'city_fuel_economy',
  'daysonmarket', 'dealer_zip',
  'engine_displacement', 'engine_type',
  'franchise_dealer',
  'fuel_tank_volume', 'fuel_type', 'has_accidents', 'height',
  'highway_fuel_economy', 'horsepower',
'is_oemcpo', 'latitude', 'length',
```

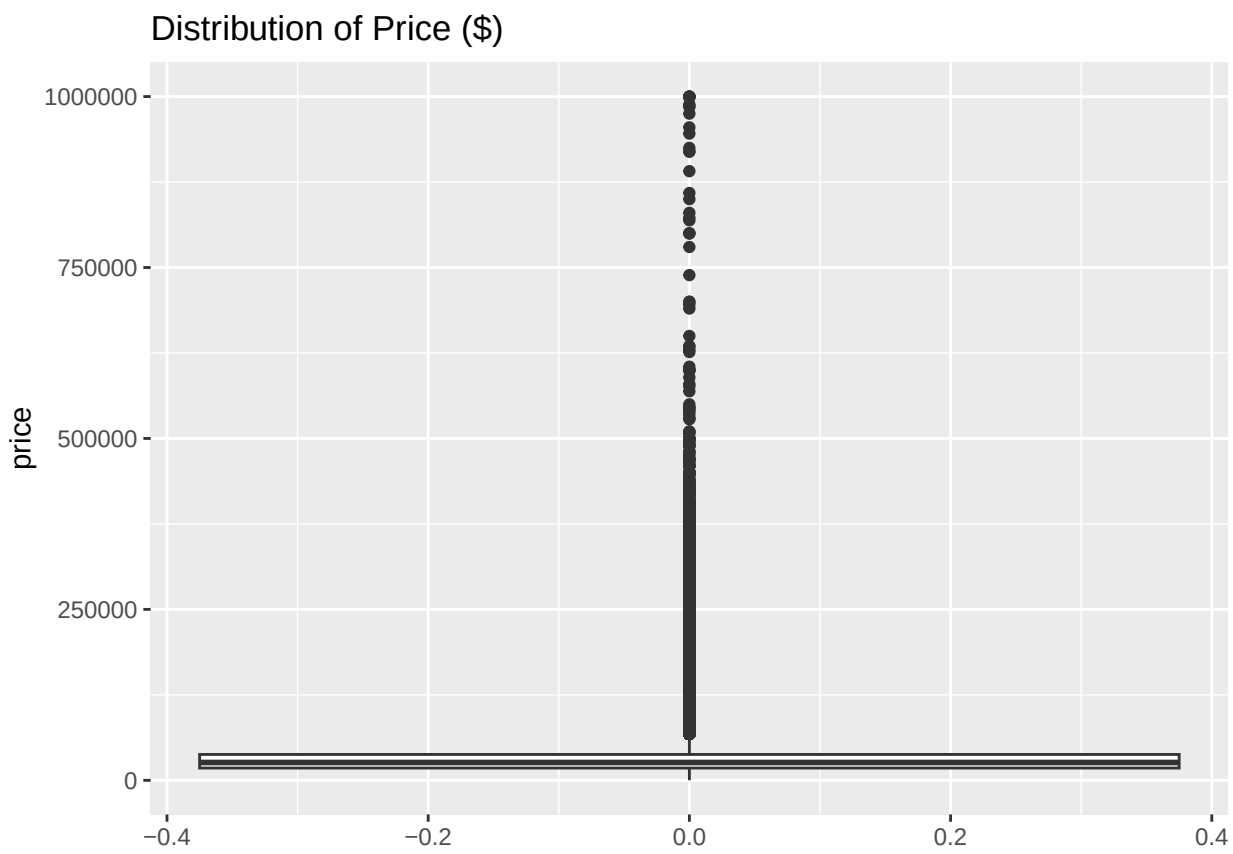
```
  'listed_date', 'listing_color', 'listing_id', 'longitude',
   'make_name', 'maximum_seating',
