WESTERN SYDNEY
UNIVERSITY

SCHOOL of: Computer, Data and Mathematical Sciences

ASSIGNMENT COVER SHEET

STUDENT DETAILS

Name: Nguyen Hue Nghi	**Student ID:** 22179455

SUBJECT AND TUTORIAL DETAILS

Subject Name: Analystics Programming		**Subject code:** COMP1013
Tutorial Group:	**Day:** Sunday	**Time:** 15:30 - 18:45

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

ASSIGNMENT DETAILS

Title: Assignment	**Length:** 2500 words
Due Date: 21/11/2025	**Date submitted:** 21/11/2025

Home campus: Vietnam

DECLARATION

By submitting your work using this link you are certifying that:

☒ You hold a copy of this submission if the original is lost or damaged.

☒ No part of this submission has been copied from any other student's work or from any other third party (including generative AI) except where due acknowledgment is made in the submission.

☒ No part of this submission has been submitted by you in another (previous or current) assessment, except where appropriately referenced, and with prior permission from the teacher/tutor/supervisor/ Subject Coordinator for this subject.

☒ No part of this submission has been written/produced for you by any other person or technology except where collaboration has been authorised by the teacher/tutor/ supervisor/Subject Coordinator either in the assessment resources section of the Learning Guide for this assessment task, in the instructions for this assessment task, or through vUWS.

☒ You are aware that this submission will be reproduced and submitted to detection software programs for the purpose of investigating possible breaches of the Student Misconduct Rule, for example, plagiarism, contract cheating, or unauthorised use of generative AI. Turnitin or other tools of investigation may retain a copy of the submission for the purposes of future investigation.

☒ You will not make this submission available to any other person unless required by the University.

***Instructions**: Please complete the requested details in the form, save it and convert to PDF before adding your signature below*

Student signature: Nguyen Hue Nghi

Note: An examiner or lecturer/tutor has the right to not mark this assignment if the above declaration has not been completed. Staff may contact you for permission to share a de-identified extract or copy of your submission with students or staff for teaching purposes, following guidelines for requesting and sharing exemplar assessment tasks.

AP-T225WSD-1_Project Trimester 2 2025

Nguyen Hue Nghi- 22179455

2025-11-21

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.

Question 1. Write the code to inspect the data structure and present the data

1.1 The missing values in the dataset were written as “?”, replace any “?” with NA Write code to check: after replacing, how many rows were affected in total? Does this change alter the data distribution?

```
# Replace ? with NA when reading data
engine_df <- read_csv("Engine.csv", na = "?")
auto_df <- read_csv("Automobile.csv", na = "?")
maint_df <- read_csv("Maintenance.csv", na = "?")

# total rows were affected
rows_affected_engine <- engine_df %>%
  filter(if_any(everything(), is.na)) %>%
  nrow()

rows_affected_auto <- auto_df %>%
  filter(if_any(everything(), is.na)) %>%
  nrow()

rows_affected_maint <- maint_df %>%
  filter(if_any(everything(), is.na)) %>%
  nrow()

total_affected <- rows_affected_engine+ rows_affected_auto +
rows_affected_maint
print(total_affected)

## [1] 6
```

This change does not alter the data distribution but to clean and improve it due to:

- Only representation changing: Replacing ? with NA is a data cleaning step. This change only clean the data so the R can correctly identify these values as “missing” rather than treating them as valid character strings

- Restoration of correct data types: If ? appear in numeric columns such as Horsepower, the whole column will be misunderstood as character. So changing it to NA, columns will restore to numeric

- Preservation of observed values: This step does not adding, removing, or modifying any of the existing valid data points. Therefore, the frequency, range, and shape of the distribution for the known values remain exactly the same.

1.2 Convert categorical variables BodyStyles, FuelTypes, ErrorCodes to factors

```
# Convert the categorical variables of the 3 datasets into factors
auto_df$BodyStyles <- as.factor(auto_df$BodyStyles)
engine_df$FuelTypes <- as.factor(engine_df$FuelTypes)
maint_df$ErrorCodes <- as.factor(maint_df$ErrorCodes)
# Check again to see if they have become factors
str(auto_df$BodyStyles)

##  Factor w/ 5 levels "convertible",..: 1 3 4 4 4 4 5 4 3 4 ...

str(engine_df$FuelTypes)

##  Factor w/ 2 levels "diesel","gas": 2 2 2 2 2 2 2 2 2 2 ...

str(maint_df$ErrorCodes)

##  Factor w/ 3 levels "-1","0","1": 1 1 1 3 1 3 3 2 1 1 ...
```

Benefits of switching to factor

- Helps statistical and plotting functions (ggplot2) understand that these are separate groups to color or divide columns, instead of mistaking them as text or numeric.
- Especially with ErrorCodes (error -1, 0, 1), if not switched to factor, R will understand them as numbers and calculate addition, subtraction, multiplication and division (e.g., average of error codes), which is logically meaningless.

