

# Untitled

Lysyl Bess H. Talaban

2025-12-14

```
#1

library(ggplot2)
write.csv(mpg, "mpg.csv", row.names = FALSE)
mpg_data <- read.csv("mpg.csv", header = TRUE)
str(mpg_data)

## '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" ...

categorical_vars <- names(mpg_data)[sapply(mpg_data, function(x) is.character(x) | is.factor(x))]
categorical_vars

## [1] "manufacturer" "model"          "trans"          "drv"           "fl"

## [6] "class"

continuous_vars <- names(mpg_data)[sapply(mpg_data, is.numeric)]
continuous_vars

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

#2a

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
```

```

data(mpg)

manufacturer_counts <- mpg %>%
  group_by(manufacturer) %>%
  summarise(unique_models = n_distinct(model)) %>%
  arrange(desc(unique_models))

print("Unique models per manufacturer:")

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

```

print(manufacturer_counts)

## # A tibble: 15 x 2
##   manufacturer unique_models
##   <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
```

```

top_manufacturer <- manufacturer_counts$manufacturer[1]
print(paste("Manufacturer with the most models:", top_manufacturer))
```

```

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

```

model_counts <- mpg %>%
  group_by(model) %>%
  summarise(variations = n()) %>%
  arrange(desc(variations))

print("Model variations:")

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

```

print(model_counts)

## # A tibble: 38 x 2
##   model           variations
##   <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

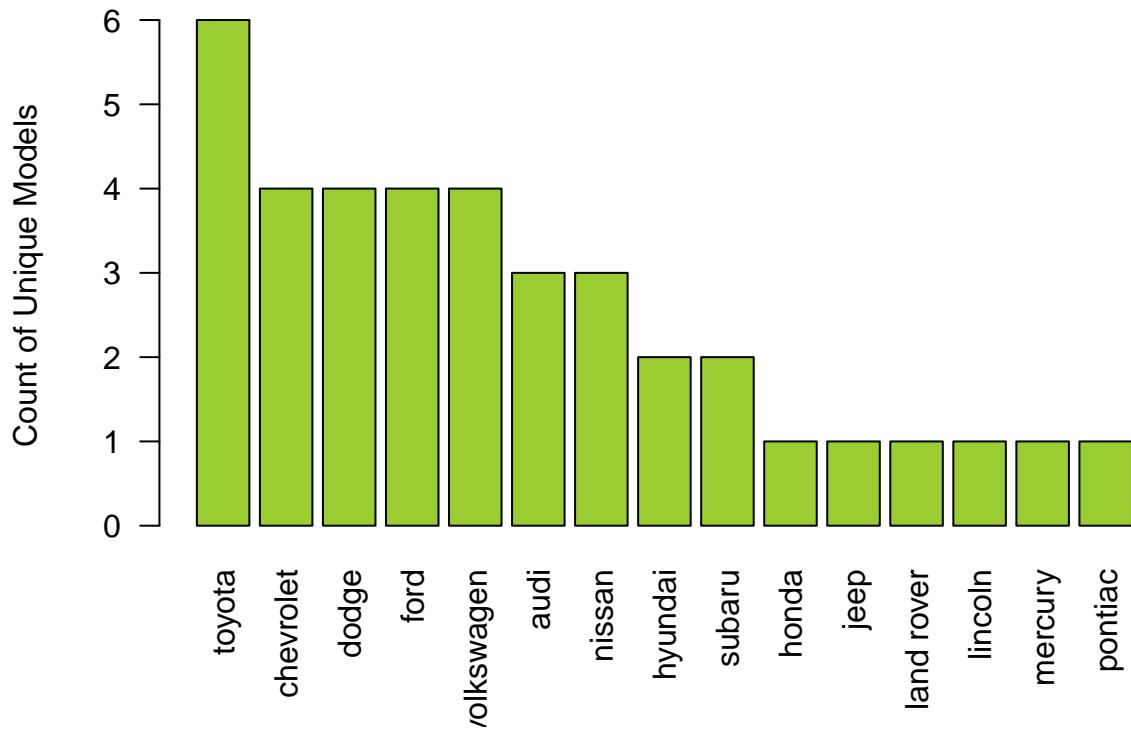
top_model <- model_counts$model[1]
print(paste("Model with the most variations:", top_model))

