RWorkshet_Garcia4c

Carol D. Garcia

2025-12-10

```
library(ggplot2)
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
```

```
library(readxl)

# 1.
write.csv(mpg, "mpg.csv", row.names = FALSE)
mpg_tbl <- read.csv("mpg.csv", header = TRUE)
str(mpg_tbl)
```

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

```
cat_vars <- names(mpg_tbl)[sapply(mpg_tbl, function(x) is.character(x) | is.factor(x))]
cat_vars
```

```
## [1] "manufacturer" "model"        "trans"        "drv"          "fl"
## [6] "class"
```

```
num_vars <- names(mpg_tbl)[sapply(mpg_tbl, is.numeric)]
num_vars
```

```
## [1] "displ" "year"  "cyl"   "cty"   "hwy"
```

```r
# 2a.
mpg_data <- mpg

manuf_summary <- mpg_data %>%
  group_by(manufacturer) %>%
  summarise(model_count = n_distinct(model)) %>%
  arrange(desc(model_count))

print("Unique models per manufacturer:")
```

```
## [1] "Unique models per manufacturer:"
```

```r
print(manuf_summary)
```

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

```r
top_manuf <- manuf_summary$manufacturer[1]
print(paste("Manufacturer with most models:", top_manuf))
```

```
## [1] "Manufacturer with most models: toyota"
```

```r
model_summary <- mpg_data %>%
  group_by(model) %>%
  summarise(variant_count = n()) %>%
  arrange(desc(variant_count))

print("Model variations:")
```

```
## [1] "Model variations:"
```

```r
print(model_summary)
```

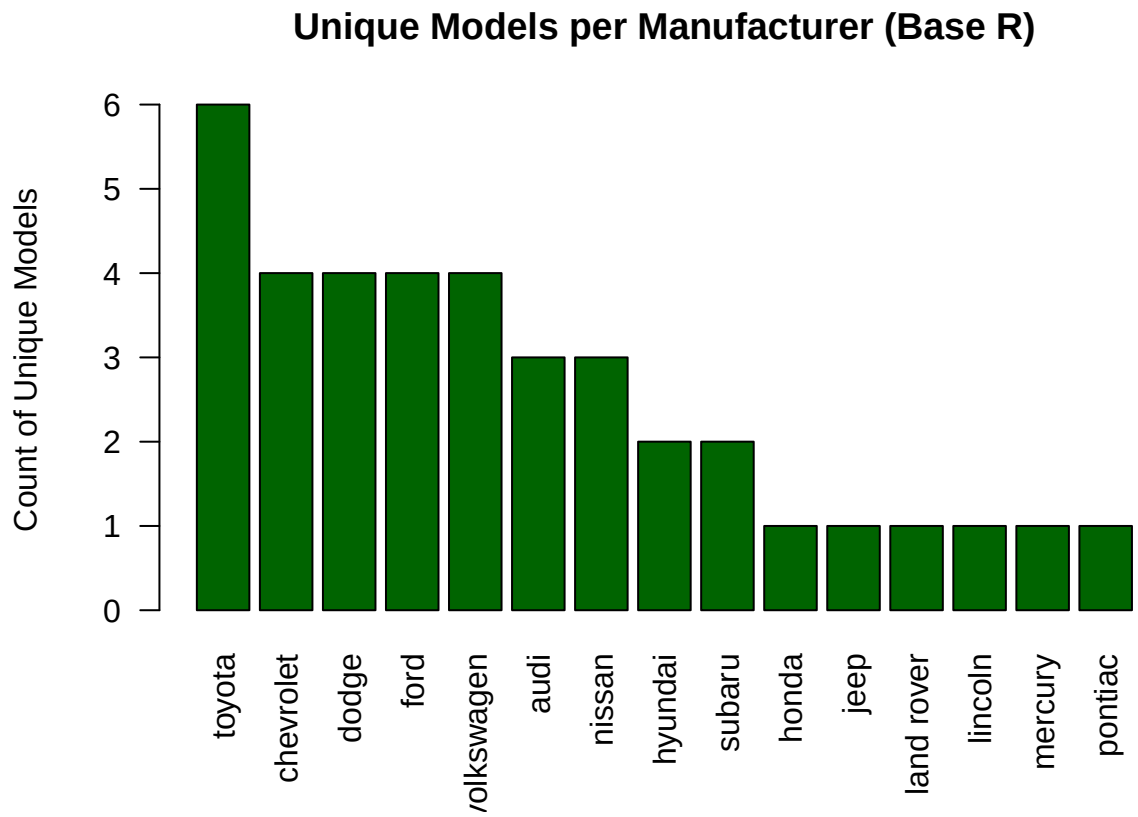
```
## # A tibble: 38 x 2
##    model                variant_count
##    <chr>                        <int>
##  1 caravan 2wd                     11
##  2 ram 1500 pickup 4wd             10
##  3 civic                            9
##  4 dakota pickup 4wd                9
##  5 jetta                            9
```

```
##  6 mustang                          9
##  7 a4 quattro                       8
##  8 grand cherokee 4wd               8
##  9 impreza awd                      8
## 10 a4                               7
## # i 28 more rows
```