  'mileage', 'model_name',  'power', 'price',  'seller_rating', 'sp_id',
'sp_name',
  'torque', 'transmission', 'trimId', 'trim_name',
  'wheel_system',
  'wheelbase', 'width', 'year')

data1 <- fread("used_cars_data.csv", select = d1_cols)
```

Droppin na's from price and mileage, keeping cars with price and mileage under one million, and cleaning engine_type column, all based on work done in Data 1 preprocessing python notebook

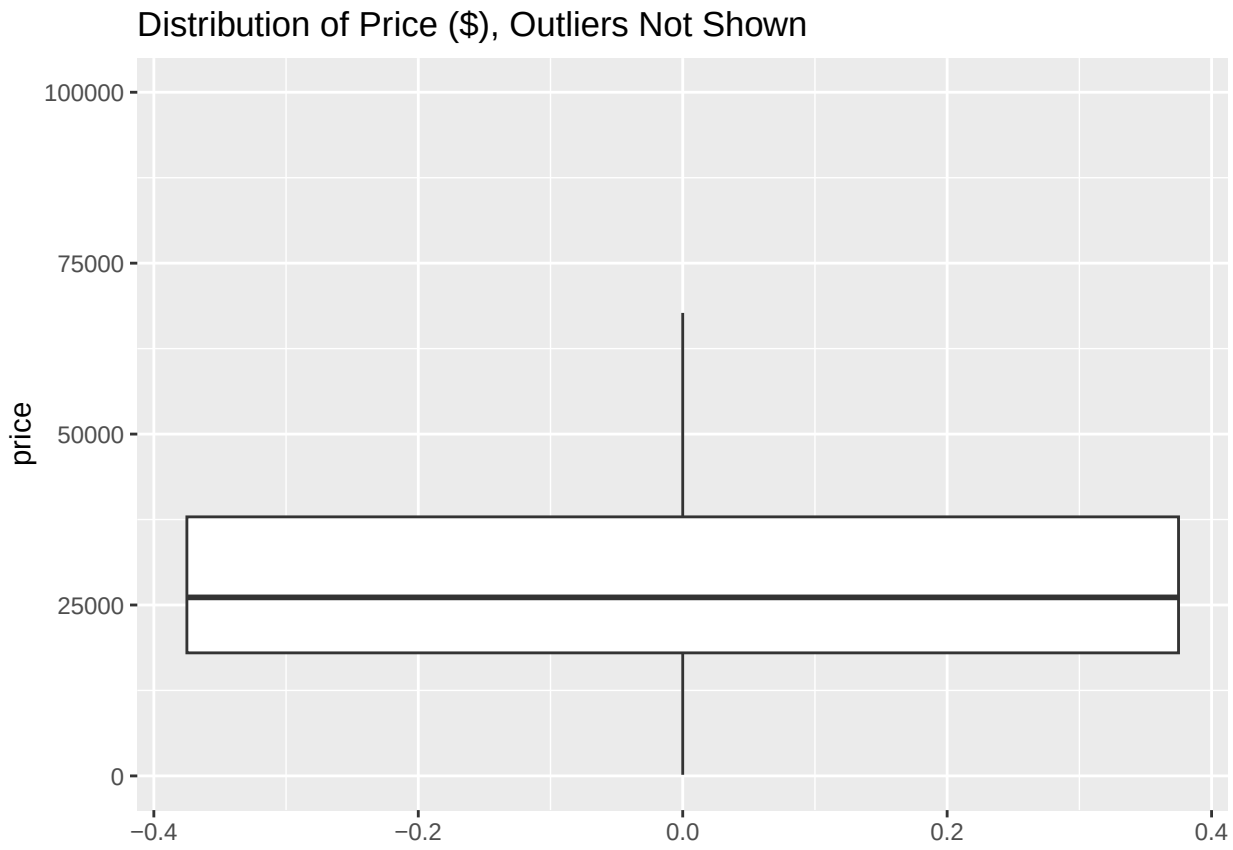
```
data2 <- data1 %>% drop_na(price, mileage)

data3 <- data2 %>% filter(price < 1000000, mileage < 1000000)
data3$engine_type[data3$engine_type == ""] <-  "Gasoline"
data3$engine_type <- gsub( " .*$", "", data3$engine_type)
```

Distribution of price, boxplots with and without outliers

```
ggplot(data = data3, aes(y = price)) + geom_boxplot() +
  ggtitle("Distribution of Price ($)")
```

