RWorksheet_Lapso4c.Rmd

Darlene Erl Lapso

2023-12-02

a. Show your solutions on how to import a csv file into the environment.

```
library(csv)

data(mpg)
```

```
## Warning in data(mpg): data set 'mpg' not found
```

```
getwd()
```

```
## [1] "C:/Users/steve/Documents/lapso-worksheetactivity/worksheet#4"
```

```
setwd("C:/Users/steve/Documents/lapso-worksheetactivity/worksheet#4")
mpg <- read.csv("D:/darlene/CS101/mpg.csv")
head(mpg)
```

```
##   X manufacturer model displ year cyl      trans drv cty hwy fl   class
## 1 1         audi    a4   1.8 1999   4   auto(l5)   f  18  29  p compact
## 2 2         audi    a4   1.8 1999   4 manual(m5)   f  21  29  p compact
## 3 3         audi    a4   2.0 2008   4 manual(m6)   f  20  31  p compact
## 4 4         audi    a4   2.0 2008   4   auto(av)   f  21  30  p compact
## 5 5         audi    a4   2.8 1999   6   auto(l5)   f  16  26  p compact
## 6 6         audi    a4   2.8 1999   6 manual(m5)   f  18  26  p compact
```

b. Which variables from mpg dataset are categorical?

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked _by_ '.GlobalEnv':
##
##     mpg
```

```
data(mpg)
str(mpg)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr [1:234] "p" "p" "p" "p" ...
##  $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

#1b. 'manufacturer' shows different kinds of vehicle manufacturer, 'model' shows different model of a vehicles 'drv' shows different types of drive (e.g., front-wheel drive, rear-wheel drive) 'fl' shows fuel types used by vehicles.

c. Which are continuous variables? #1c in the mpg dataset, the continuous variables are those shown as numbers (like engine displacement in liters for 'displ'). Additionally, there are other number-based variables like 'hwy' and 'cty' (representing miles per gallon on the highway and in the city), along with 'year'. However, these numeric variables might not be purely continuous; some, like 'year', could represent categories or ordered values rather than a smooth range of numbers.

2.1 Which manufacturer has the most models in this data set? Which model has the most variations? Show your answer

```
mostManu <- names(sort(table(mpg$manufacturer), decreasing = TRUE))[1]
mostVar <- names(sort(table(mpg$model), decreasing = TRUE))[1]
```

```
mostManu
```

```
## [1] "dodge"
```

```
mostVar
```

```
## [1] "caravan 2wd"
```

a. Group the manufacturers and find the unique models. Show your codes and result