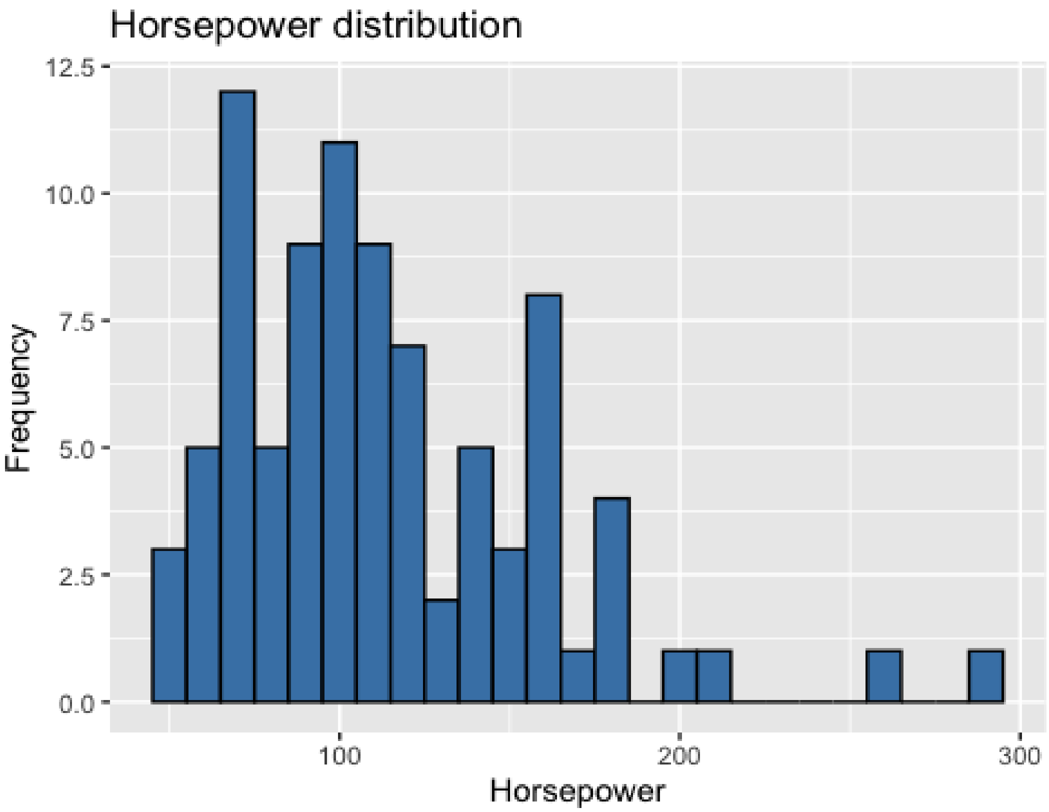
1.3 Replace the missing values in column Horsepower with the median horsepower; Select the appropriate chart type and display: horsepower distribution.

```
# Calculate median, excluding NA (because including it will make it difficult
to determine the median)
median_hp <- median(engine_df$Horsepower, na.rm = TRUE)
# Replace NA with median
engine_df <- engine_df %>%
  mutate( Horsepower = replace_na(Horsepower, median_hp)
)
# Check if everything has been replaced
print(engine_df$Horsepower)

##  [1] 111 154 102 115 110 140 160 101 121 121 182  48  70  68 102  88 145
58  76
## [20]  60  86 101 100  78  90 176 262  68 101 135  84  64 120  72 123 155
184 184
## [39] 175  68 102 116 145 116  69  55  97 152 160 200  97  95  95 142  68
```

```
143 207
## [58] 288 102 110 160  69  73  82  94 111  62  70  56 112 116  92  73 161
156 156
## [77]  52  85  68 100  90  88 114 162 160 134 106 114

# Visualize the Horsepower distribution using histogram
# Using plot in library("ggplot2")
ggplot(data = engine_df) +
  geom_histogram(mapping = aes(x = Horsepower), binwidth = 10, col = "black",
fill="steelblue") +
  labs(title = "Horsepower distribution",
       x = "Horsepower",
       y = "Frequency")
```



Because Horsepower is continuous from 48 to 288 so histogram is the standard and most intuitive tool to display the frequency distribution. It shows us where the data's peak, spread and shape of the distribution (symmetrical or skewed).