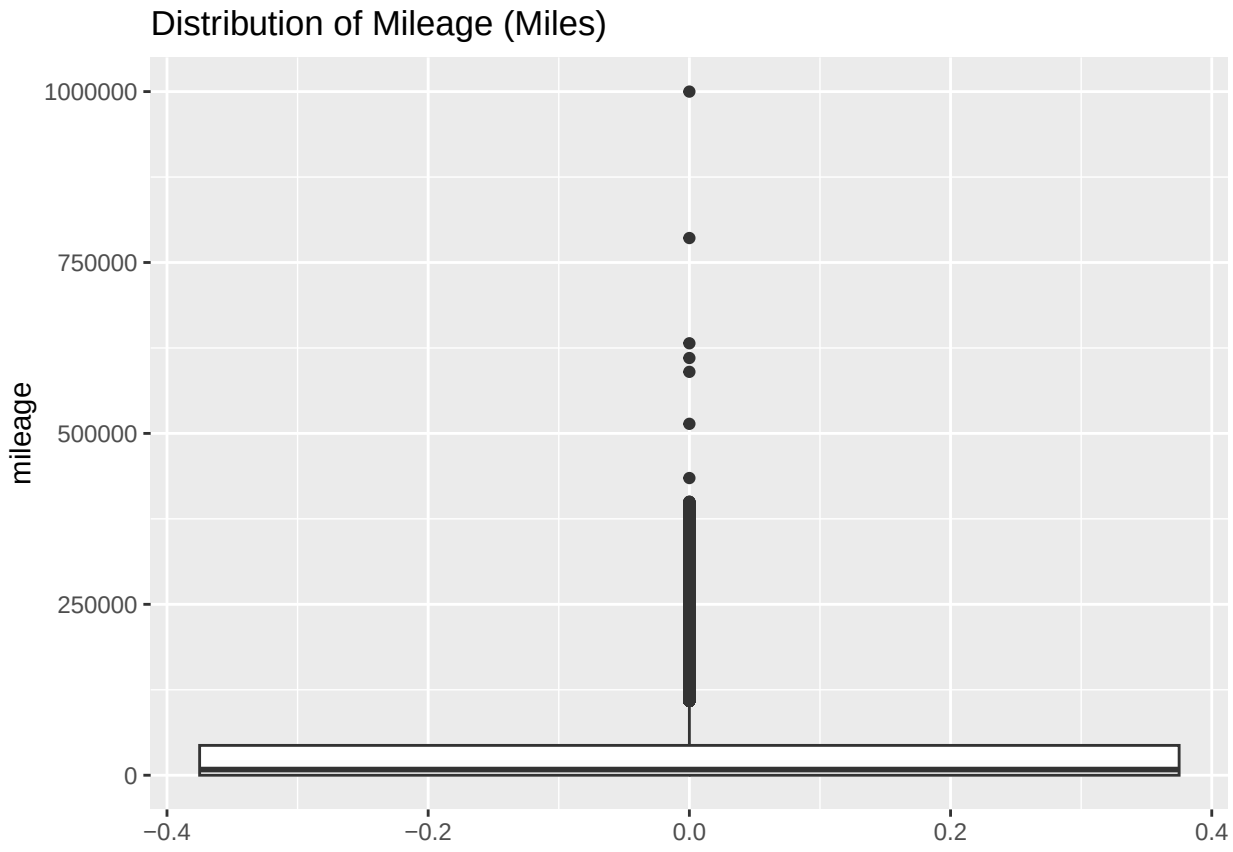



```
ggplot(data = data3, aes(y = price)) + geom_boxplot(outlier.shape = NA) +
  coord_cartesian(ylim = c(0, 100000))+
  ggtitle("Distribution of Price ($), Outliers Not Shown")
```

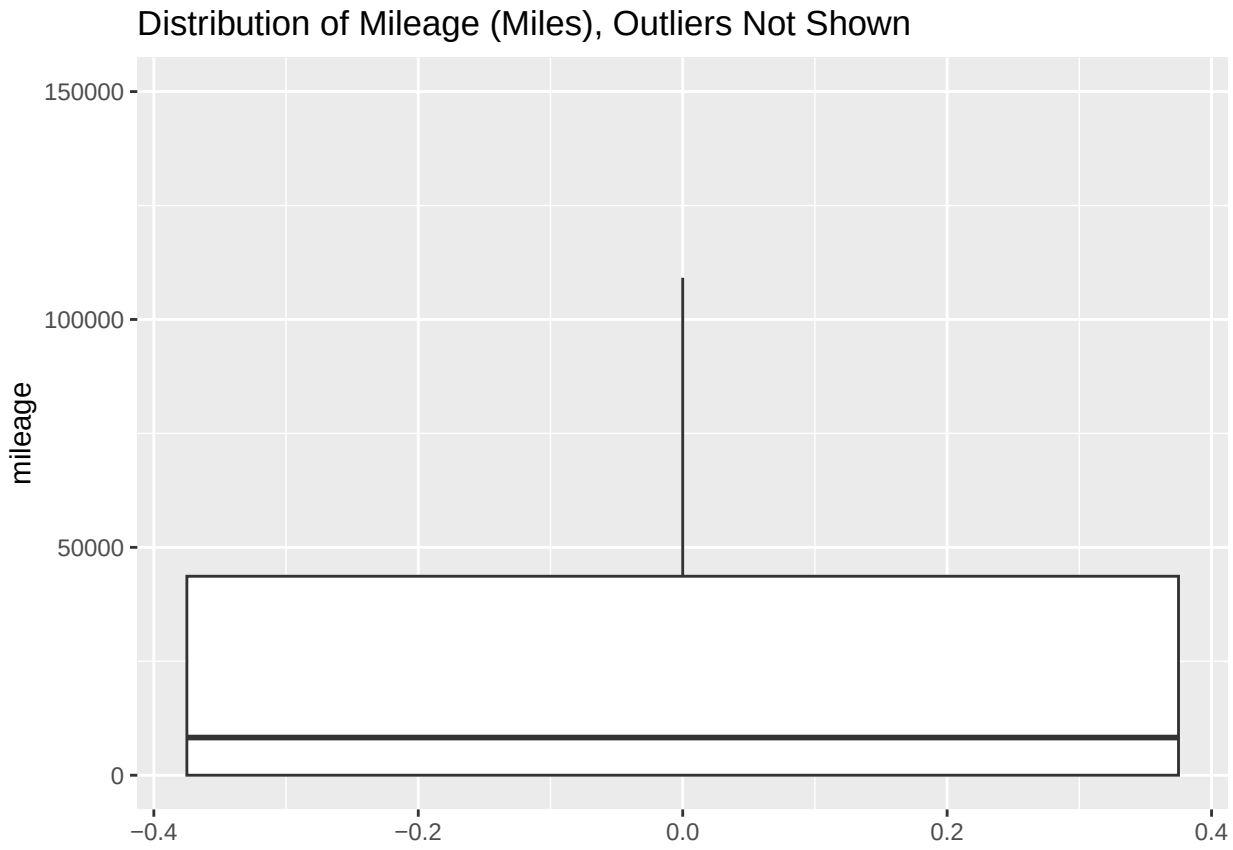


Distribution of mileage, boxplots with and without outliers

```
ggplot(data = data3, aes(y = mileage)) + geom_boxplot() +
  ggtitle("Distribution of Mileage (Miles)")