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

# Corrected base R plot
barplot(height = manufacturer_counts$unique_models,
        names.arg = manufacturer_counts$manufacturer,
        col = "yellowgreen",
        las = 2,
        main = "Unique Models per Manufacturer (Base R)",
        ylab = "Count of Unique Models")

```

**Unique Models per Manufacturer (Base R)**

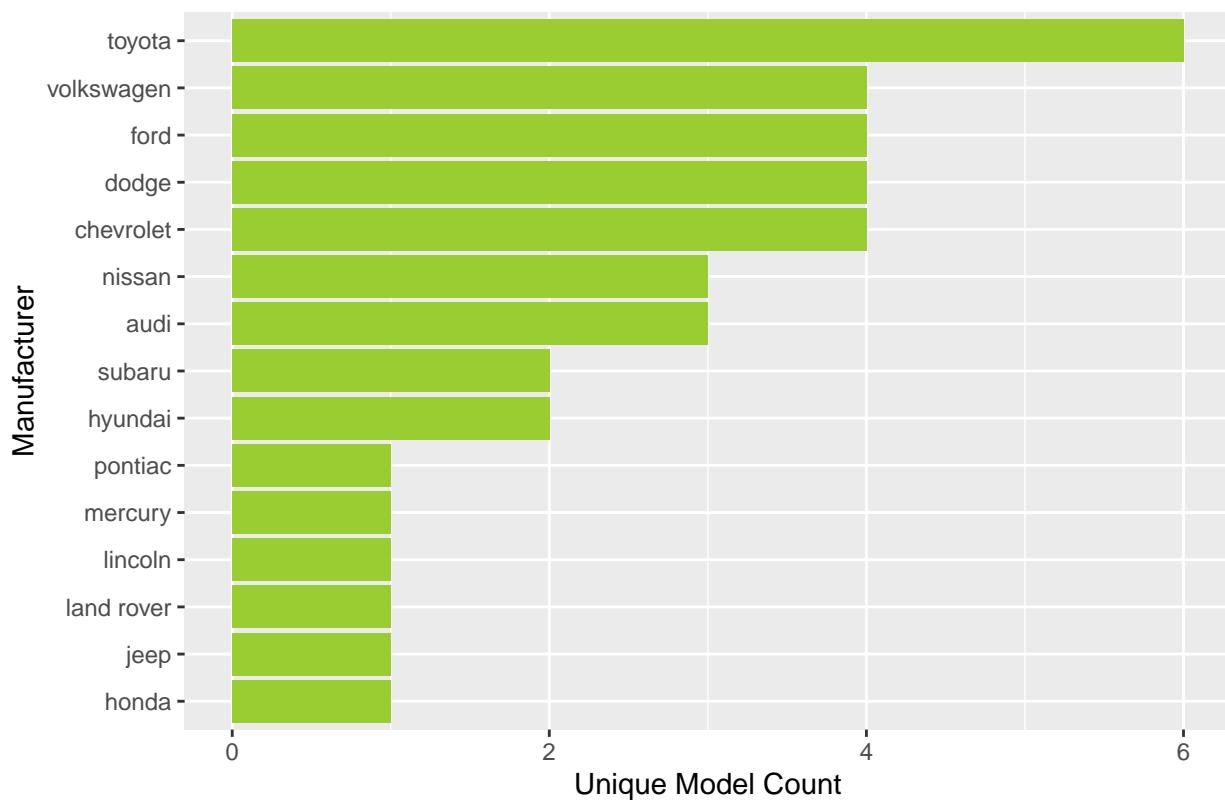


```

# ggplot2 version
ggplot(manufacturer_counts, aes(x = reorder(manufacturer, unique_models),
                                  y = unique_models)) +
  geom_col(fill = "yellowgreen") +
  coord_flip() +
  labs(title = "Unique Models per Manufacturer (ggplot2)",
       x = "Manufacturer",
       y = "Unique Model Count")

```

### Unique Models per Manufacturer (ggplot2)



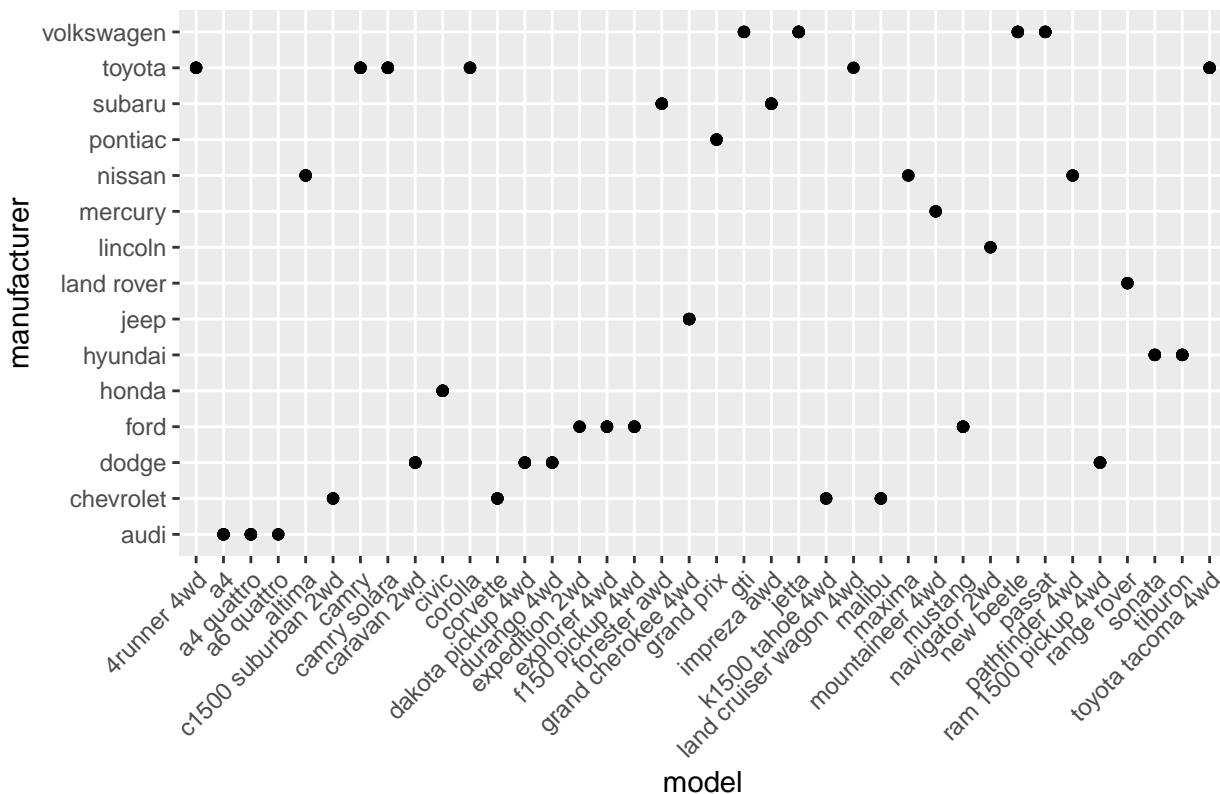
#2b