Looking at the graph above, we see that the graph is right-skewed, meaning that most of the cars have low to medium horsepower and it decreases as the horsepower increases. The most common horsepower is in the range of 60-100. This shows that most of the popular cars are in this range to save fuel and cost.

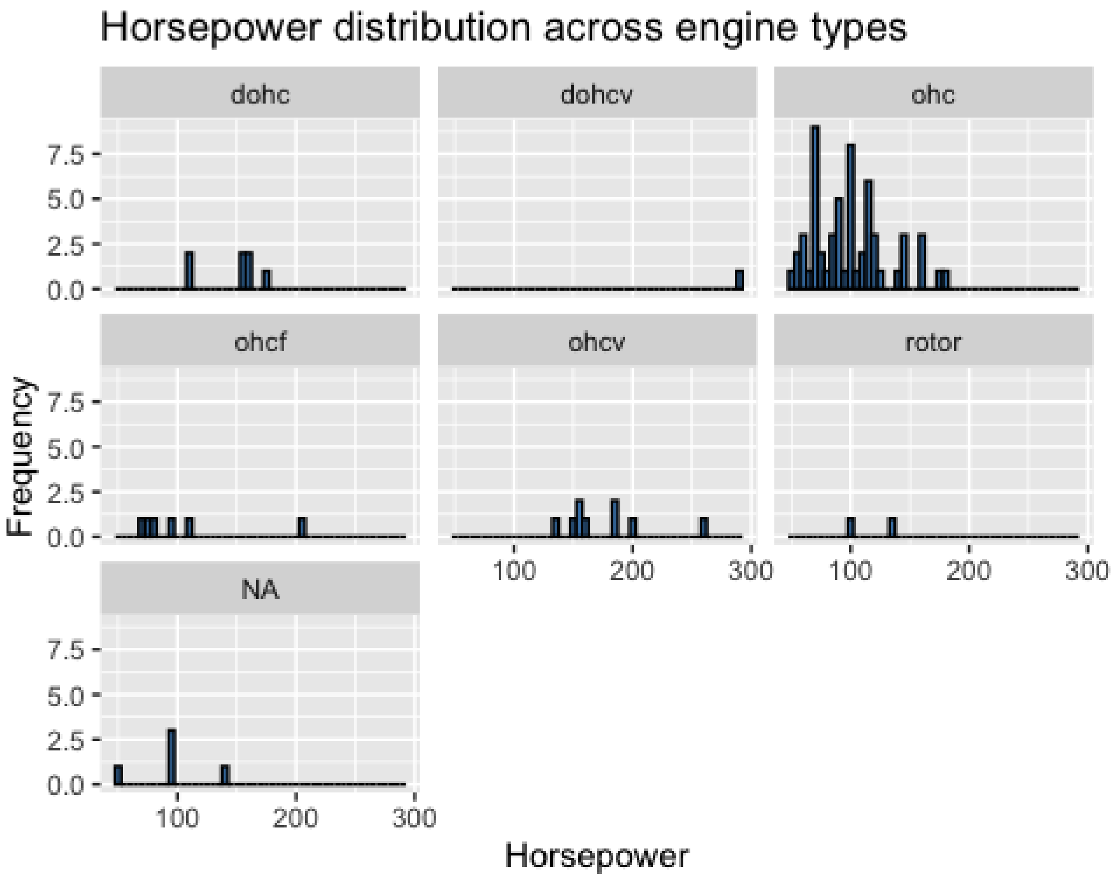
Question 2

2.1 Write the code to analyse the distribution of the horsepower across the engine types

```
# Observe means Horsepower divided by EngineType before plotting
aggregate(Horsepower~EngineType, engine_df, mean)
```

```
##   EngineType Horsepower
## 1       dohc  147.42857
## 2      dohcv  288.00000
## 3        ohc   99.63793
## 4       ohcf  106.00000
## 5       ohcv  176.11111
## 6      rotor  118.00000
```

```
# Visualize Horsepower distribution across EngineTypes using histogram
ggplot(data = engine_df) +
  geom_histogram(mapping = aes(x = Horsepower), binwidth = 5, col = "black",
fill="steelblue", na.rm = TRUE) +
  facet_wrap(~ EngineType) +
  labs(title = "Horsepower distribution across engine types",
       x = "Horsepower",
       y = "Frequency")
```



