Quick EDA on mpd dataset

Osayame Eghosa

2024-04-18

```
#installing needed library and restricting filter to work based on dplyr
library(tidyverse)
library(conflicted)
conflicts_prefer(dplyr::filter)
```

In the code above, we are installing the conflicted package if it is not already installed. Then, we load the tidyverse library, which includes several packages for data manipulation and visualization, and the conflicted library. The conflicts_prefer() function from conflicted ensures that when there is a conflict between function names, we prefer the one from the dplyr package.

```
#Running quick EDA on the dataset
data()
summary(mpg)
```

```
##  manufacturer          model              displ            year
##  Length:234         Length:234         Min.   :1.600   Min.   :1999
##  Class :character   Class :character   1st Qu.:2.400   1st Qu.:1999
##  Mode  :character   Mode  :character   Median :3.300   Median :2004
##                                        Mean   :3.472   Mean   :2004
##                                        3rd Qu.:4.600   3rd Qu.:2008
##                                        Max.   :7.000   Max.   :2008
##       cyl           trans               drv                 cty
##  Min.   :4.000   Length:234         Length:234         Min.   : 9.00
##  1st Qu.:4.000   Class :character   Class :character   1st Qu.:14.00
##  Median :6.000   Mode  :character   Mode  :character   Median :17.00
##  Mean   :5.889                                         Mean   :16.86
##  3rd Qu.:8.000                                         3rd Qu.:19.00
##  Max.   :8.000                                         Max.   :35.00
##       hwy             fl                class
##  Min.   :12.00   Length:234         Length:234
##  1st Qu.:18.00   Class :character   Class :character
##  Median :24.00   Mode  :character   Mode  :character
##  Mean   :23.44
##  3rd Qu.:27.00
##  Max.   :44.00
```

```
glimpse(mpg)
```

```
## Rows: 234
## Columns: 11
## $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "audi", "…
## $ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 quattro", "…
## $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0, 2.0, 2.…
## $ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1999, 200…
## $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 8, 8, …
## $ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)", "auto…
## $ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4", "4", "4…
## $ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 17, 17, 1…
## $ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 25, 25, 2…
## $ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p…
## $ class        <chr> "compact", "compact", "compact", "compact", "compact", "c…
```

Here we perform some quick exploratory data analysis (EDA) on the mpg dataset. We use functions like data() to load the dataset, summary() to get summary statistics, and glimpse() to get a concise summary of the dataset's structure.

```
# Filter data
filter(mpg, cty >=20)
```

```
## # A tibble: 56 × 11
##    manufacturer model      displ  year   cyl trans drv     cty   hwy fl    class
##    <chr>        <chr>      <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
##  1 audi         a4           1.8  1999     4 manu… f        21    29 p     comp…
##  2 audi         a4           2    2008     4 manu… f        20    31 p     comp…
##  3 audi         a4           2    2008     4 auto… f        21    30 p     comp…
##  4 audi         a4 quattro   2    2008     4 manu… 4        20    28 p     comp…
##  5 chevrolet    malibu       2.4  2008     4 auto… f        22    30 r     mids…
##  6 honda        civic        1.6  1999     4 manu… f        28    33 r     subc…
##  7 honda        civic        1.6  1999     4 auto… f        24    32 r     subc…
##  8 honda        civic        1.6  1999     4 manu… f        25    32 r     subc…
##  9 honda        civic        1.6  1999     4 manu… f        23    29 p     subc…
## 10 honda        civic        1.6  1999     4 auto… f        24    32 r     subc…
## # ℹ 46 more rows
```

```
mpg_efficient <- filter(mpg, cty >=20)
mpg_ford <- filter(mpg, manufacturer == "ford")
```