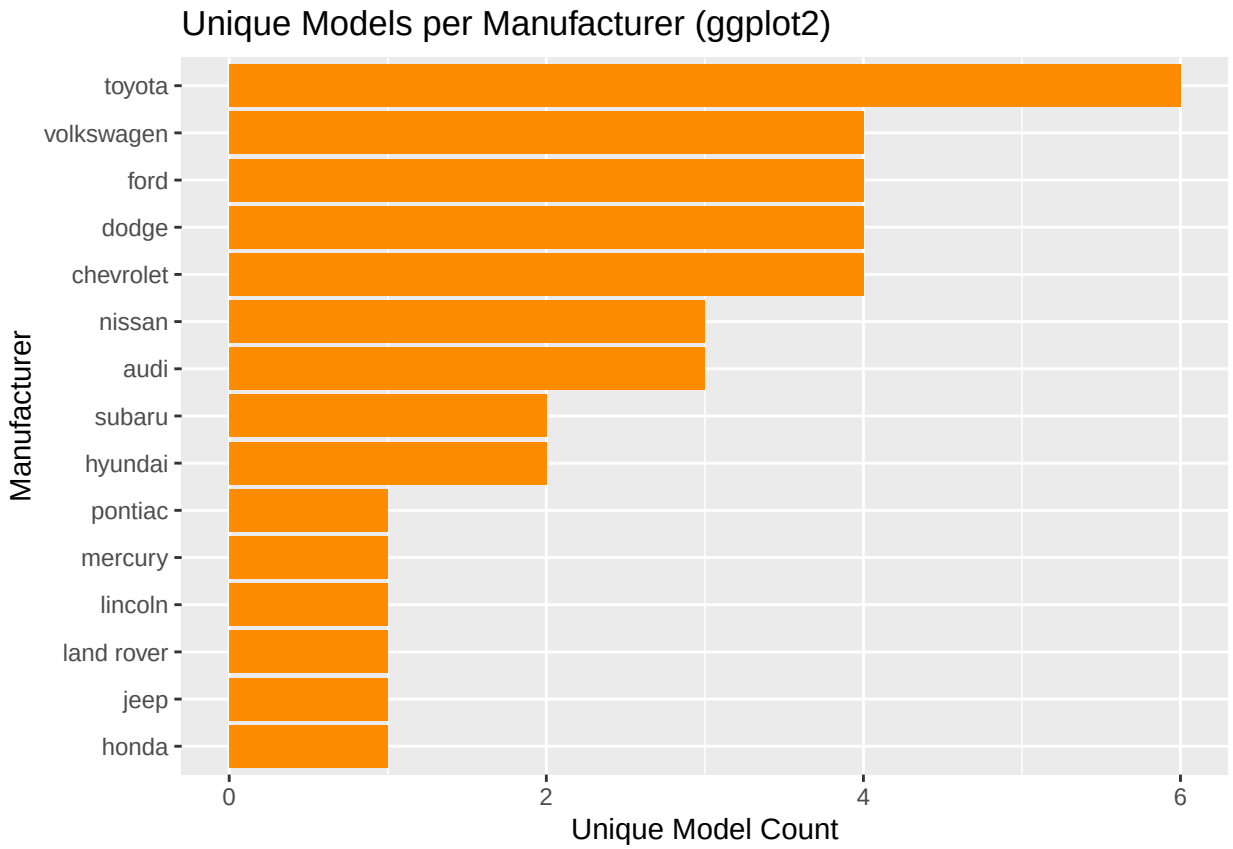
```r
top_model <- model_summary$model[1]
print(paste("Model with most variations:", top_model))
```

```
## [1] "Model with most variations: caravan 2wd"
```

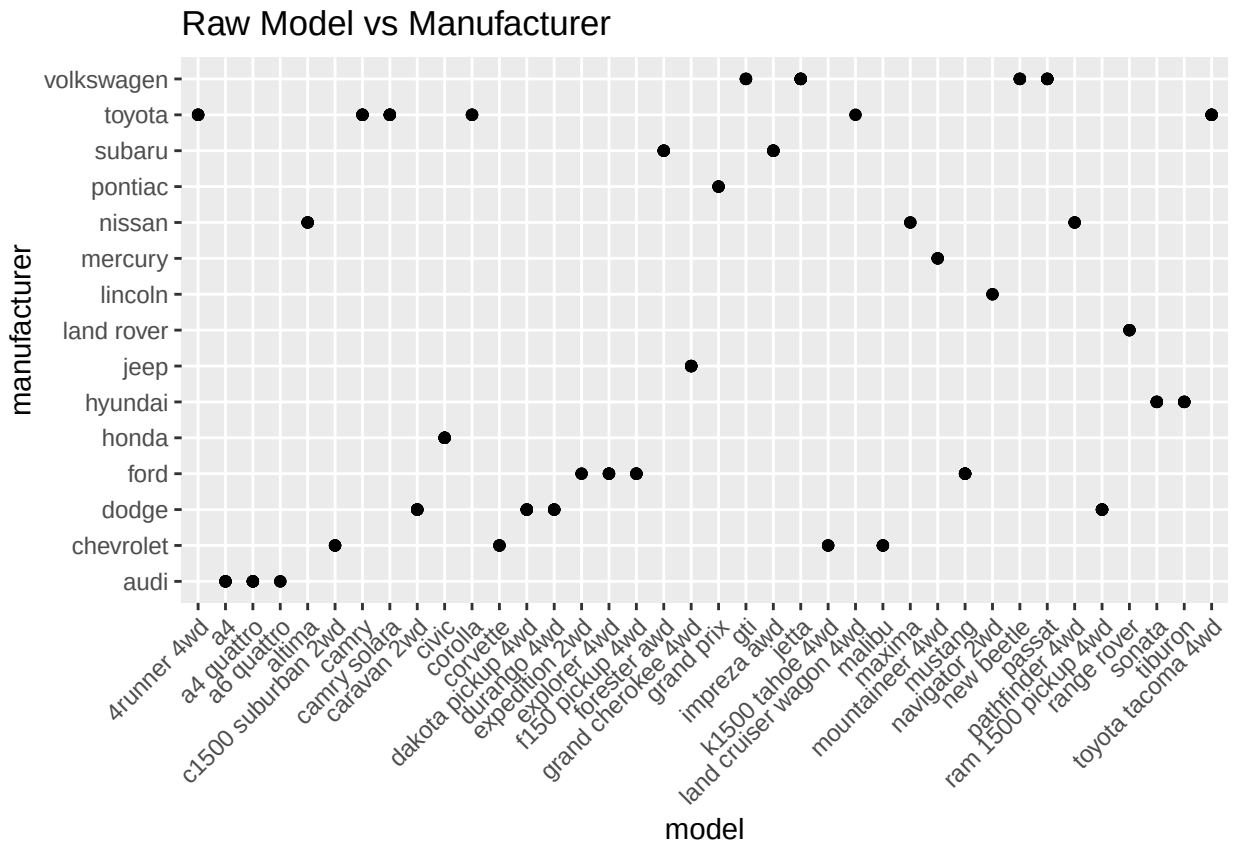
```r
barplot(height = manuf_summary$model_count,
        names.arg = manuf_summary$manufacturer,
        col = "darkgreen",
        las = 2,
        main = "Unique Models per Manufacturer (Base R)",
        ylab = "Count of Unique Models")
```




```r
ggplot(manuf_summary, aes(x = reorder(manufacturer, model_count), y = model_count)) +
  geom_col(fill = "darkorange") +
  coord_flip() +
  labs(title = "Unique Models per Manufacturer (ggplot2)",
       x = "Manufacturer",
       y = "Unique Model Count")
```