```
library(ggplot2)
library(dplyr)

data(mpg)

ggplot(mpg, aes(model, manufacturer)) +
  geom_point() +
  labs(title = "Original Model vs Manufacturer Plot (raw points)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

## Original Model vs Manufacturer Plot (raw points)



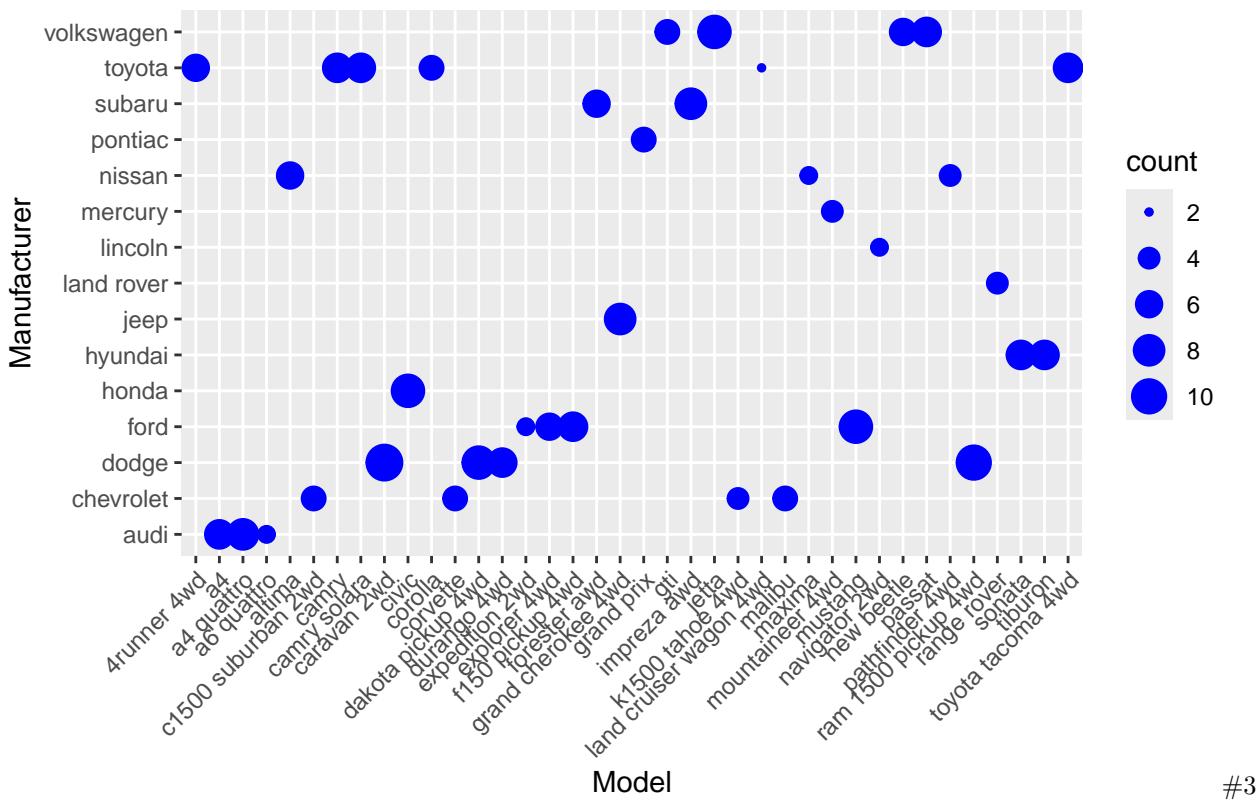
```
model_summary <- mpg %>%
```

```
  group_by(manufacturer, model) %>%
  summarise(count = n()) %>%
  ungroup()
```