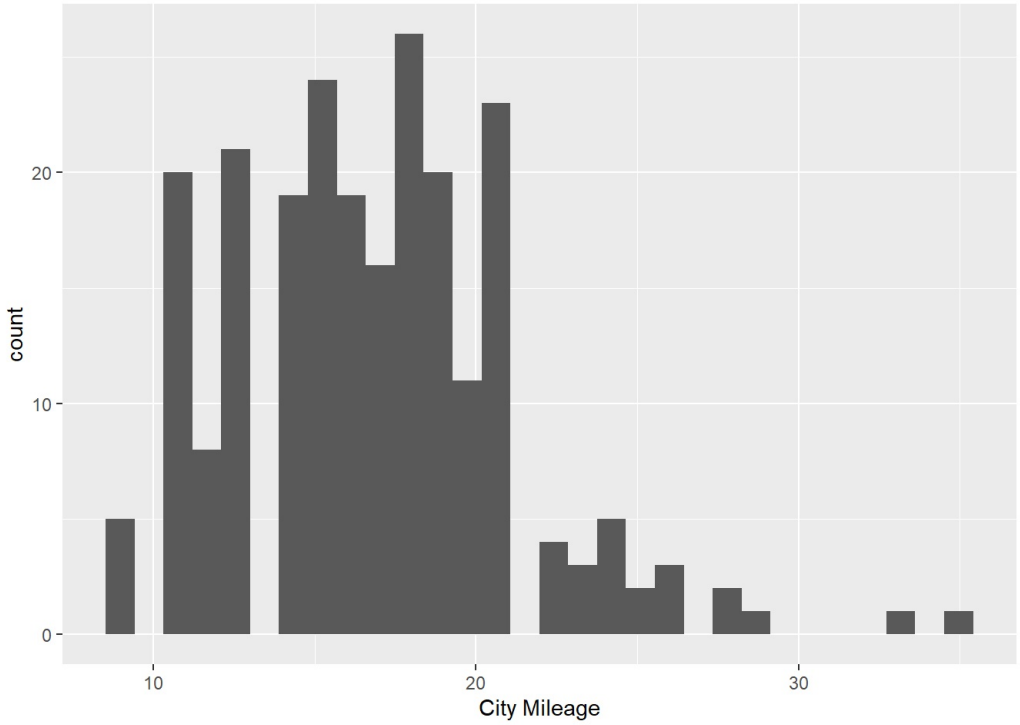
In this section, we use the filter() function from dplyr to subset the data based on specific conditions. We filter the mpg dataset to include only rows where city mileage (cty) is greater than or equal to 20 and create a new dataset mpg_efficient with these filtered rows. We also create another dataset mpg_ford by filtering rows where the manufacturer is “ford”.

```
# Adding new column
mpg_metric <- mutate(mpg, cty_metric = 0.425144 * cty)

# Introducing pipe operators
mpg_metric <- mpg %>%
  mutate(cty_metric = 0.425144 * cty)

mpg %>%
  group_by(class) %>%
  summarise(mean(cty), median(cty))
```

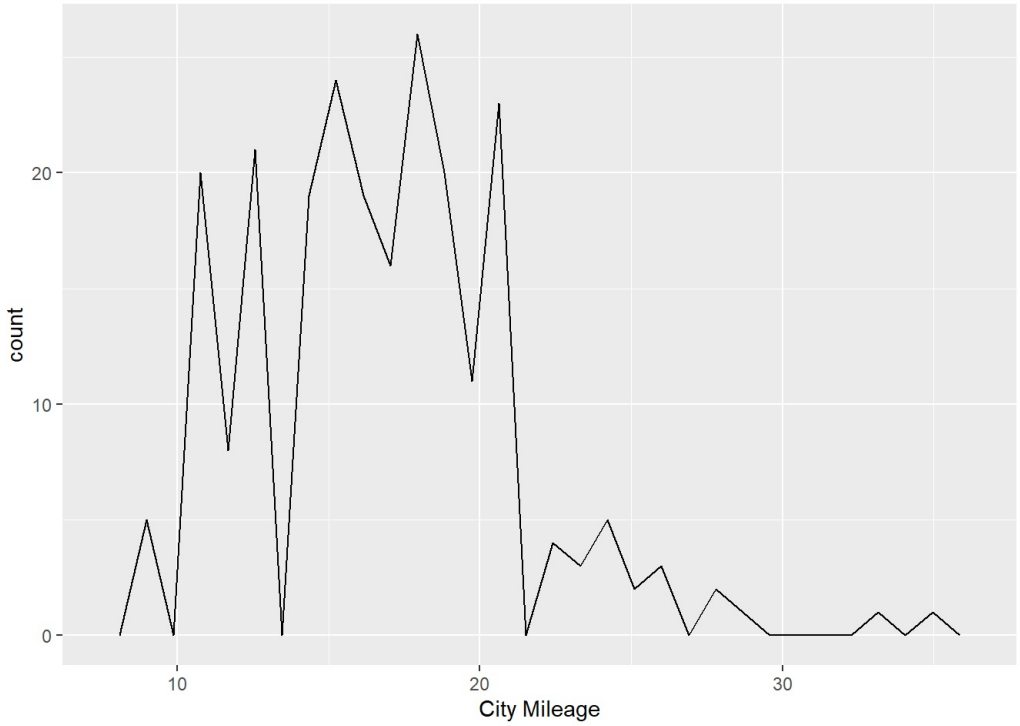
```
## # A tibble: 7 × 3
##   class      `mean(cty)` `median(cty)`
##   <chr>            <dbl>         <dbl>
## 1 2seater           15.4            15
## 2 compact           20.1            20
## 3 midsize           18.8            18
## 4 minivan           15.8            16
## 5 pickup            13              13
## 6 subcompact        20.4            19
## 7 suv               13.5            13
```

