WESTERN SYDNEY
UNIVERSITY

SCHOOL OF COMPUTING, ENGINEERING AND MATHEMATICS

ASSIGNMENT COVER SHEET

STUDENT DETAILS

Student name: Hoàng Ngọc Thuỷ Tiên Student ID number: 22167438

UNIT AND TUTORIAL DETAILS

Unit name: Analytics Programming Unit number: COMP1013

Tutorial group: _____ Tutorial day and time: Sunday, 15:30-18:45

Lecturer or Tutor name: Assoc. Prof. NGUYEN Tan Luy

ASSIGNMENT DETAILS

Title: Assignment T1 2025 Submission

Length: 2000 words - 15 pages Due date: 04/04/2025 Date submitted: 04/04/2025

Home campus (where you are enrolled): Vietnam

DECLARATION

Before submitting the assignment, include the following declaration in a clearly visible and readable place on the cover page of your project report.

***

By including this statement, we the authors of this work, verify that:

- We hold a copy of this assignment that we can produce if the original is lost or damaged.
- We hereby certify that no part of this assignment/product has been copied from any other student's work or from any other source except where due acknowledgement is made in the assignment.
- No part of this assignment/product has been written/produced for us by another person except where such collaboration has been authorised by the subject lecturer/tutor concerned.
- We are aware that this work may be reproduced and submitted to plagiarism detection software programs for the purpose of detecting possible plagiarism (**which may retain a copy on its database for future plagiarism checking**).
- We hereby certify that we have read and understand what the School of Computing, Engineering and Mathematics defines as minor and substantial breaches of misconduct as outlined in the learning guide for this unit.

***

Note: An examiner or lecturer/tutor has the right not to mark this project report if the above declaration has not been added to the cover of the report.

Student's signature: Hoang Ngoc Thuy Tien

Project Trimester 1 2025 - Individual Report - COMP1013

Hoàng Ngọc Thuỷ Tiên – 22167438

2025-04-04

Question 1

```
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.4.2

library(dplyr)#for the purpose of data visualization of horsepower
distribution

## Warning: package 'dplyr' was built under R version 4.4.3

##
## 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

# Load the datasets and use stringsAsFactors in this step to change the type
of strings to factors
engine_data <- read.csv("Engine.csv", stringsAsFactors = FALSE)
automobile_data <- read.csv("Automobile.csv", stringsAsFactors = FALSE)
maintenance_data <- read.csv("Maintenance.csv", stringsAsFactors = FALSE)

# Replace "?" with NA
engine_data[engine_data == "?"] <- NA
automobile_data[automobile_data == "?"] <- NA
maintenance_data[maintenance_data == "?"] <- NA

# Inspect the data structure
str(engine_data)

## 'data.frame':    88 obs. of  8 variables:
##  $ EngineModel : chr  "E-0001" "E-0002" "E-0003" "E-0004" ...
##  $ EngineType  : chr  "dohc" "ohcv" "ohc" "ohc" ...
##  $ NumCylinders: chr  "four" "six" "four" "five" ...
##  $ EngineSize  : int  130 152 109 136 136 131 131 108 164 164 ...
##  $ FuelSystem  : chr  "mpfi" "mpfi" "mpfi" "mpfi" ...
##  $ Horsepower  : chr  "111" "154" "102" "115" ...
```

```
##  $ FuelTypes   : chr  "gas" "gas" "gas" "gas" ...
##  $ Aspiration  : chr  "std" "std" "std" "std" ...
```

```
summary(engine_data) #present the data
```

```
##  EngineModel         EngineType        NumCylinders         EngineSize
##  Length:88          Length:88          Length:88          Min.   : 60.0
##  Class :character   Class :character   Class :character   1st Qu.:108.0
##  Mode  :character   Mode  :character   Mode  :character   Median :121.0
##                                                           Mean   :134.1
##                                                           3rd Qu.:151.2
##                                                           Max.   :320.0
##   FuelSystem         Horsepower         FuelTypes          Aspiration
##  Length:88          Length:88          Length:88          Length:88
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
```

```
str(automobile_data)
```

```
## 'data.frame':    204 obs. of  13 variables:
##  $ PlateNumber   : chr  "53N-001" "53N-002" "53N-003" "53N-004" ...
##  $ Manufactures  : chr  "Alfa-romero" "Alfa-romero" "Audi" "Audi" ...
##  $ BodyStyles    : chr  "convertible" "hatchback" "sedan" "sedan" ...
##  $ DriveWheels   : chr  "rwd" "rwd" "fwd" "4wd" ...
##  $ EngineLocation: chr  "front" "front" "front" "front" ...
##  $ WheelBase     : num  88.6 94.5 99.8 99.4 99.8 ...
##  $ Length        : num  169 171 177 177 177 ...
##  $ Width         : num  64.1 65.5 66.2 66.4 66.3 71.4 71.4 71.4 67.9 64.8
...
##  $ Height        : num  48.8 52.4 54.3 54.3 53.1 55.7 55.7 55.9 52 54.3
...
##  $ CurbWeight    : int  2548 2823 2337 2824 2507 2844 2954 3086 3053 2395
...
##  $ EngineModel   : chr  "E-0001" "E-0002" "E-0003" "E-0004" ...
##  $ CityMpg       : int  21 19 24 18 19 19 19 17 16 23 ...
##  $ HighwayMpg    : int  27 26 30 22 25 25 25 20 22 29 ...
```

```
summary(automobile_data)  #present the data
```

```
##  PlateNumber        Manufactures        BodyStyles        DriveWheels
##  Length:204         Length:204         Length:204         Length:204
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  EngineLocation       WheelBase          Length          Width
##  Length:204         Min.   : 86.60   Min.   :141.1   Min.   :60.30
```

```
##  Class :character   1st Qu.: 94.50   1st Qu.:166.3   1st Qu.:64.08
##  Mode  :character   Median : 97.00   Median :173.2   Median :65.50
##                     Mean   : 98.81   Mean   :174.1   Mean   :65.92
##                     3rd Qu.:102.40   3rd Qu.:183.2   3rd Qu.:66.90
##                     Max.   :120.90   Max.   :208.1   Max.   :72.30
##      Height        CurbWeight   EngineModel           CityMpg
##  Min.   :47.80   Min.   :1488   Length:204         Min.   :10.00
##  1st Qu.:52.00   1st Qu.:2145   Class :character   1st Qu.:19.00
##  Median :54.10   Median :2414   Mode  :character   Median :24.00
##  Mean   :53.75   Mean   :2556                      Mean   :25.23
##  3rd Qu.:55.50   3rd Qu.:2939                      3rd Qu.:30.00
##  Max.   :59.80   Max.   :4066                      Max.   :50.00
##    HighwayMpg
##  Min.   :15.00
##  1st Qu.:25.00
##  Median :30.00
##  Mean   :30.76
##  3rd Qu.:34.50
##  Max.   :55.00
```