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

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

## Model vs Manufacturer (point size = number of variations)



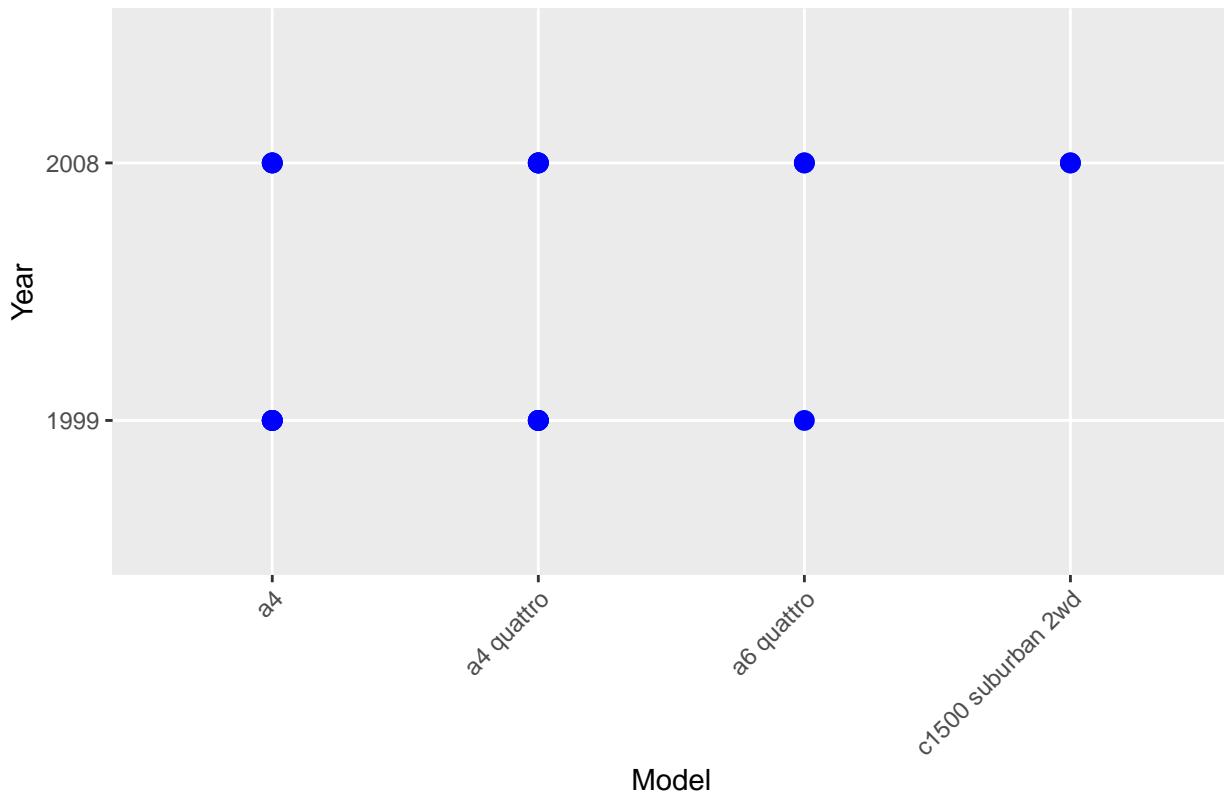
```
library(ggplot2)
library(dplyr)

data(mpg)

top20 <- mpg %>% slice(1:20)

ggplot(top20, aes(x = model, y = factor(year))) +
  geom_point(color = "blue", size = 3) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Top 20 Observations: Model vs Year",
       x = "Model",
       y = "Year")
```

## Top 20 Observations: Model vs Year



```
#4
library(ggplot2)
library(dplyr)

data(mpg)

model_count <- mpg %>%
  group_by(model) %>%
  summarise(num_cars = n()) %>%
  arrange(desc(num_cars))

print(model_count)

## # A tibble: 38 x 2
##   model           num_cars
##   <chr>          <int>
## 1 caravan        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
```

```

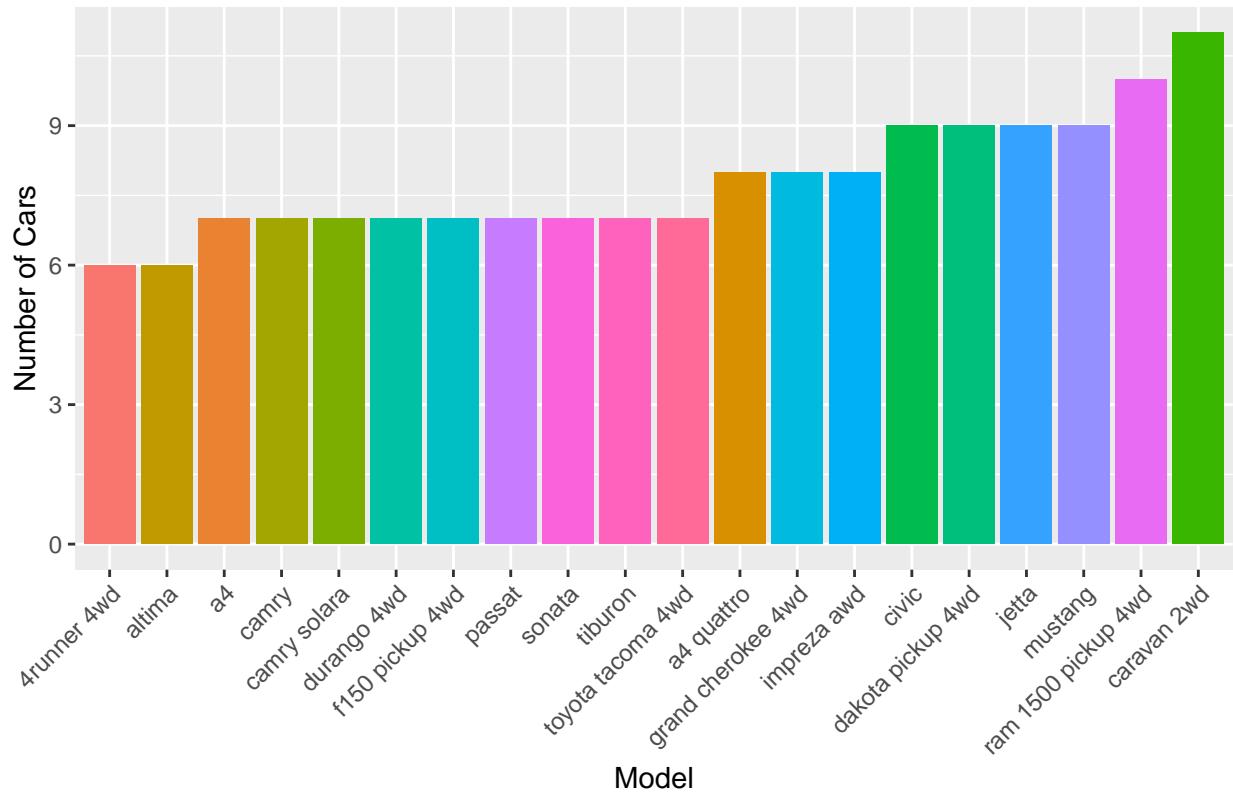
top20_models <- model_count %>% slice(1:20)

ggplot(top20_models, aes(x = reorder(model, num_cars), y = num_cars, fill = model)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Top 20 Models by Number of Cars",
       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.