Here, we add a new column cty_metric to the mpg dataset by multiplying the cty column by a conversion factor. We demonstrate two ways of achieving this: using the mutate() function directly and using pipe operators (%>%). Additionally, we group the data by the class variable and calculate the mean and median city mileage for each class.

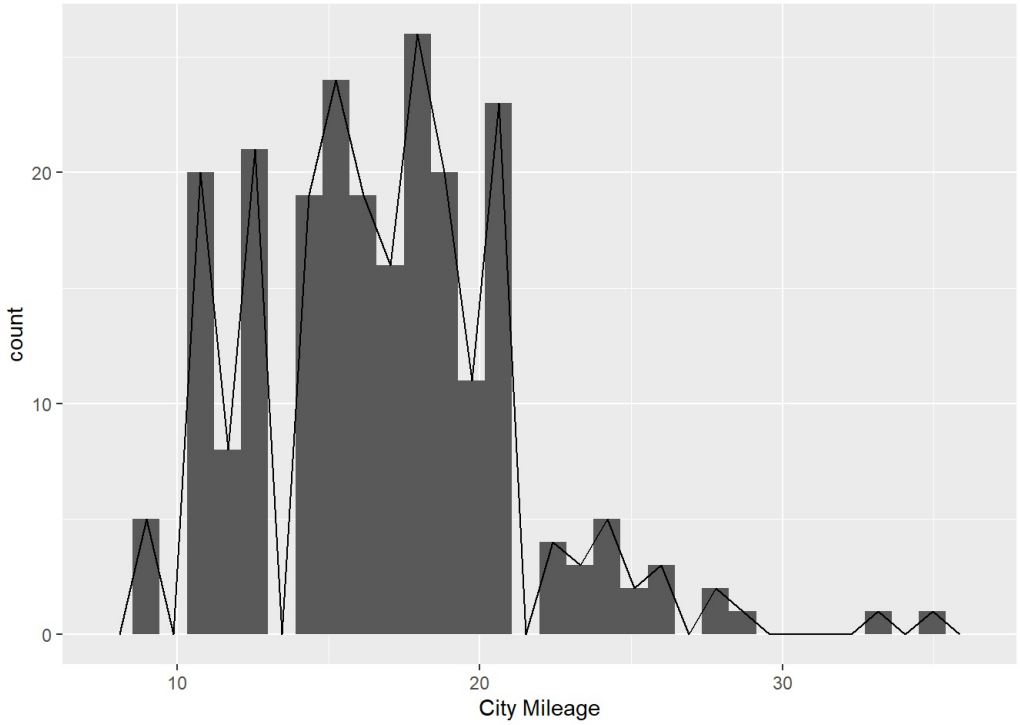
```
# Data visualization with ggplot
ggplot(mpg, aes(x = cty) )+
  geom_histogram()+
  labs(x = "City Mileage")
```