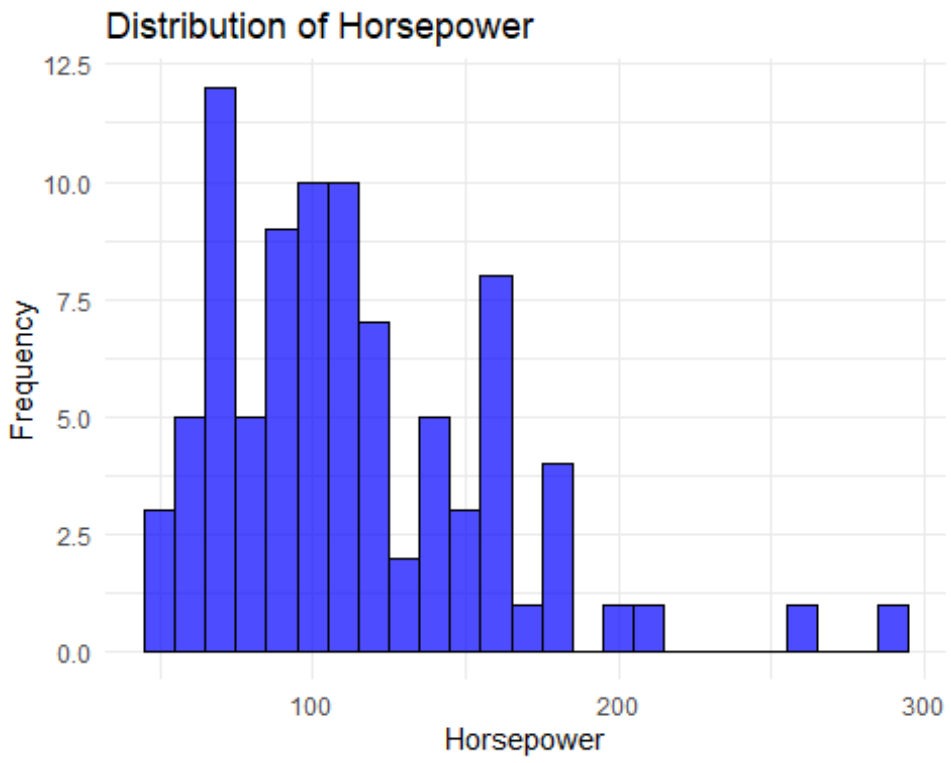
```
str(maintenance_data)
```

```
## 'data.frame':    374 obs. of  7 variables:
##  $ ID         : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ PlateNumber: chr  "53N-001" "53N-001" "53N-001" "53N-001" ...
##  $ Date       : chr  "15/02/2024" "16/03/2024" "15/04/2024" "15/05/2024"
...
##  $ Troubles   : chr  "Break system" "Transmission" "Suspected clutch"
"Ignition (finding)" ...
##  $ ErrorCodes : int  -1 -1 -1 1 -1 1 1 0 -1 -1 ...
##  $ Price      : int  110 175 175 180 85 1000 180 0 180 180 ...
##  $ Methods    : chr  "Replacement" "Replacement" "Adjustment" "Adjustment"
...
```

```
summary(maintenance_data)  #present the data
```