```

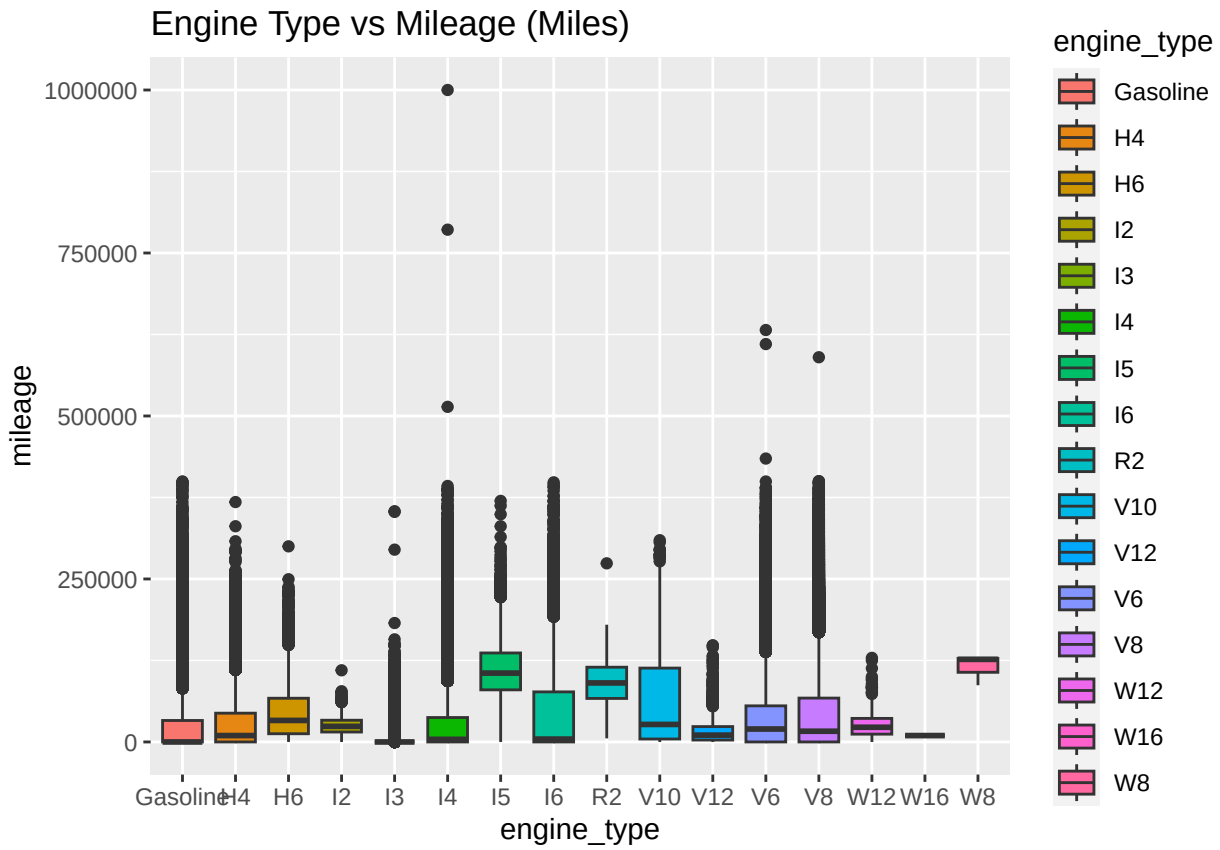



```
ggplot(data = data3, aes(y = mileage)) + geom_boxplot(outlier.shape = NA) +
  coord_cartesian(ylim = c(0, 150000))+
  ggtitle("Distribution of Mileage (Miles), Outliers Not Shown")
```



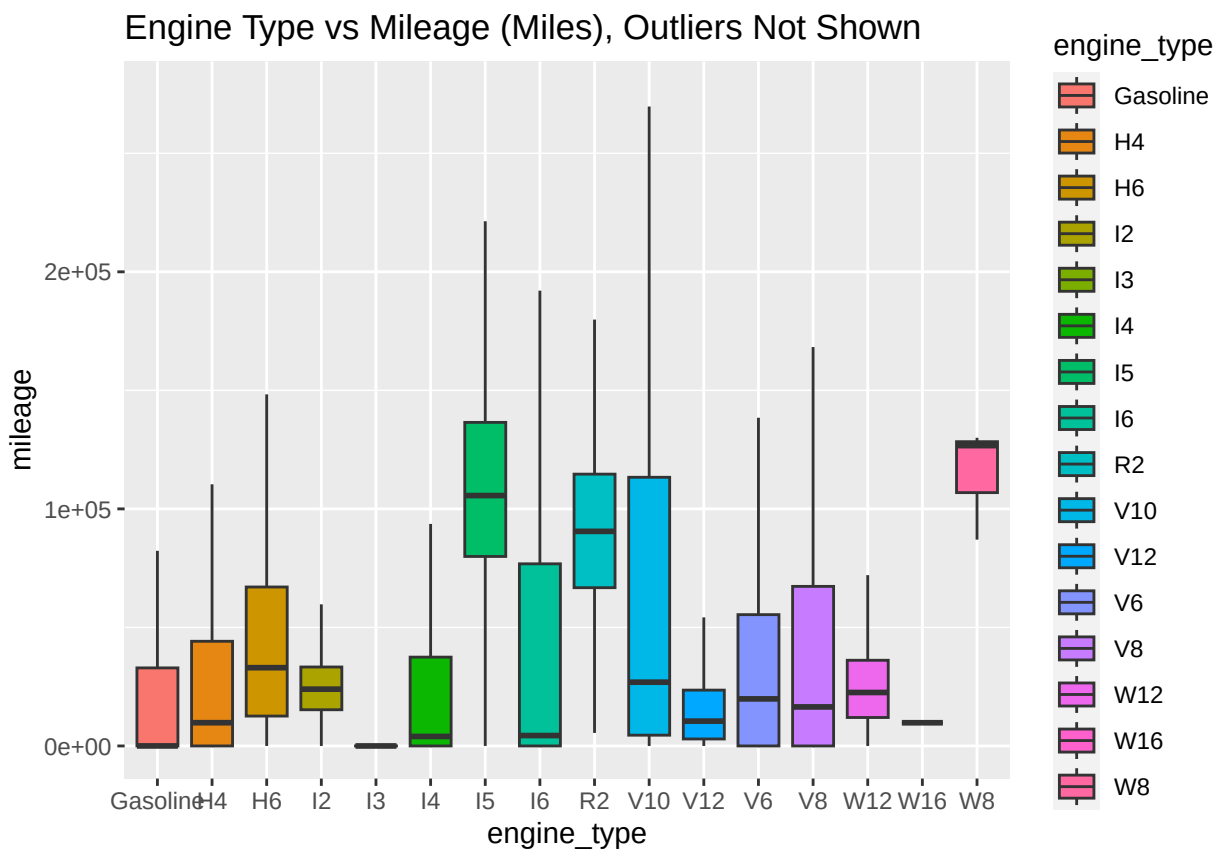
Engine type vs mileage with outliers boxplot, with outliers

```
ggplot(data3, aes(x = engine_type, y = mileage, fill = engine_type))+