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

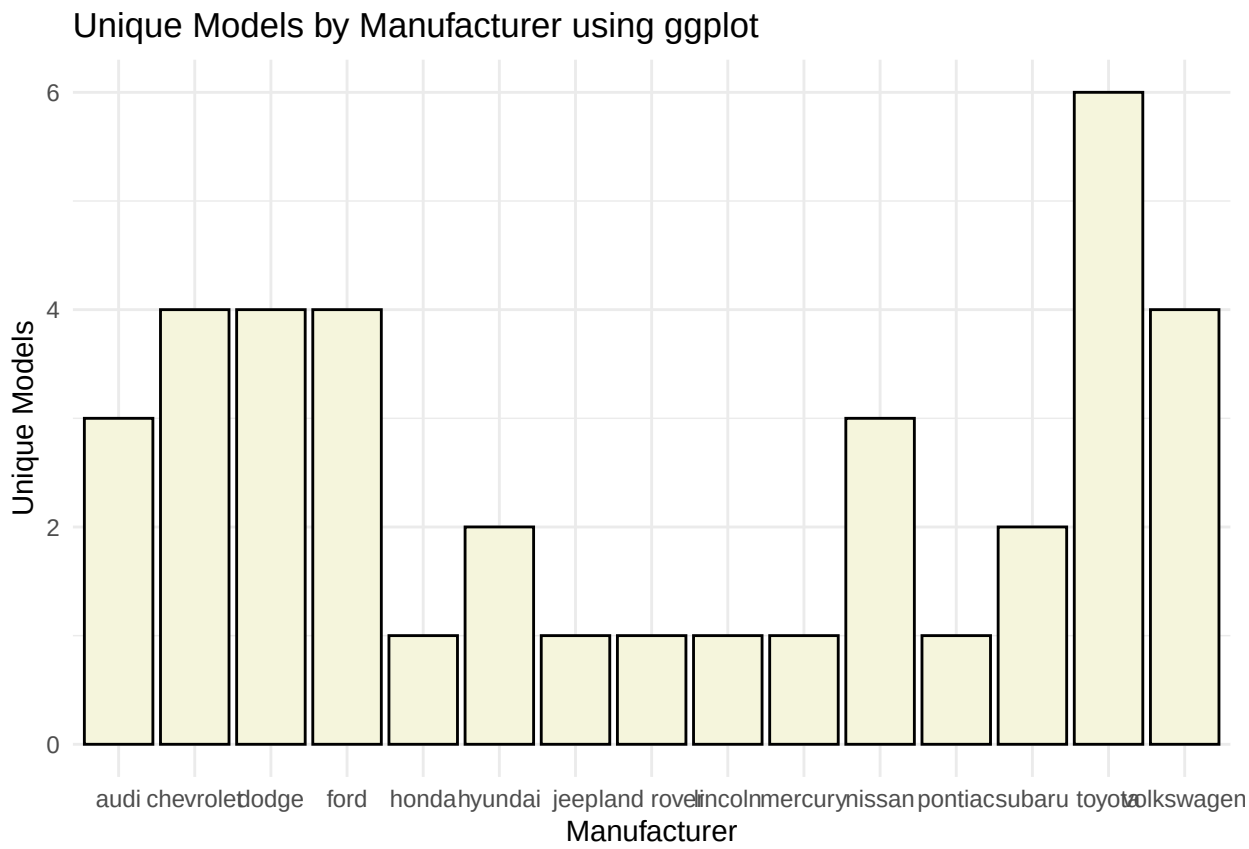
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
uniqueModMan <- mpg %>%
  group_by(manufacturer) %>%
  summarize(unique_models = n_distinct(model))
uniqueModMan
```