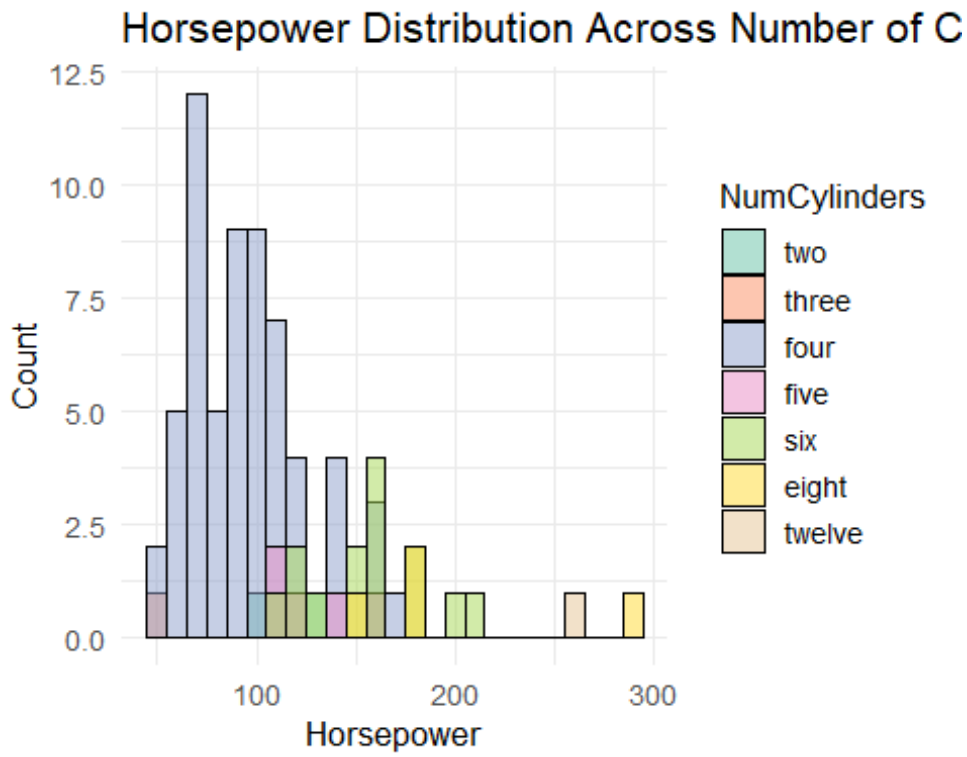
```
##        ID         PlateNumber           Date             Troubles
##  Min.   :  1.00   Length:374         Length:374         Length:374
##  1st Qu.: 94.25   Class :character   Class :character   Class :character
##  Median :187.50   Mode  :character   Mode  :character   Mode  :character
##  Mean   :187.50
##  3rd Qu.:280.75
##  Max.   :374.00
##    ErrorCodes           Price          Methods
##  Min.   :-1.00000   Min.   :   0.0   Length:374
##  1st Qu.:-1.00000   1st Qu.:  85.0   Class :character
##  Median : 0.00000   Median : 120.0   Mode  :character
##  Mean   : 0.04813   Mean   : 204.8
##  3rd Qu.: 1.00000   3rd Qu.: 180.0
##  Max.   : 1.00000   Max.   :1000.0
```

```r
# Convert categorical variables BodyStyles, FuelTypes, ErrorCodes (columns)
to factors
# Convert BodyStyles
automobile_data$BodyStyles <- factor(automobile_data$BodyStyles,
                                     levels = c("hardtop", "wagon", "sedan",
"hatchback", "convertible"))

# Convert FuelTypes
engine_data$FuelType <- factor(engine_data$FuelType,
                               levels = c("diesel", "gas"))

# Convert ErrorCodes
maintenance_data$ErrorCodes <- factor(maintenance_data$ErrorCodes,
                                      levels = c(0, 1, -1),
                                      labels = c("No Error", "Engine
Failure", "Other Component Failure"))

# Replace missing values NA in Horsepower column with the mean Horsepower
engine_data$Horsepower <- as.numeric(as.character(engine_data$Horsepower))  #
convert Horsepower column to numeric
mean_horsepower <- mean(engine_data$Horsepower, na.rm = TRUE)  # Calculate
the mean of the Horsepower column
engine_data$Horsepower[is.na(engine_data$Horsepower)] <- mean_horsepower  #
Replace NA values with mean values

# Plot histogram of Horsepower distribution
ggplot(engine_data, aes(x = Horsepower)) +
  geom_histogram(binwidth = 10, fill = "blue", color = "black", alpha = 0.7)
+
  labs(title = "Distribution of Horsepower",
       x = "Horsepower",
       y = "Frequency") +
  theme_minimal()
```