```

Top 20 Models by Number of Cars

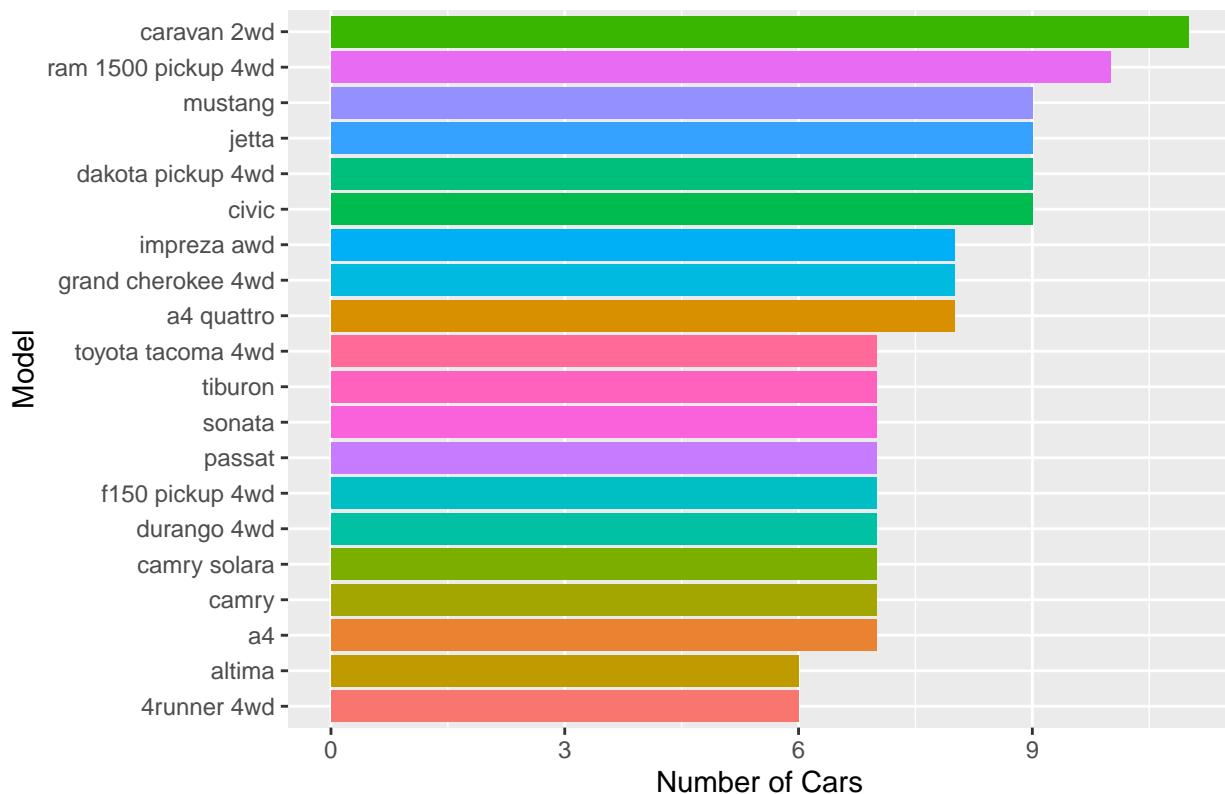


```

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