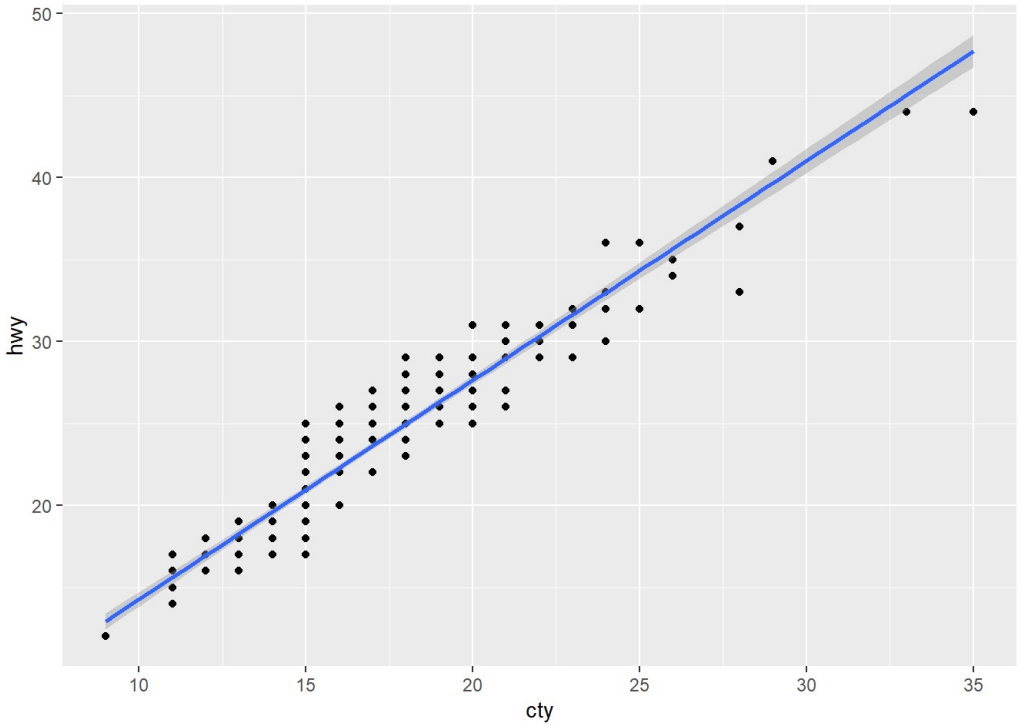

```
ggplot(mpg, aes(x = cty) )+
  geom_freqpoly()+
  labs(x = "City Mileage")
```



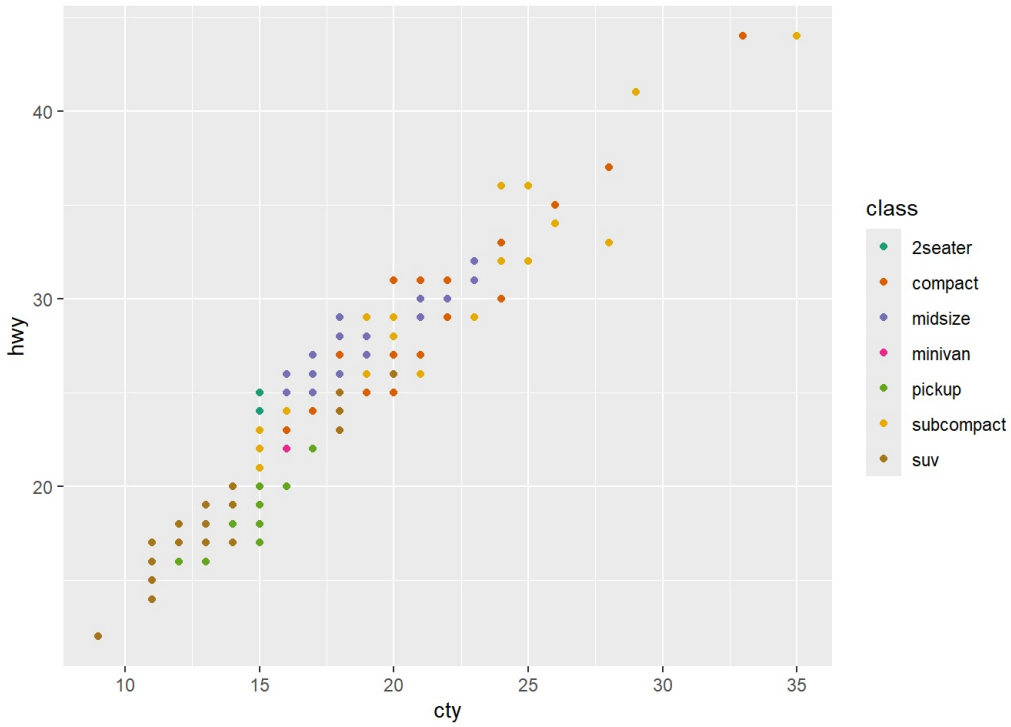

```
ggplot(mpg, aes(x = cty) )+
  geom_histogram()+
  geom_freqpoly()+
  labs(x = "City Mileage")
```




```
ggplot(mpg, aes(x = cty,
                y = hwy) )+
  geom_point()+
  geom_smooth(method = "lm")
```




```
ggplot(mpg, aes(x = cty,
                y = hwy,
                color = class) )+
  geom_point()+
  scale_color_brewer(palette = "Dark2")
```



Finally, we create several data visualizations using the ggplot2 package. We generate histograms, frequency polygons, scatter plots, and linear regression plots to explore the relationship between variables in the mpg dataset.