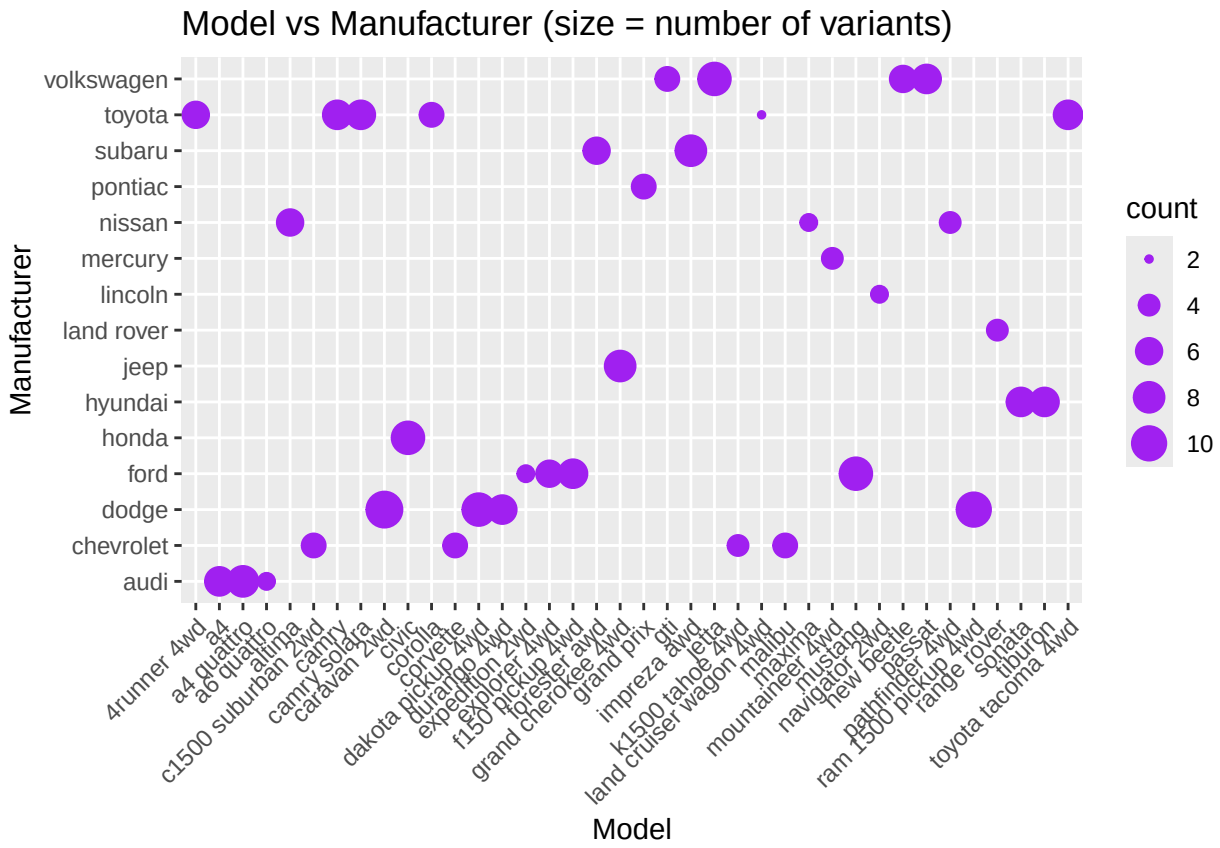

```r
# 2b.
ggplot(mpg_data, aes(x = model, y = manufacturer)) +
  geom_point() +
  labs(title = "Raw Model vs Manufacturer") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



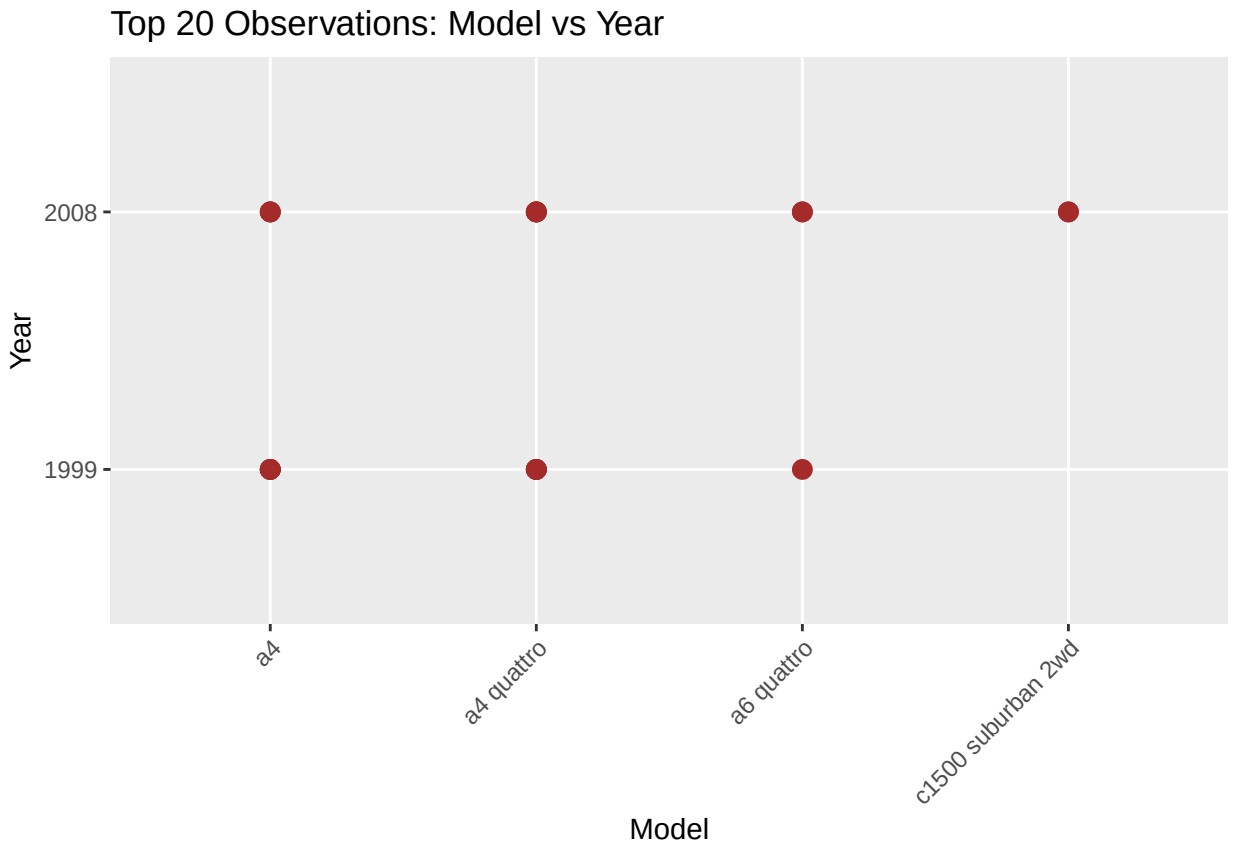