```

## Top 20 Models by Number of Cars (Horizontal)



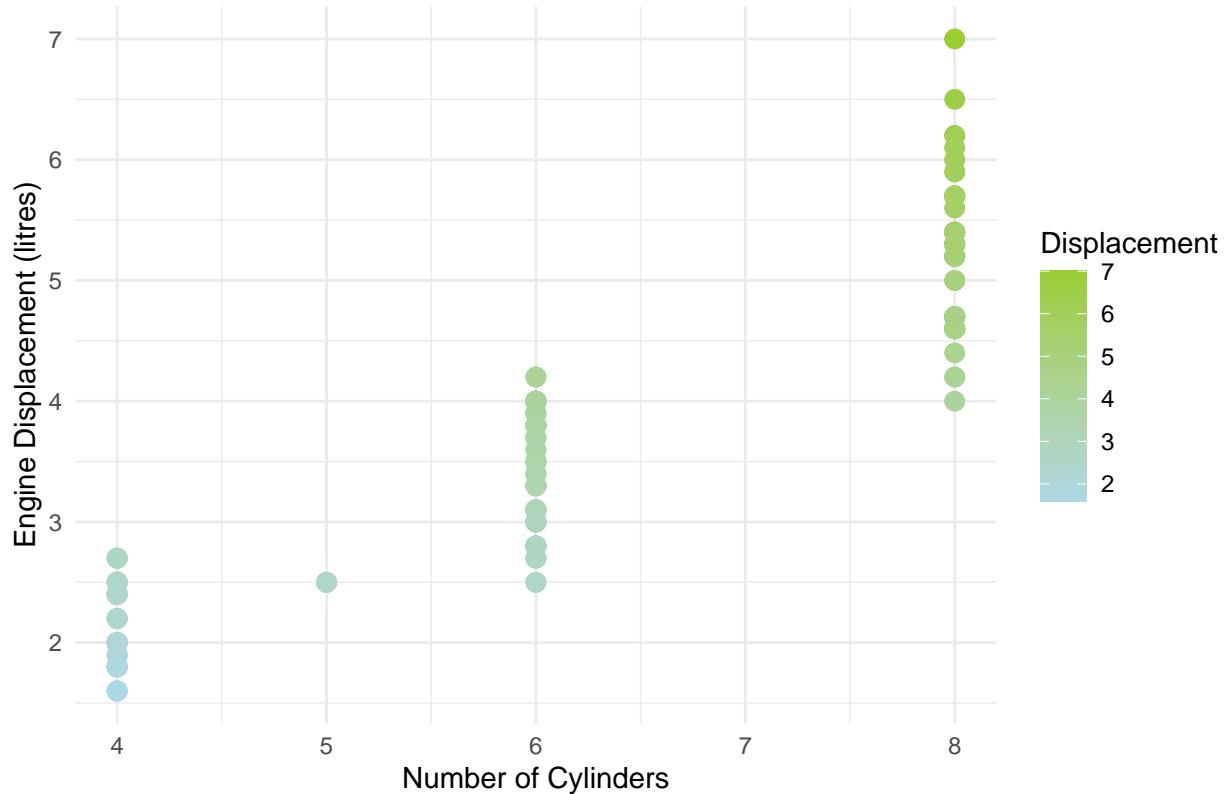
```
#5
```

```
library(ggplot2)
library(dplyr)

data(mpg)

ggplot(mpg, aes(x = cyl, y = displ, color = displ)) +
  geom_point(size = 3) +
  scale_color_gradient(low = "lightblue", high = "yellowgreen") +
  labs(title = "Relationship between No. of Cylinders and Engine Displacement",
       x = "Number of Cylinders",
       y = "Engine Displacement (litres)",
       color = "Displacement") +
  theme_minimal()
```

## Relationship between No. of Cylinders and Engine Displacement

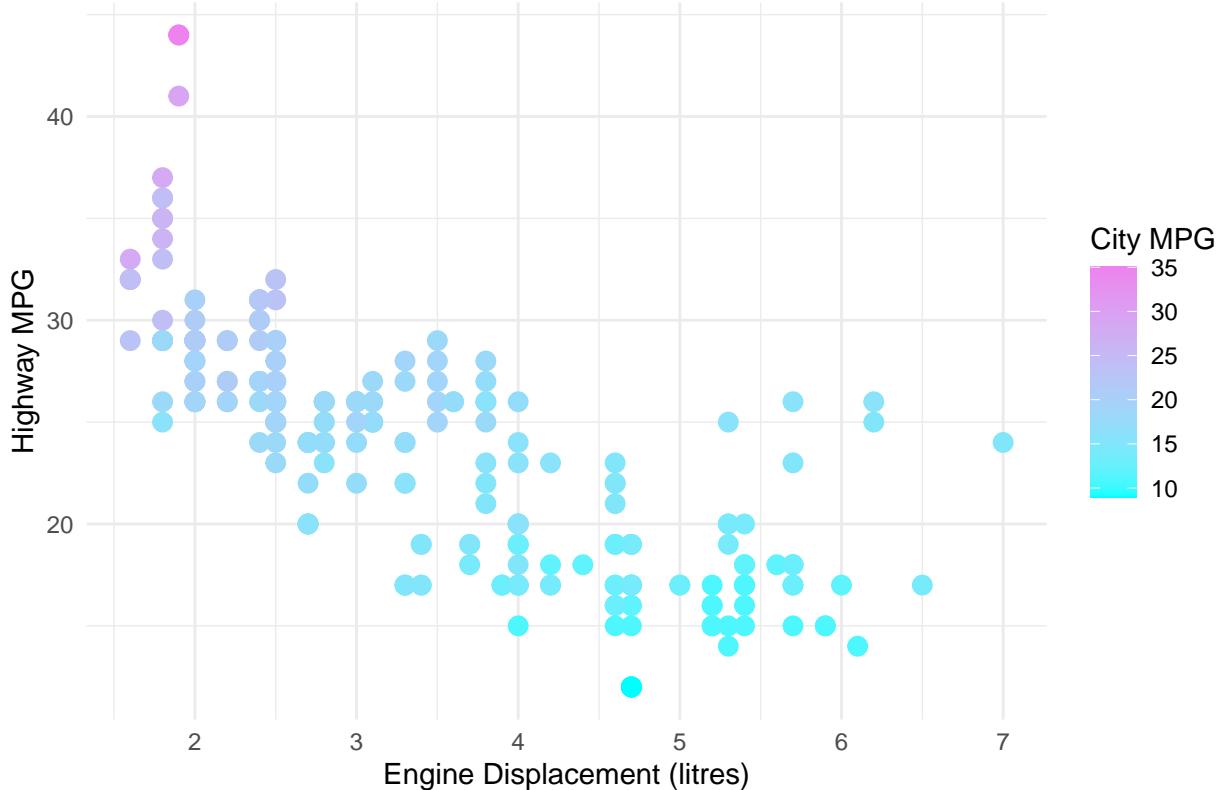


```
#6
library(ggplot2)
library(dplyr)

data(mpg)

ggplot(mpg, aes(x = displ, y = hwy, color = cty)) +
  geom_point(size = 3) +
  scale_color_gradient(low = "cyan", high = "violet") +
  labs(title = "Engine Displacement vs Highway MPG colored by City MPG",
       x = "Engine Displacement (litres)",
       y = "Highway MPG",
       color = "City MPG") +
  theme_minimal()
```