Question 2

```r
engine_data$NumCylinders <- factor(engine_data$NumCylinders,
                                   levels = c("two", "three", "four", "five",
"six", "eight", "twelve"),
                                   ordered = TRUE) # a categorical variable
with increasing order

# Analyze Horsepower distribution across NumCylinders
ggplot(engine_data, aes(x = Horsepower, fill = NumCylinders)) +
  geom_histogram(binwidth = 10, position = "identity", alpha = 0.5, color =
"black") +
  labs(title = "Horsepower Distribution Across Number of Cylinders",
       x = "Horsepower",
       y = "Count") +
  theme_minimal(base_size = 13) +
  scale_fill_brewer(palette = "Set2")
```

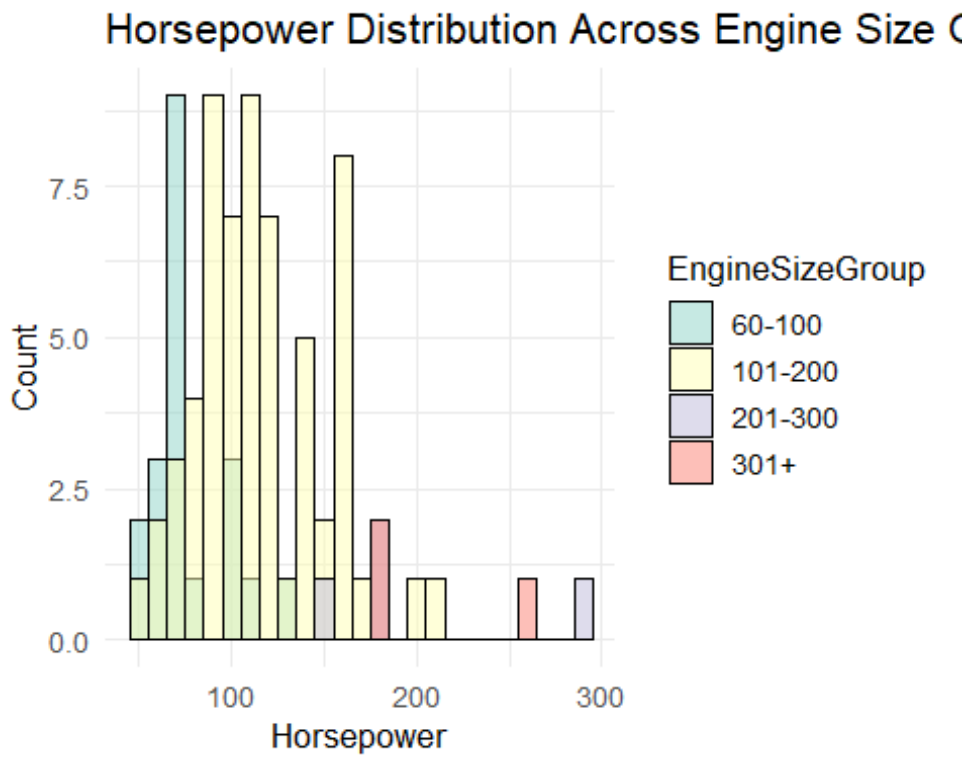


#Explain the findings

+ **Findings from the Horsepower Distribution Across Number of Cylinders** A direct relationship exists between the number of cylinders and horsepower: **larger cylinders produce more horsepower than smaller cylinders**. Most cars on the market have **four cylinders**, with horsepower values ranging from 50-150. However, **horsepower values become more dispersed for cars with 5 to 12 cylinders**, reaching higher levels around 150-200 and even close to 300. However, there are very few cars with high-cylinder engines. One can see a strange point in the chart: some cars have 6-cylinder engines but have relatively lower horsepower than 5-cylinders. It can be concluded that most cars on the market belong to the small or medium-engine segment. The horsepower groups from 201 to 300 have low frequencies, indicating that the number of cars with high power is still small. However, there are some exceptions where the car has high cylinders but low horsepower.

```
# Categorize EngineSize into groups because EngineSize is continuous
numerical variable
engine_data$EngineSizeGroup <- cut(engine_data$EngineSize,
```

```
                                    breaks = c(60, 100, 200, 300, Inf),
                                    labels = c("60-100", "101-200", "201-300",
"301+"),