```
model_variants <- mpg_data %>%
  group_by(manufacturer, model) %>%
  summarise(count = n()) %>%
  ungroup()
```

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

```
ggplot(model_variants, aes(x = model, y = manufacturer, size = count)) +
  geom_point(color = "purple") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Model vs Manufacturer (size = number of variants)",
       x = "Model",
       y = "Manufacturer")
```




```r
# 3.
top_20 <- mpg_data %>% slice(1:20)

ggplot(top_20, aes(x = model, y = factor(year))) +
  geom_point(color = "brown", size = 3) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Top 20 Observations: Model vs Year",
       x = "Model",
       y = "Year")
```



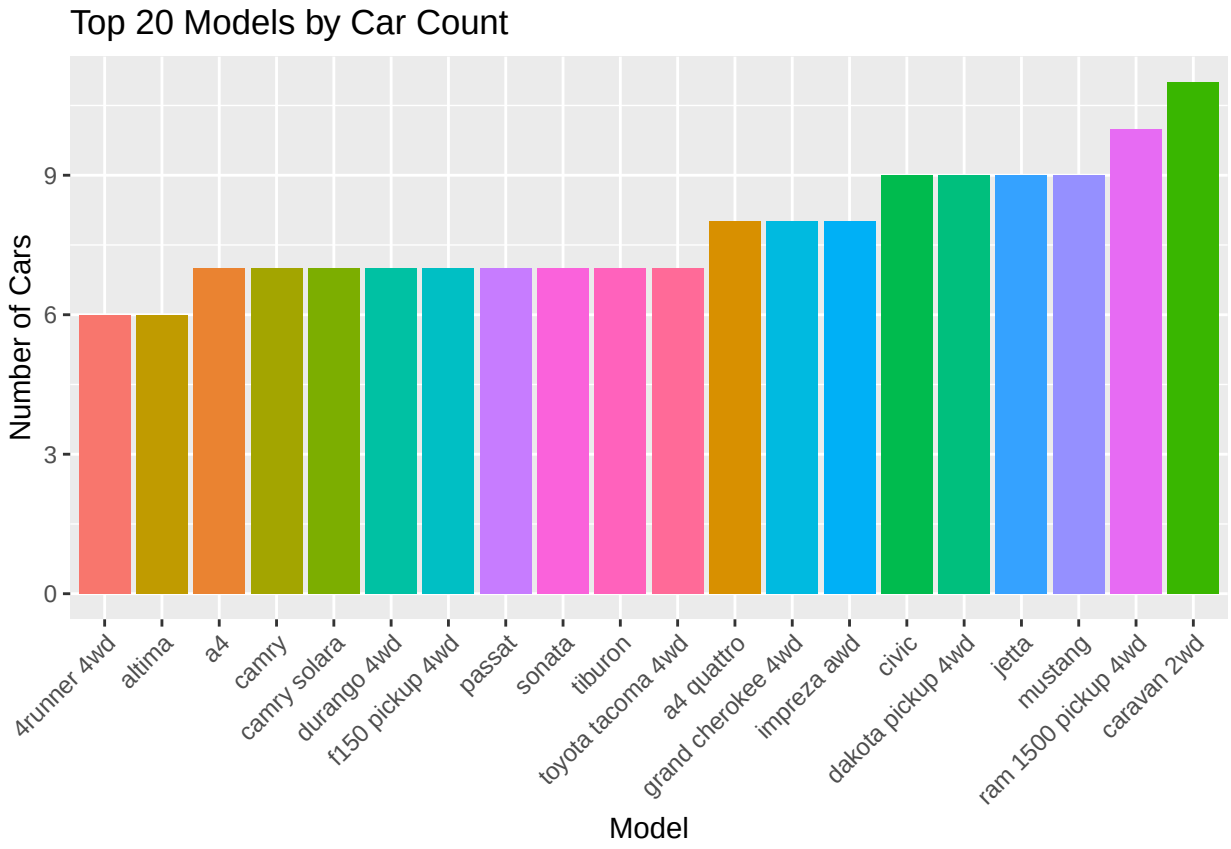

```
# 4.
model_freq <- mpg_data %>%
  group_by(model) %>%
  summarise(car_count = n()) %>%
  arrange(desc(car_count))