```
## # A tibble: 15 x 2
##    manufacturer unique_models
##    <chr>                <int>
##  1 audi                     3
##  2 chevrolet                4
##  3 dodge                    4
##  4 ford                     4
##  5 honda                    1
##  6 hyundai                  2
##  7 jeep                     1
##  8 land rover               1
##  9 lincoln                  1
## 10 mercury                  1
## 11 nissan                   3
## 12 pontiac                  1
## 13 subaru                   2
## 14 toyota                   6
## 15 volkswagen               4
```

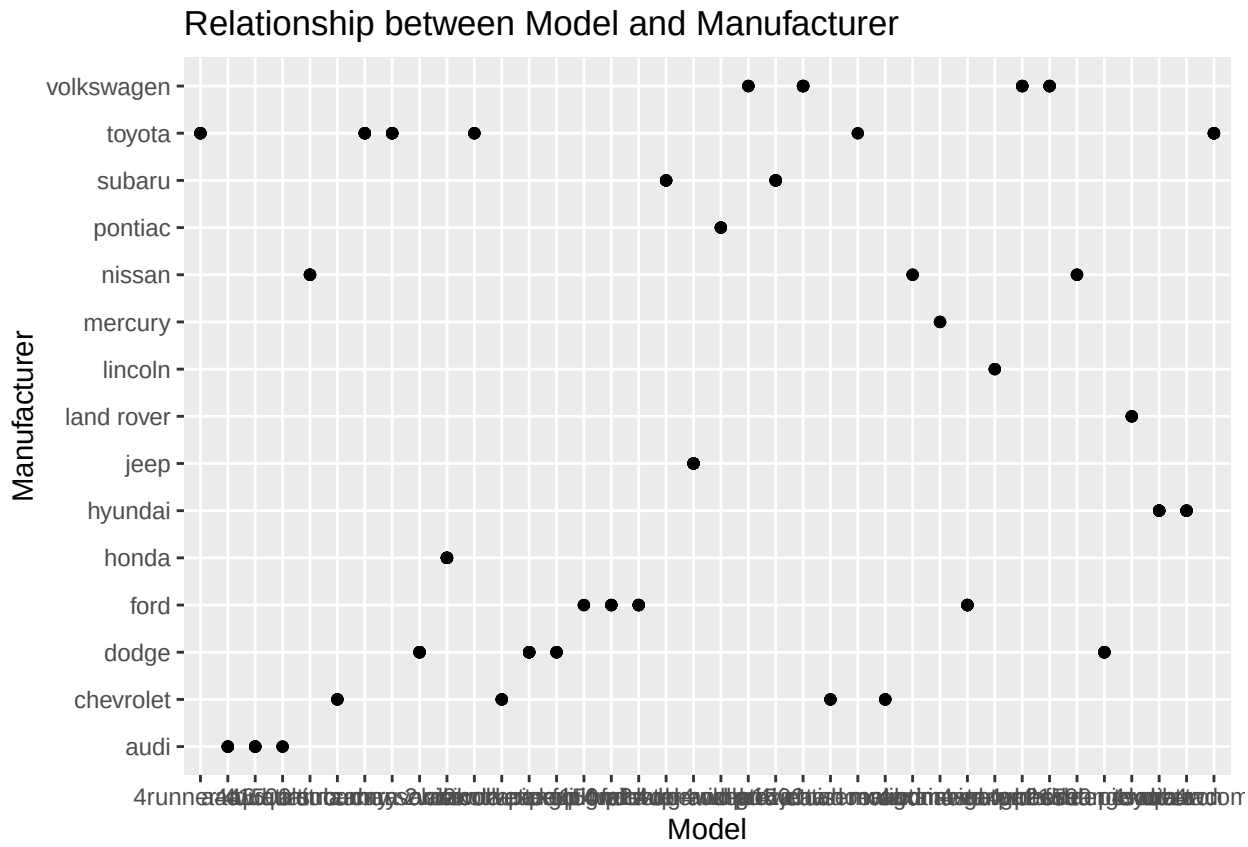
b. Graph the result by using plot() and ggplot(). Write the codes and its result

```
library(ggplot2)
ggplot(uniqueModMan, aes(x = manufacturer, y = unique_models)) +
  geom_bar(stat = "identity", fill = "beige", color = "black") +
  labs(x = "Manufacturer", y = "Unique Models",
       title = "Unique Models by Manufacturer using ggplot") +
  theme_minimal()
```



2.2 a. What does ggplot(mpg, aes(model, manufacturer)) + geom_point() show?