                                    right = TRUE,
                                    include.lowest = TRUE)

engine_data$EngineSizeGroup <- factor(engine_data$EngineSizeGroup,
                                      levels = c("60-100", "101-200", "201-
300", "301+"),
                                      ordered = TRUE) # helps the chart
display in the correct order from small to large

# Analyze Horsepower distribution across EngineSize groups
ggplot(engine_data, aes(x = Horsepower, fill = EngineSizeGroup)) +
  geom_histogram(binwidth = 10, position = "identity", alpha = 0.5, color =
"black") +
  labs(title = "Horsepower Distribution Across Engine Size Groups",
       x = "Horsepower",
       y = "Count") +
  theme_minimal(base_size = 13) +
  scale_fill_brewer(palette = "Set3")
```



Explain the findings

+ **Findings from the Horsepower Distribution Across Engine Size Groups** The second chart classifies vehicles by engine power group. Most of the engine sizes of most vehicles on the market will be in the 60-100 and 101-200 groups with low to medium power. These are vehicles with moderate horsepower performance. Vehicles with larger engines will have more horsepower. -> When the engine has more horsepower, it will accelerate faster. Conversely, engines with lower horsepower will be more fuel efficient but accelerate slower.

Conclusion

Horsepower, engine size and cylinders directly reflect the vehicle's performance. **The more cylinders and engine size, the higher the horsepower**. In the mass market, *4-cylinder engines dominate* with an *average power range of 101-200 HP*. Meanwhile, vehicles with a high number of cylinders, from 6 to 12, and power of over 200 HP are less common.

Question 3

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 4.4.3
```

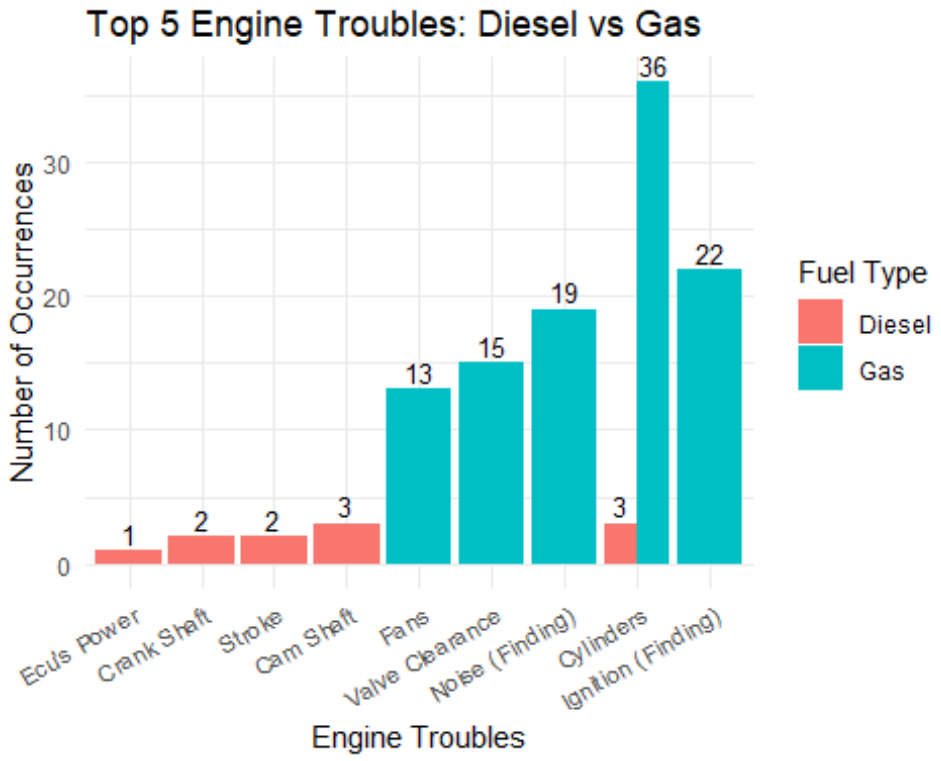
```
#Combine maintenance_data with automobile_data to get Engine Model
maint_auto <- maintenance_data %>%
  left_join(automobile_data[, c("PlateNumber", "EngineModel")], by =
"PlateNumber")