print(model_freq)
```

```
## # A tibble: 38 x 2
##    model               car_count
##    <chr>                   <int>
##  1 caravan 2wd                11
##  2 ram 1500 pickup 4wd        10
##  3 civic                       9
##  4 dakota pickup 4wd           9
##  5 jetta                       9
##  6 mustang                     9
##  7 a4 quattro                  8
##  8 grand cherokee 4wd          8
##  9 impreza awd                 8
## 10 a4                          7
## # i 28 more rows
```

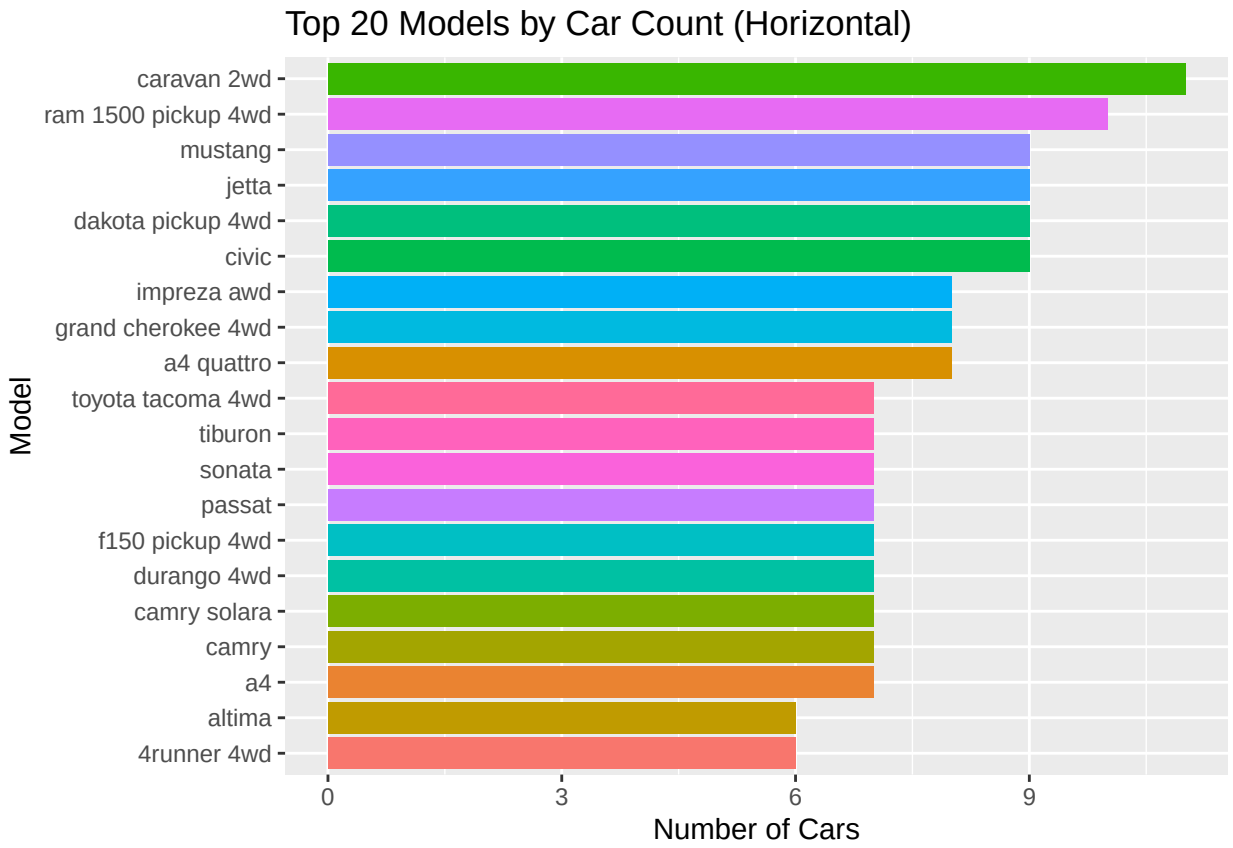
```
top20_models <- model_freq %>% slice(1:20)

ggplot(top20_models, aes(x = reorder(model, car_count), y = car_count, fill = model)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
```

```
  labs(title = "Top 20 Models by Car Count",
       x = "Model",
       y = "Number of Cars") +
  guides(fill = FALSE)
```

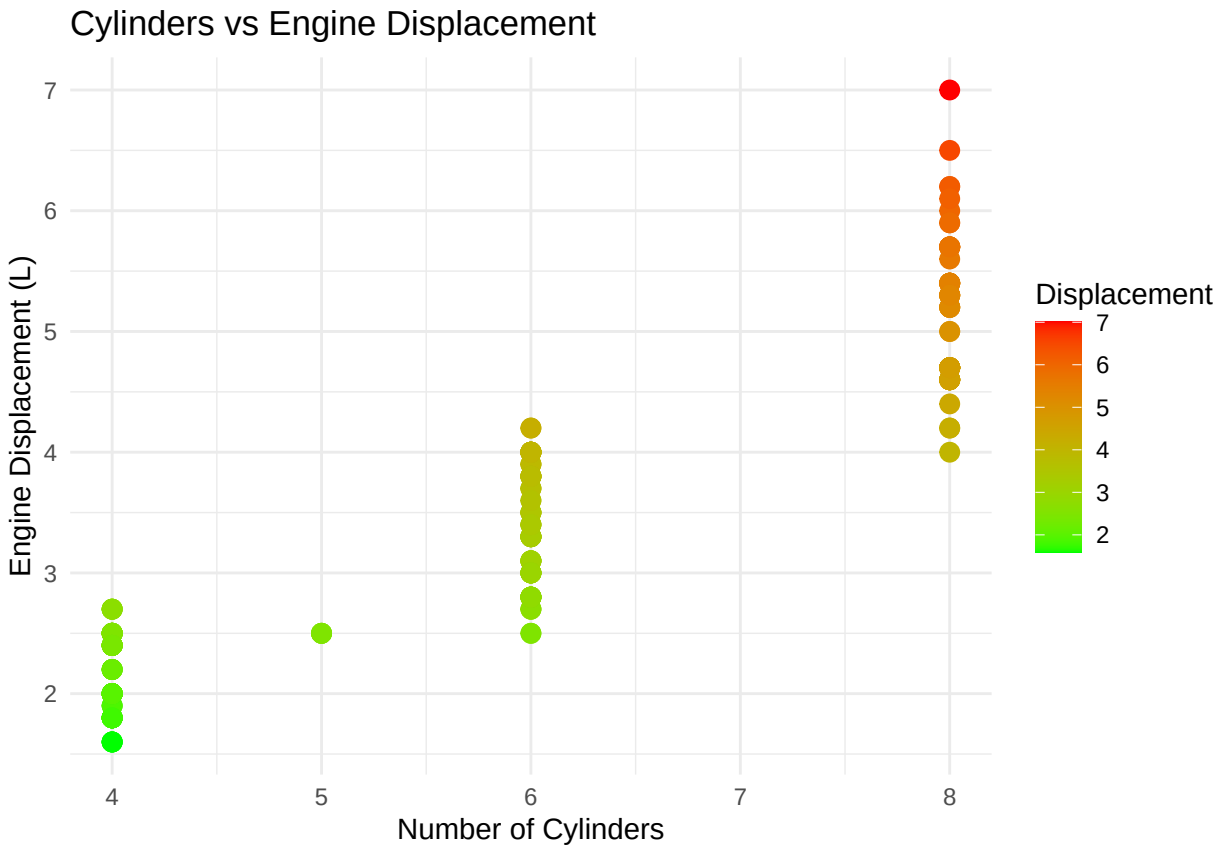
```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