## Engine Displacement vs Highway MPG colored by City MPG



```

traffic <- data.frame(
  Date = as.Date('2025-11-01') + 0:9,
  Location = rep(c("Intersection A", "Intersection B"), each = 5),
  Vehicles = c(120, 150, 130, 160, 140, 200, 210, 190, 205, 220),
  Average_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, "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"           "Location"        "Vehicles"        "Average_Speed"
junction_A <- traffic_data %>% filter(Location == "Intersection A")
junction_B <- traffic_data %>% filter(Location == "Intersection B")

print("Junction A data:")

## [1] "Junction A data:"

```

```

print(junction_A)

##           Date      Location Vehicles Average_Speed
## 1 2025-11-01 Intersection A     120      35.5
## 2 2025-11-02 Intersection A     150      34.2
## 3 2025-11-03 Intersection A     130      36.0
## 4 2025-11-04 Intersection A     160      33.8
## 5 2025-11-05 Intersection A     140      34.5

print("Junction B data:")

## [1] "Junction B data:"

print(junction_B)

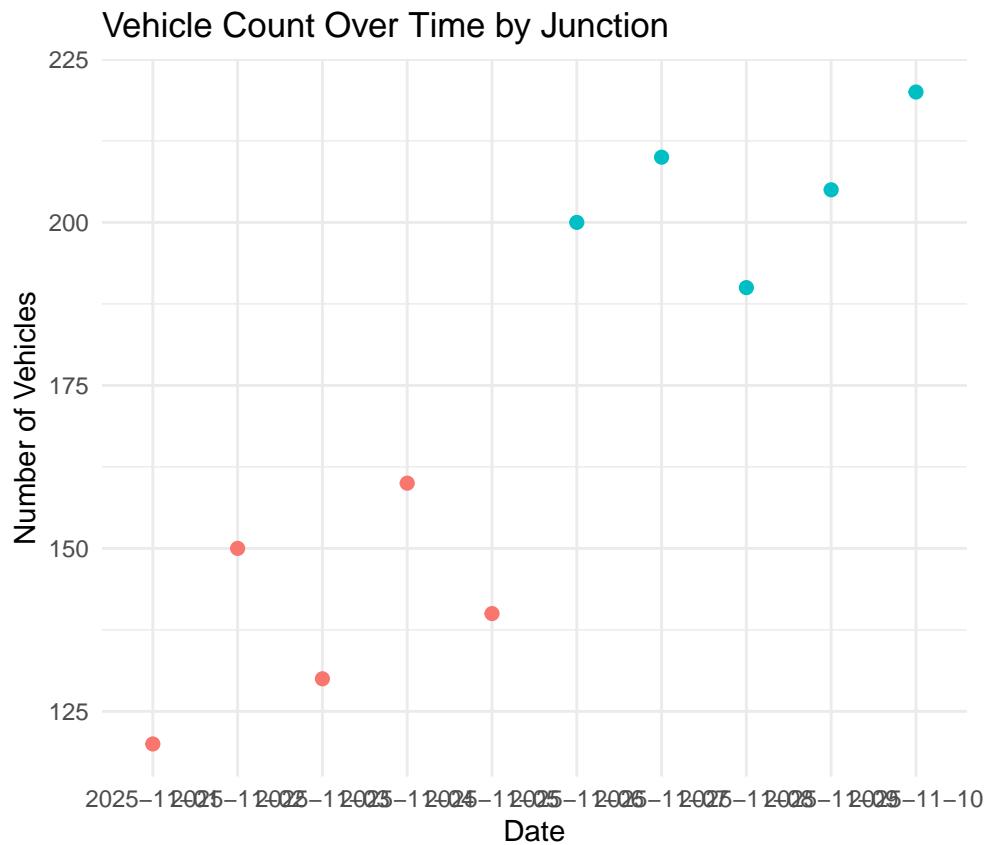
##           Date      Location Vehicles Average_Speed
## 1 2025-11-06 Intersection B     200      32.0
## 2 2025-11-07 Intersection B     210      31.5
## 3 2025-11-08 Intersection B     190      33.0
## 4 2025-11-09 Intersection B     205      30.8
## 5 2025-11-10 Intersection B     220      29.5

ggplot(traffic_data, aes(x = Date, y = Vehicles, color = Location)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  labs(title = "Vehicle Count Over Time by Junction",