```
library(ggplot2)
pointsMPG <- ggplot(mpg, aes(model, manufacturer)) +
  geom_point() +
  labs(x = "Model", y = "Manufacturer",
       title = "Relationship between Model and Manufacturer")
pointsMPG
```



This plot could display how car models are spread out among various manufacturers or how particular models are distributed across different manufacturers. Yet, due to potential overlapping points or the categorical nature of the variables, it might not offer a clear or meaningful representation of the direct relationship between the model and the manufacturer.

b. For you, is it useful? If not, how could you modify the data to make it more informative?

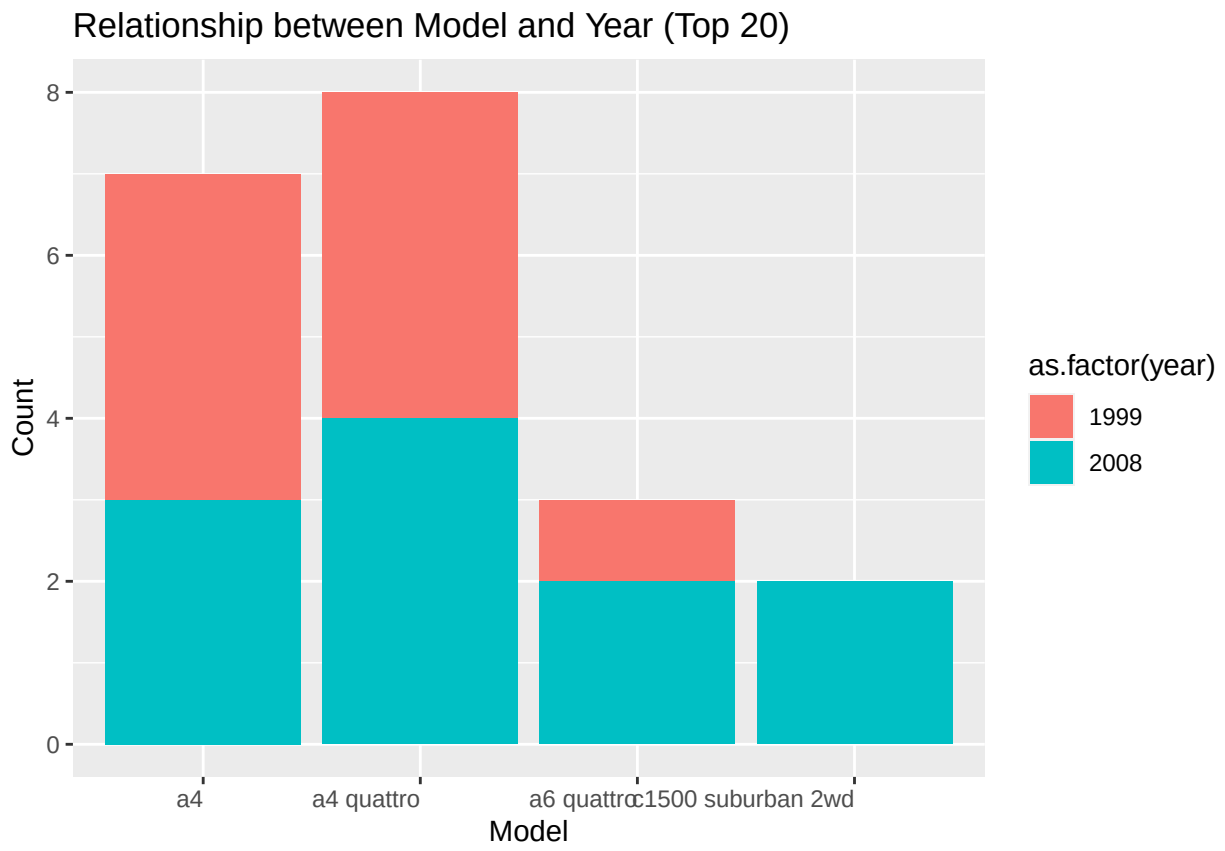
I think the plot isn't very helpful because it might have overlapping points, making it hard to see how car models relate to manufacturers.

What I could do to do it more is to group/combine data or use a different geom graph.

---

3. Plot the model and the year using ggplot(). Use only the top 20 observations. Write the codes and its results.

```
library(ggplot2)
top_20 <- head(mpg, 20)
ggplot(top_20, aes(x = model, fill = as.factor(year))) +
  geom_bar() +
  labs(x = "Model", y = "Count",
       title = "Relationship between Model and Year (Top 20)") +
  theme(axis.text.x = element_text(angle = 360, hjust = 1))
```



I used geom_bar, for an attainable and understable summary of visualization.

4. Using the pipe (%>%), group the model and get the number of cars per model. Show codes and its result

```
library(dplyr)

cars_per_model <- mpg %>%
  group_by(model) %>%
  summarize(num_cars = n())

cars_per_model
```

```
## # A tibble: 38 x 2
##    model              num_cars
##    <chr>                 <int>
##  1 4runner 4wd               6
##  2 a4                        7
##  3 a4 quattro                8
##  4 a6 quattro                3
##  5 altima                    6
##  6 c1500 suburban 2wd        5
##  7 camry                     7
##  8 camry solara              7
```