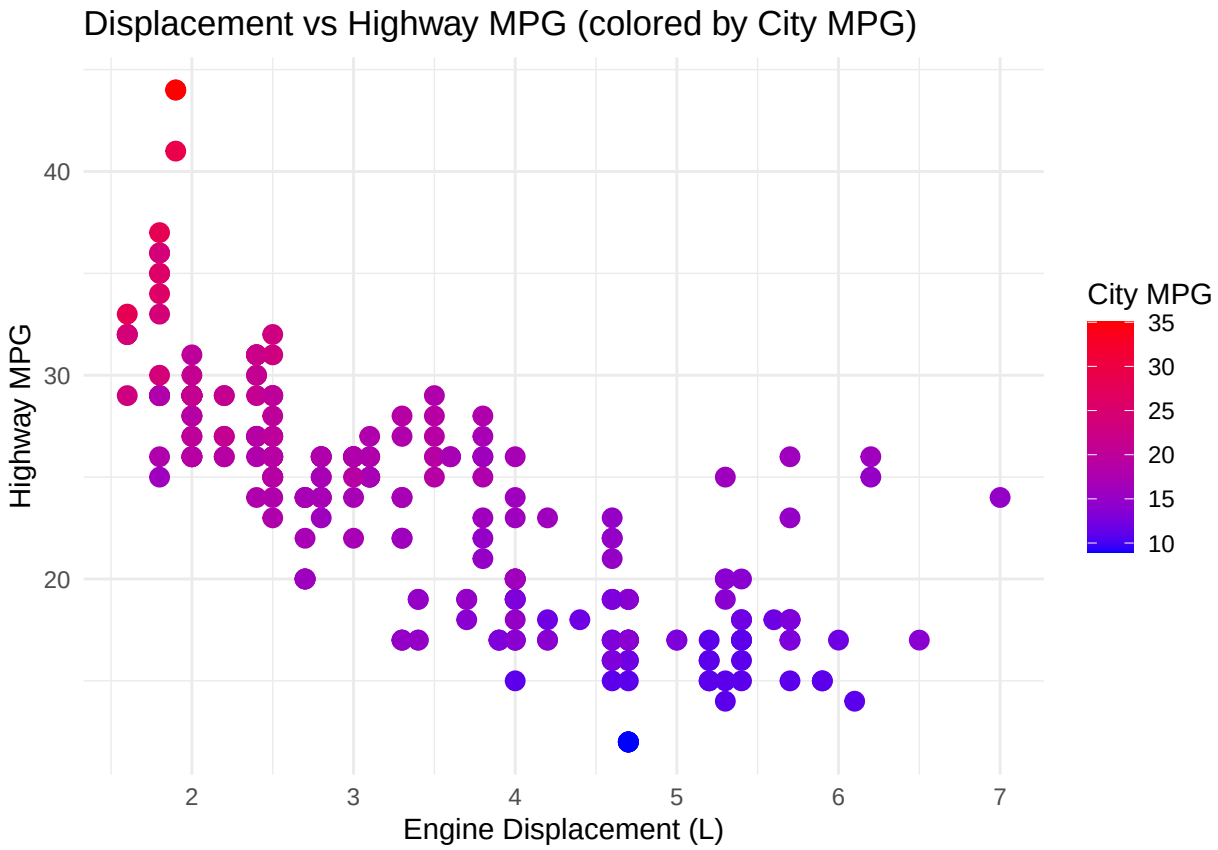

```
ggplot(top20_models, aes(x = reorder(model, car_count), y = car_count, fill = model)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top 20 Models by Car Count (Horizontal)",
       x = "Model",
       y = "Number of Cars") +
  guides(fill = FALSE)
```




```r
# 5.
ggplot(mpg_data, aes(x = cyl, y = displ, color = displ)) +
  geom_point(size = 3) +
  scale_color_gradient(low = "green", high = "red") +
  labs(title = "Cylinders vs Engine Displacement",
       x = "Number of Cylinders",
       y = "Engine Displacement (L)",
       color = "Displacement") +
  theme_minimal()
```




```r
# 6.
ggplot(mpg_data, aes(x = displ, y = hwy, color = cty)) +
  geom_point(size = 3) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "Displacement vs Highway MPG (colored by City MPG)",
       x = "Engine Displacement (L)",
       y = "Highway MPG",
       color = "City MPG") +
  theme_minimal()
```




```
traffic_tbl <- data.frame(
  Date = as.Date('2025-11-01') + 0:9,
  Junction = rep(c("Junction A", "Junction B"), each = 5),
  Vehicle_Count = c(120, 150, 130, 160, 140, 200, 210, 190, 205, 220),
  Avg_Speed = c(35.5, 34.2, 36.0, 33.8, 34.5, 32.0, 31.5, 33.0, 30.8, 29.5)
)

write.csv(traffic_tbl, "traffic.csv", row.names = FALSE)
traffic_data <- read.csv("traffic.csv", stringsAsFactors = FALSE)

cat("Number of observations:", nrow(traffic_data), "\n")
```

```
## Number of observations: 10
```

```
cat("Variables in the traffic dataset:\n")
```

```
## Variables in the traffic dataset:
```

```
print(names(traffic_data))
```

```
## [1] "Date"          "Junction"      "Vehicle_Count" "Avg_Speed"
```

```
junction_A_tbl <- traffic_data %>% filter(Junction == "Junction A")
junction_B_tbl <- traffic_data %>% filter(Junction == "Junction B")

print("Junction A data:")
```

```
## [1] "Junction A data:"
```

```r
print(junction_A_tbl)
```

```
##         Date   Junction Vehicle_Count Avg_Speed
## 1 2025-11-01 Junction A           120      35.5
## 2 2025-11-02 Junction A           150      34.2
## 3 2025-11-03 Junction A           130      36.0
## 4 2025-11-04 Junction A           160      33.8
## 5 2025-11-05 Junction A           140      34.5
```

```r
print("Junction B data:")
```

```
## [1] "Junction B data:"
```