#Attach FuelTypes to the data, based on the EngineModel
maint_full <- maint_auto %>%
  left_join(engine_data[, c("EngineModel", "FuelTypes")], by = "EngineModel")
```

```
## Warning in left_join(., engine_data[, c("EngineModel", "FuelTypes")], by =
"EngineModel"): Detected an unexpected many-to-many relationship between `x`
and `y`.
## ℹ Row 36 of `x` matches multiple rows in `y`.
## ℹ Row 1 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
#Filter for engine-related troubles
engine_troubles <- maint_full %>%
```

```r
  filter(
    ErrorCodes == 1 |
    str_detect(tolower(Troubles), "cam shaft|crank shaft|cylinders|ecu's
power|fans|ignition|ignition \\(finding\\)|noise \\(finding\\)|o2 sensors|oil
filter|pedals|pressure sensors|stroke|suspected battery|temperature
sensors|valve clearance") #detect keywords that are suspicious of engine
failure in the error description
  ) %>%
  filter(Troubles != "No error") %>%
  mutate(Troubles = str_trim(str_to_title(Troubles)))

#Top 5 most common troubles for Diesel
top5_diesel_only <- engine_troubles %>%
  filter(FuelTypes == "diesel") %>%
  group_by(Troubles) %>%
  summarise(Count = n()) %>%
  arrange(desc(Count)) %>%
  slice_head(n = 5)

print("Top 5 Engine Troubles for Diesel:")

## [1] "Top 5 Engine Troubles for Diesel:"

print(top5_diesel_only)

## # A tibble: 5 × 2
##   Troubles    Count
##   <chr>       <int>
## 1 Cam Shaft       3
## 2 Cylinders       3
## 3 Crank Shaft     2
## 4 Stroke          2
## 5 Ecu's Power     1

#Top 5 most common troubles for Gas
top5_gas_only <- engine_troubles %>%
  filter(FuelTypes == "gas") %>%
  group_by(Troubles) %>%
  summarise(Count = n()) %>%
  arrange(desc(Count)) %>%
  slice_head(n = 5)

print("Top 5 Engine Troubles for Gas:")

## [1] "Top 5 Engine Troubles for Gas:"

print(top5_gas_only)

## # A tibble: 5 × 2
##   Troubles           Count
##   <chr>              <int>
```

```
## 1 Cylinders             36
## 2 Ignition (Finding)    22
## 3 Noise (Finding)       19
## 4 Valve Clearance       15
## 5 Fans                  13

#Attach fuel labels before combining and visualizing the data
top5_diesel_only <- top5_diesel_only %>%
  mutate(FuelType = "Diesel")
top5_gas_only <- top5_gas_only %>%
  mutate(FuelType = "Gas")
#Combine for plotting
top5_combined <- bind_rows(top5_diesel_only, top5_gas_only) # Combine data of
2 vehicle groups into 1 table