```
##  9 caravan 2wd              11
## 10 civic                     9
## # i 28 more rows
```

a. Plot using geom_bar() using the top 20 observations only. The graphs should have a title, labels and colors. Show code and results.

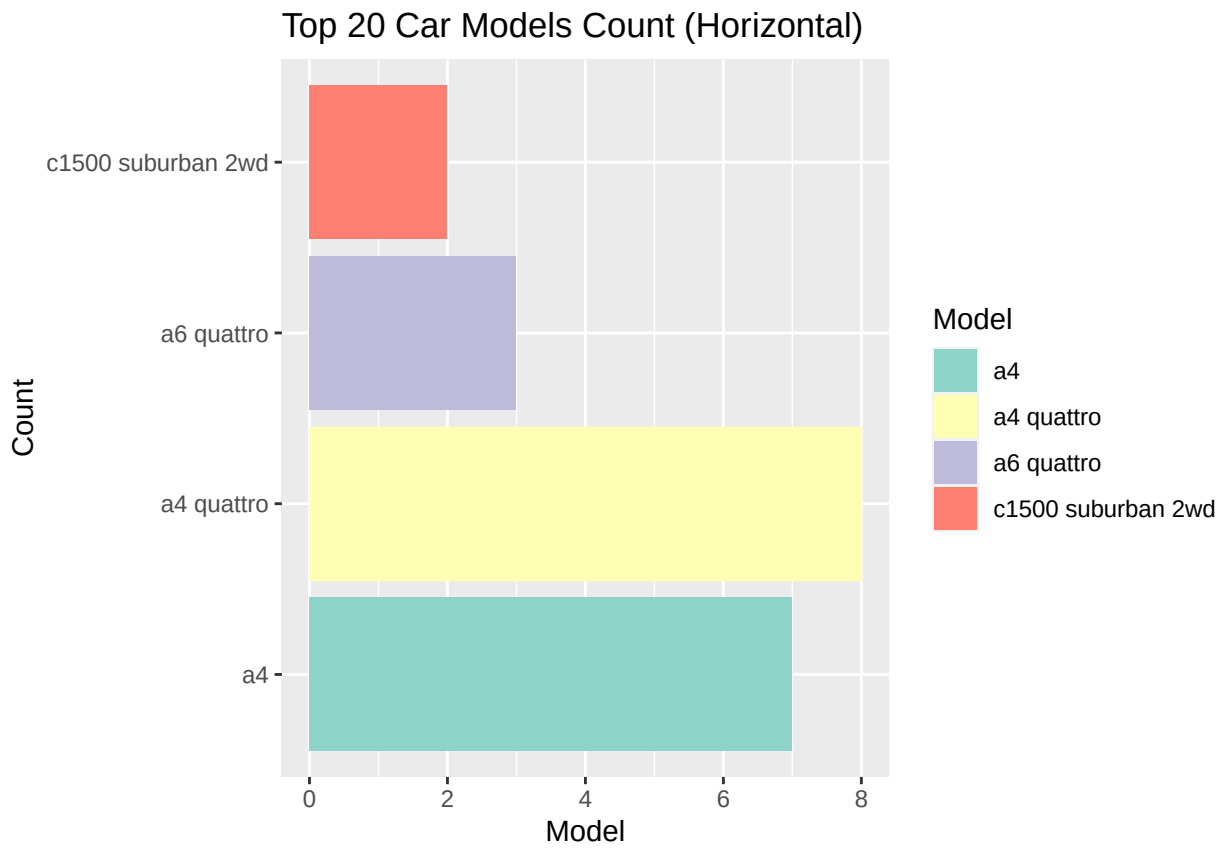
```r
library(ggplot2)

top_20 <- head(mpg, 20)