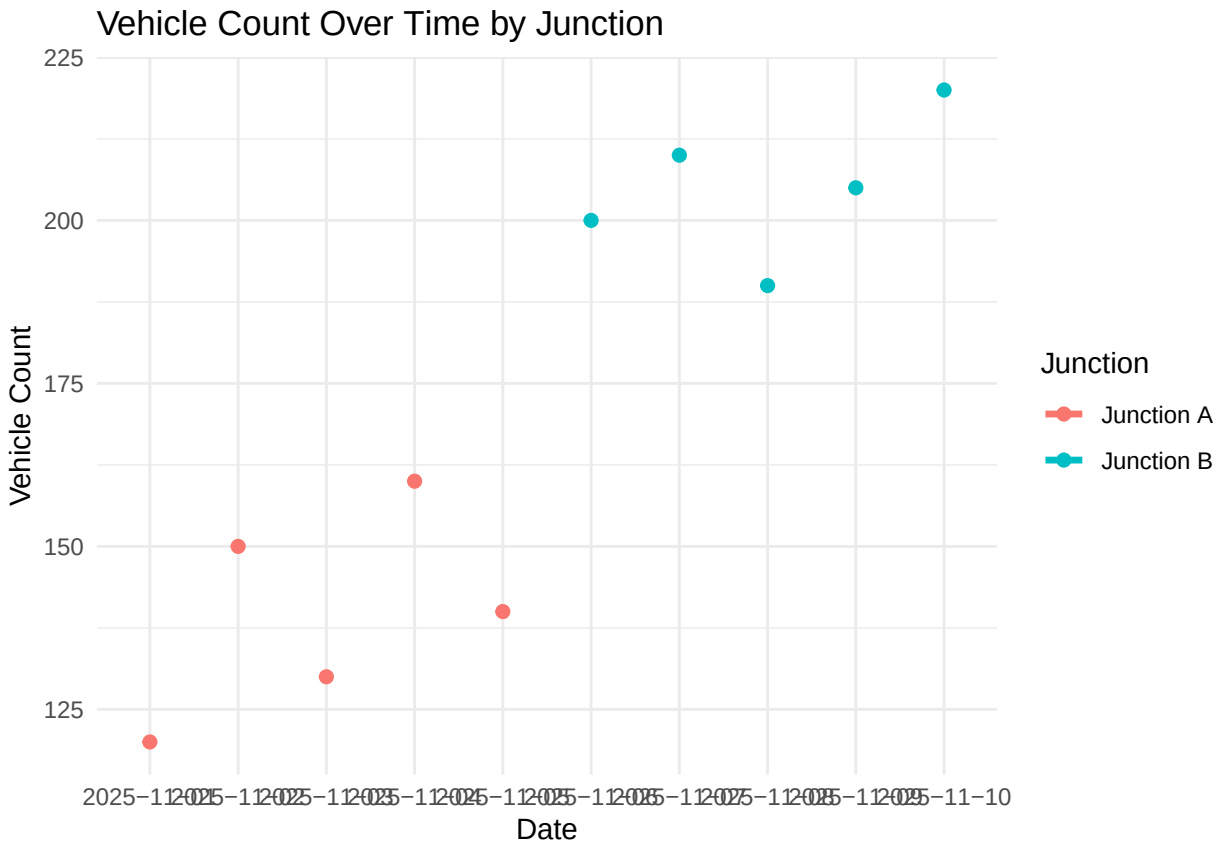
```r
print(junction_B_tbl)
```

```
##         Date   Junction Vehicle_Count Avg_Speed
## 1 2025-11-06 Junction B           200      32.0
## 2 2025-11-07 Junction B           210      31.5
## 3 2025-11-08 Junction B           190      33.0
## 4 2025-11-09 Junction B           205      30.8
## 5 2025-11-10 Junction B           220      29.5
```

```r
ggplot(traffic_data, aes(x = Date, y = Vehicle_Count, color = Junction)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  labs(title = "Vehicle Count Over Time by Junction",
       x = "Date",
       y = "Vehicle Count",
       color = "Junction") +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```

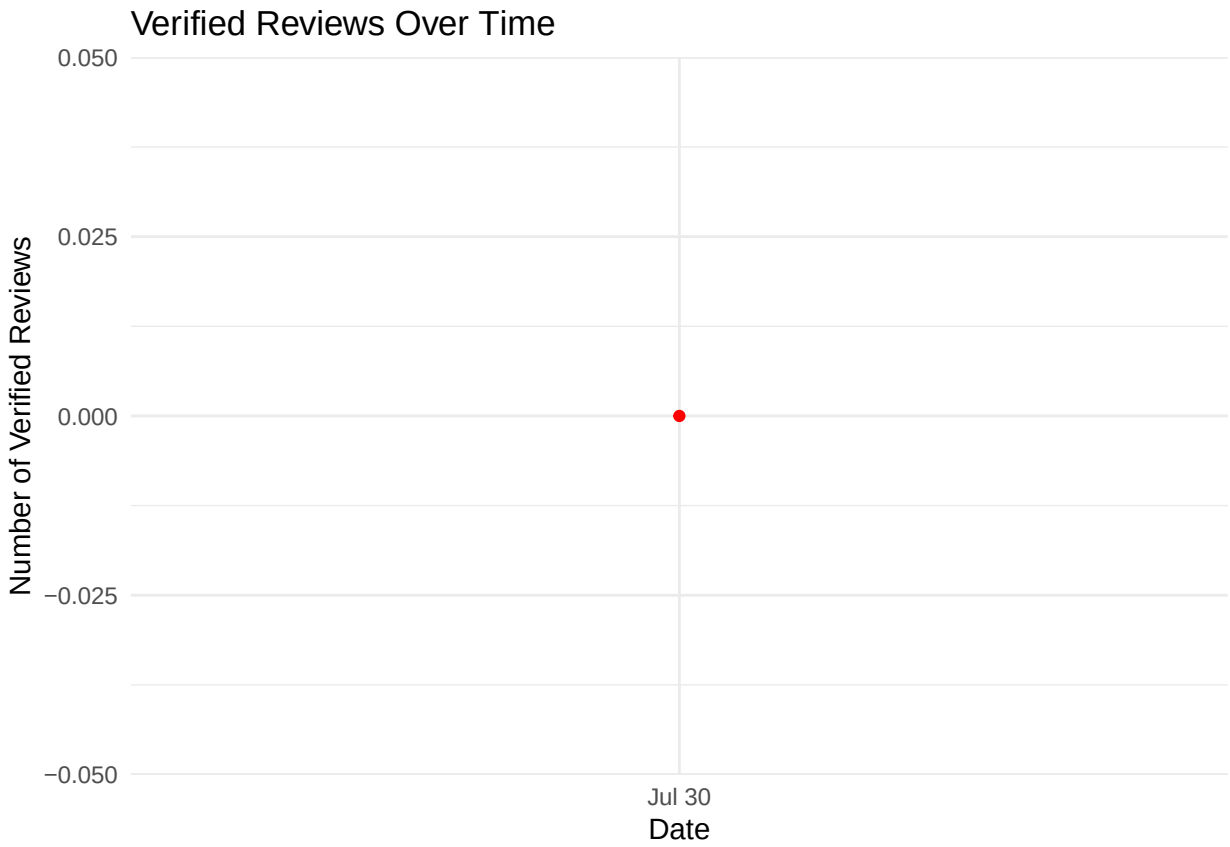