       x = "Date",
       y = "Number of Vehicles",
       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
```

```
library(dplyr)
library(readxl)
library(ggplot2)

alexa <- read_excel("alexa_file.xlsx")

alexa <- alexa %>%
  mutate(
    verified_reviews = as.numeric(gsub("[^0-9.]", "", verified_reviews)),
    rating = as.numeric(gsub("[^0-9.]", "", rating)),
    date = as.Date(date)
  )

## Warning: There was 1 warning in `mutate()` .
## i In argument: `verified_reviews = as.numeric(gsub("[^0-9.]", "", verified_reviews))` .
## Caused by warning:
## ! NAs introduced by coercion

reviews_over_time <- alexa %>%
  group_by(date) %>%
  summarise(total_reviews = sum(verified_reviews, na.rm = TRUE))

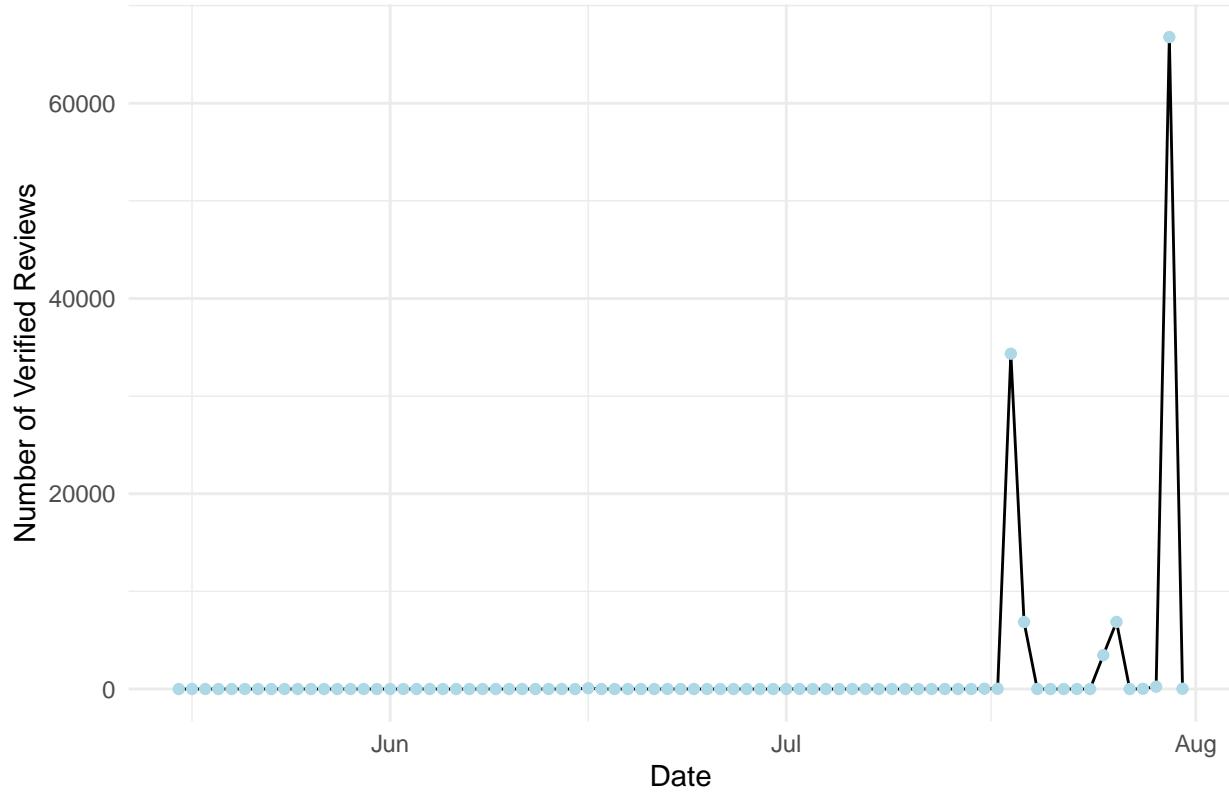
ggplot(reviews_over_time, aes(x = date, y = total_reviews)) +
  geom_line(color = "black") +
  geom_point(color = "lightblue") +
  labs(title = "Verified Reviews Over Time",
```

```

x = "Date",
y = "Number of Verified Reviews") +
theme_minimal()

```

Verified Reviews Over Time



```

variation_rating <- alexa %>%
  group_by(variation) %>%
  summarise(avg_rating = mean(rating, na.rm = TRUE),
            count = n()) %>%
  arrange(desc(avg_rating))

ggplot(variation_rating, aes(x = reorder(variation, avg_rating), y = avg_rating, fill = variation)) +
  geom_col() +
  coord_flip() +
  labs(title = "Average Rating by Variation",
       x = "Variation",
       y = "Average Rating") +
  theme_minimal() +
  guides(fill = FALSE)

```