#Plot for comparision
ggplot(top5_combined, aes(x = reorder(Troubles, Count), y = Count, fill =
FuelType)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.9)) +
  geom_text(aes(label = Count), position = position_dodge(width = 0.9), vjust
= -0.3, size = 3.5) +
  labs(
    title = "Top 5 Engine Troubles: Diesel vs Gas",
    x = "Engine Troubles",
    y = "Number of Occurrences",
    fill = "Fuel Type"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))
```



Elaborate on findings

Based on the chart, it can be seen that **Gasoline-powered vehicles have more frequent failures than diesel-powered vehicles**. In particular, Cylinders, Ignition and Noise are the most frequently occurrence parts in Gas, with Cylinders appearing 36 times in Gas vehicles and only 3 cases in Diesel.

Diesel vehicles have fewer errors, often occurring sporadically in the Cam Shaft, Crank Shaft, and Stroke parts. However, errors in the Cam Shaft or Crank Shaft are serious errors related to power transmission, as well as the parts that ensure the engine's operating cycle takes place normally. This shows that **although diesel vehicles rarely make errors, they will be serious errors if they do occur**. Meanwhile, *Gas vehicles tend to have minor faults frequently*.

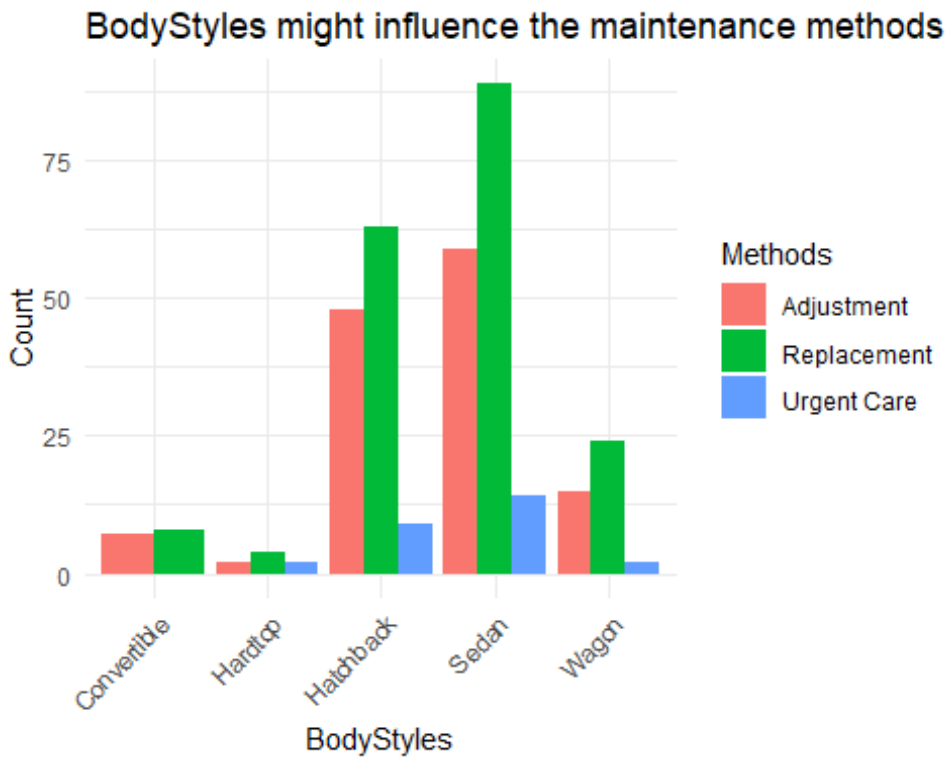
Question 4

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.3
```

```
## Warning: package 'tibble' was built under R version 4.4.2

## Warning: package 'tidyr' was built under R version 4.4.2

## Warning: package 'readr' was built under R version 4.4.3

## Warning: package 'purrr' was built under R version 4.4.2

## Warning: package 'forcats' was built under R version 4.4.3

## Warning: package 'lubridate' was built under R version 4.4.3

## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ forcats   1.0.0     ✓ readr     2.1.5
## ✓ lubridate 1.9.4     ✓ tibble    3.2.1
## ✓ purrr     1.0.2     ✓ tidyr     1.3.1
## ── Conflicts ────────────────────────────────────────── ──
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors

# Normalize column names by cleaning up column names to avoid errors when
calling column names with extra spaces
colnames(maintenance_data) <- trimws(colnames(maintenance_data))

# Clean and standardize Methods
maintenance_data <- maintenance_data %>%
  mutate(
    ErrorCodes = str_trim(tolower(as.character(ErrorCodes))),
    Troubles = tolower(as.character(Troubles)),
    Methods = str_to_title(str_trim(as.character(Methods)))  # Normalize for
plotting
  )

# Convert error descriptions to numeric and non-matching values will be
assigned NA
maintenance_data <- maintenance_data %>%
  mutate(ErrorCodes = case_when(
    ErrorCodes %in% c("no error", "0") ~ 0,
    ErrorCodes %in% c("engine failure", "engine fails", "1") ~ 1,
    ErrorCodes %in% c("other component failure", "other vehicle component
fails", "-1") ~ -1,
    TRUE ~ NA_real_
  ))

# Filter vehicles that had trouble and suspected
trouble_vehicles <- maintenance_data %>%
```