top_20 <- ggplot(top_20, aes(x = model, fill = model)) +
  geom_bar() +
  labs(x = "Model", y = "Count", fill = "Model",
       title = "Top 20 Car Models Count") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +  # Rotate x-axis labels
  scale_fill_brewer(palette = "Set3")
```

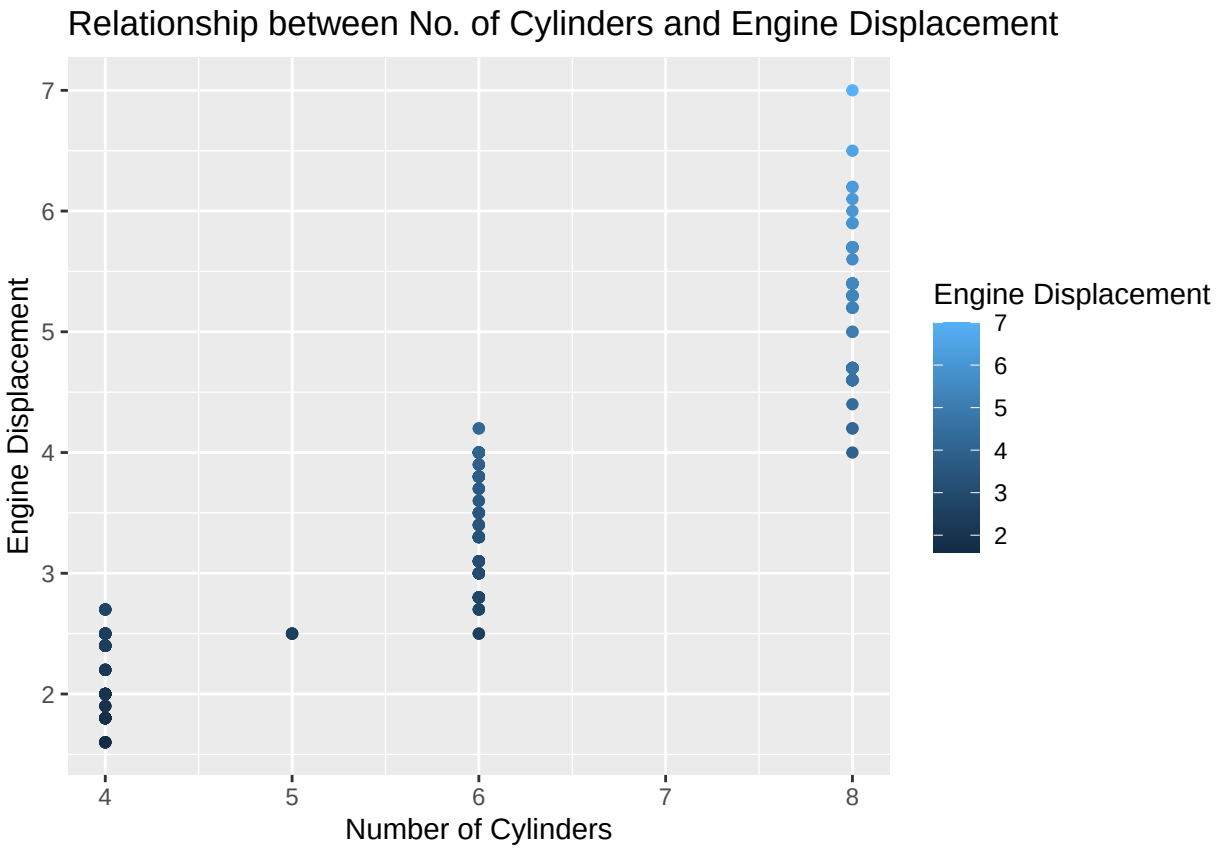
b. Plot using the geom_bar() + coord_flip() just like what is shown below. Show codes and its result.

```r
top_20 <- head(mpg, 20)
ggplot(top_20, aes(x = model, fill = model)) +
  geom_bar() +
  labs(x = "Count", y = "Model", fill = "Model",
       title = "Top 20 Car Models Count (Horizontal)") +
  coord_flip() +
  theme(axis.text.y = element_text(hjust = 1)) +
  scale_fill_brewer(palette = "Set3")
```



5. Plot the relationship between cyl - number of cylinders and displ - engine displacement using geom_point with aesthetic color = engine displacement. Title should be "Relationship between No. of Cylinders and Engine Displacement".

```
ggplot(mpg, aes(x = cyl, y = displ, color = displ)) +
  geom_point() +
  labs(x = "Number of Cylinders", y = "Engine Displacement",
       title = "Relationship between No. of Cylinders and Engine Displacement") +
  scale_color_continuous(name = "Engine Displacement")
```



When working with categorical variables like 'model' and 'year', a bar plot might not effectively illustrate their direct relationship. However, by displaying the counts for each unique pairing of 'model' and 'year', we can identify patterns or frequencies within this constrained data set. e