  geom_boxplot(aes(fill = engine_type)) +
  ggtitle("Engine Type vs Mileage (Miles)")
```



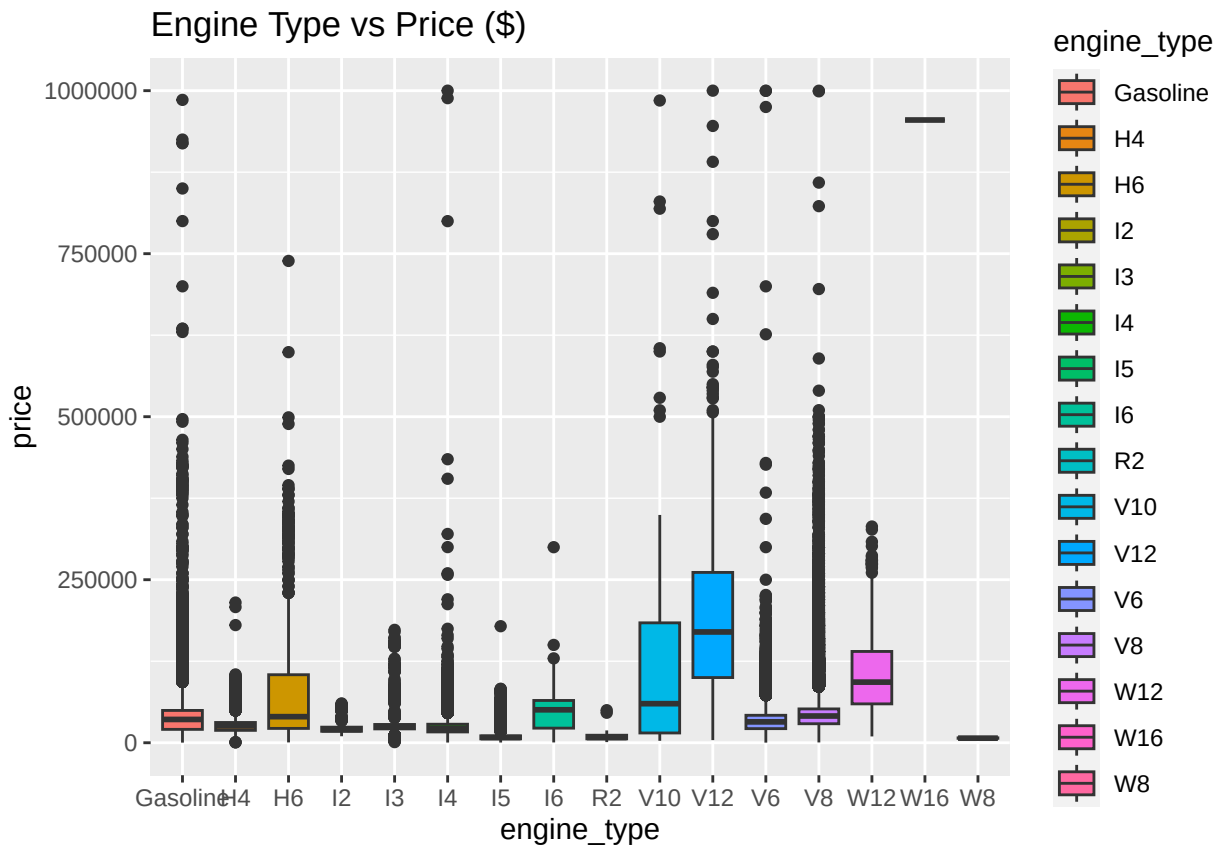
Engine type vs mileage no outliers boxplot

```
ggplot(data3, aes(x = engine_type, y = mileage, fill = engine_type)) +
  geom_boxplot(outlier.shape = NA) + coord_cartesian(ylim = c(0, 275000))+
  ggtitle("Engine Type vs Mileage (Miles), Outliers Not Shown")
```



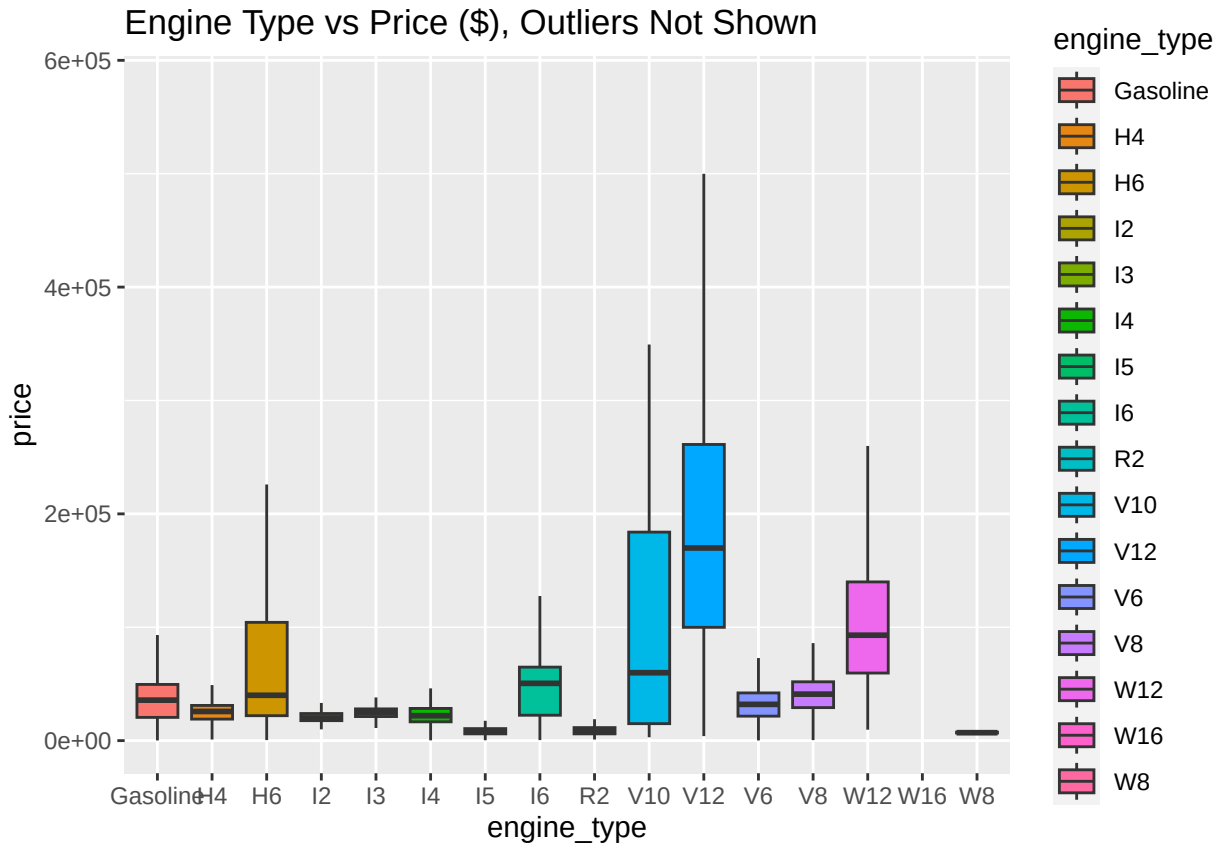
Engine type vs price with outliers boxplot

```
ggplot(data3, aes(x = engine_type, y = price, fill = engine_type)) +
  geom_boxplot() + ggtitle("Engine Type vs Price ($)")
```



Engine type vs price no outliers boxplot

```
ggplot(data3, aes(x = engine_type, y = price, fill = engine_type)) +
  geom_boxplot(outlier.shape = NA) + coord_cartesian(ylim = c(0, 575000))+
  ggtitle("Engine Type vs Price ($), Outliers Not Shown")
```



Price vs Mileage scatterplot Did encounter a Mac error where plot won't display sometimes. I saved a png version of an instance where the plot did run Restarting R session allowed scatter to run Plot is exclusively in the saved images folder to mitigate strain on R studio

```
scatter <- ggplot(data3, aes(x = price, y = mileage)) +
  geom_point() + ggtitle("Price ($) vs Mileage (Miles)")

#scatter
```