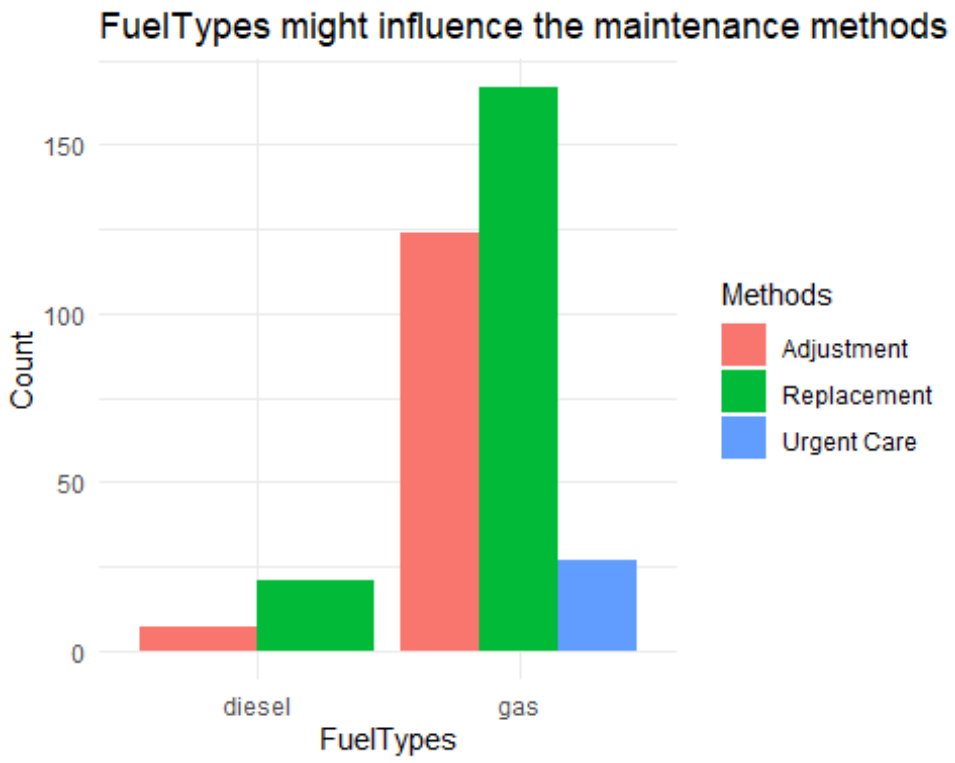
```r
  filter(
    ErrorCodes != 0 | str_detect(Troubles, "suspected")
  )

#Make sure each value appears only once before performing the join and
prevent many-to-many relationship warnings
automobile_data <- automobile_data %>% distinct(PlateNumber, .keep_all =
TRUE)
engine_data <- engine_data %>% distinct(EngineModel, .keep_all = TRUE)



#Merge data based on PlateNumber and EngineModel keys
merged_data <- trouble_vehicles %>%
  left_join(automobile_data, by = "PlateNumber") %>%
  left_join(engine_data, by = "EngineModel")

# Clean and normalize BodyStyles
merged_data <- merged_data %>%
  mutate(BodyStyles = str_to_title(str_trim(as.character(BodyStyles))))

# Force factor levels to include all method types
merged_data$Methods <- factor(
  merged_data$Methods,
  levels = c("Adjustment", "Replacement", "Urgent Care")
)

# Also factor BodyStyles for consistent x-axis
merged_data$BodyStyles <- as.factor(merged_data$BodyStyles)

# Plot
ggplot(merged_data, aes(x = BodyStyles, fill = Methods)) +
  geom_bar(position = "dodge") +
  labs(title = "BodyStyles might influence the maintenance methods",
       x = "BodyStyles", y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Findings

Based on the chart, it can be seen that **Sedans and Hatchbacks are the two most common body styles with troubles**, with the maintenance method being Replacement. Meanwhile, the Convertible, Hardtop, and Wagon models have fewer problems. The Urgent Care maintenance method is very low frequency and does not occur in Convertible body styles.

```r
# Plot FuelTypes vs Maintenance Methods
if (nrow(merged_data) > 0) {
  ggplot(merged_data, aes(x = FuelTypes, fill = Methods)) +
    geom_bar(position = "dodge") +
    labs(title = "FuelTypes might influence the maintenance methods",
         x = "FuelTypes", y = "Count") +
    theme_minimal()
} else {
  cat("No data available to plot FuelTypes.\n")
}
```



Findings

Based on the chart above, it can be seen that **gasoline vehicles have the most significant problems**, leading to high replacements and adjustments. It is entirely *consistent with the results found in Question 3*, as gasoline vehicles have more frequent problems than diesel vehicles. Meanwhile, diesel vehicles rarely have problems and no urgent care cases.