```
# 7.
alexa_data <- read_excel("alexa-file.xlsx")

alexa_data <- alexa_data %>%
  mutate(
    verified_reviews_num = as.numeric(verified_reviews),
    avg_rating = as.numeric(rating),
    review_date = as.Date(date)
  )
```

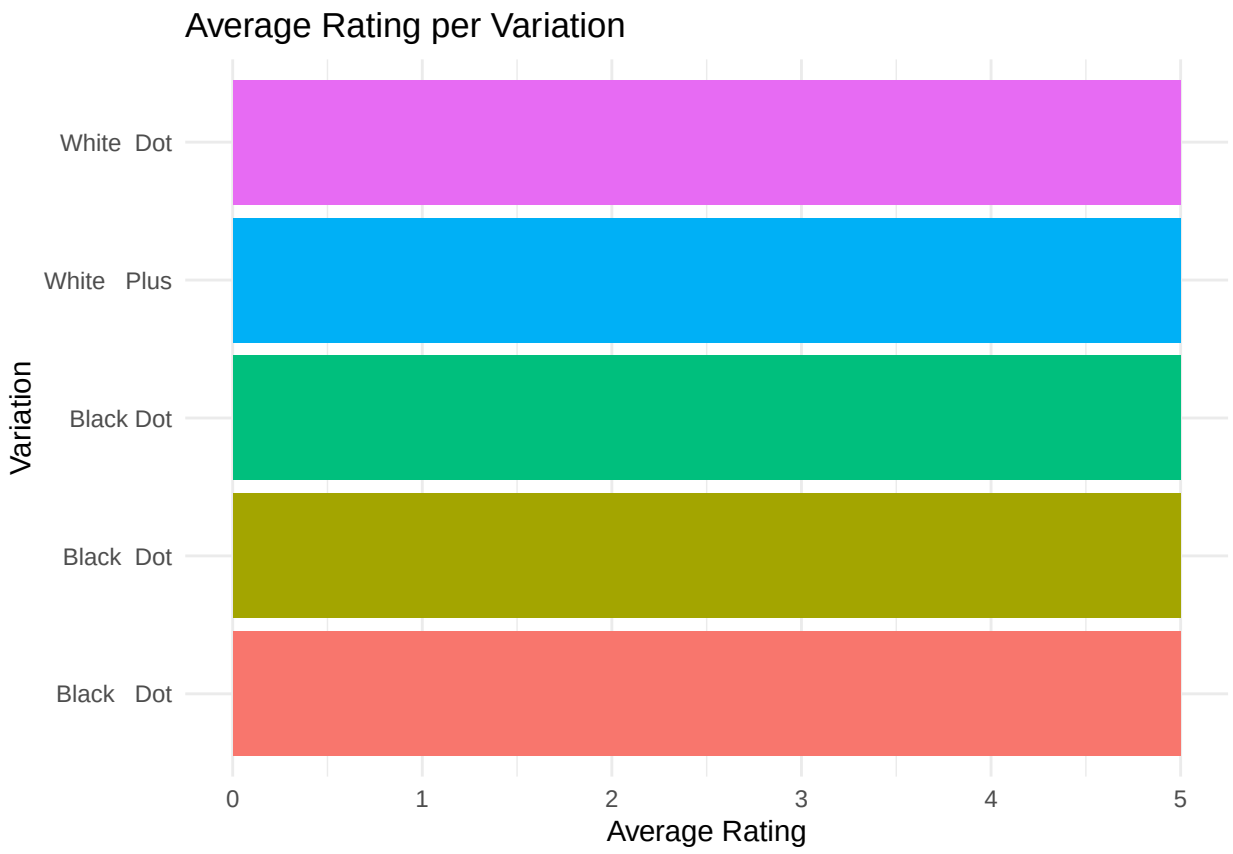
```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `verified_reviews_num = as.numeric(verified_reviews)`.
## Caused by warning:
## ! NAs introduced by coercion
```

```
# Reviews over time
reviews_time <- alexa_data %>%
  group_by(review_date) %>%
  summarise(total_reviews = sum(verified_reviews_num, na.rm = TRUE))

ggplot(reviews_time, aes(x = review_date, y = total_reviews)) +
  geom_line(color = "steelblue") +
  geom_point(color = "red") +
  labs(title = "Verified Reviews Over Time",
       x = "Date",
       y = "Number of Verified Reviews") +
  theme_minimal()
```

```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```




```r
# Variation vs Ratings
variation_ratings <- alexa_data %>%
  group_by(variation) %>%
  summarise(mean_rating = mean(avg_rating, na.rm = TRUE),
            count = n()) %>%
  arrange(desc(mean_rating))

ggplot(variation_ratings, aes(x = reorder(variation, mean_rating), y = mean_rating, fill = variation)) +
  geom_col() +
  coord_flip() +
  labs(title = "Average Rating per Variation",
       x = "Variation",
       y = "Average Rating") +
  theme_minimal() +
  guides(fill = FALSE)
```


Average Rating per Variation
White Dot
White Plus
Black Dot
Black Dot
Black Dot
Variation
0
1
2
3
4
5
Average Rating