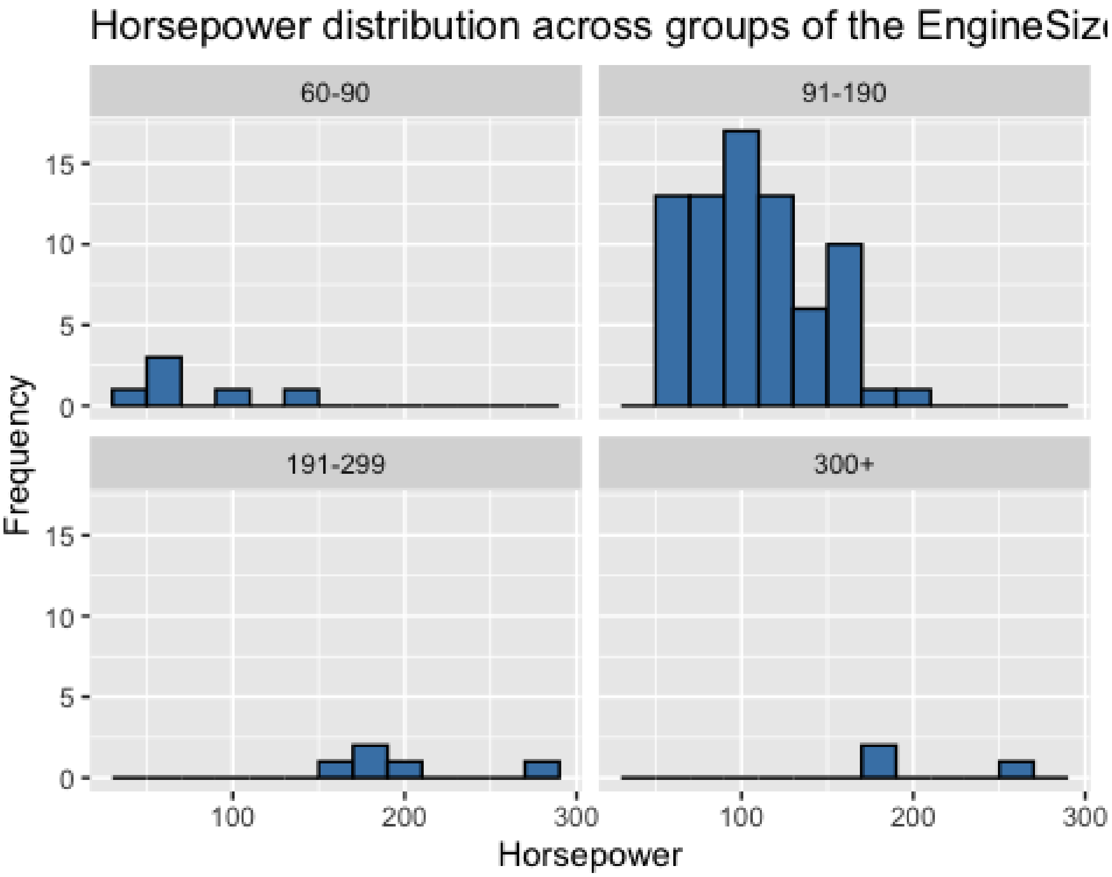
2.2 Write the code to investigate the distribution of the horsepower across the groups of the engine sizes (e.g., 60-90, 91-190, 191-299, 300+).

```
# Group engine sizes
data_with_groups <- engine_df %>%
  filter(!is.na(EngineSize)) %>% # Filter out car with NA
  mutate(
    EngineSize_Group = cut(EngineSize, # Divide EngineSize into discrete bind
      breaks = c(60, 90, 190, 299, Inf),
      labels = c("60-90", "91-190", "191-299", "300+"),
      right = TRUE, # 60 < x <= 90
      include.lowest = TRUE # 60 <= x <= 90
    )
  )

# Check if group division is correct
print(data_with_groups)
```

```
## # A tibble: 88 × 9
##    EngineModel EngineType NumCylinders EngineSize FuelSystem Horsepower
##    <chr>       <chr>      <chr>             <dbl> <chr>           <dbl>
##  1 E-0001      dohc       four                130 mpfi              111
##  2 E-0002      ohcv       six                 152 mpfi              154
##  3 E-0003      ohc        four                109 mpfi              102
##  4 E-0004      ohc        five                136 mpfi              115
##  5 E-0005      ohc        five                136 mpfi              110
##  6 E-0006      ohc        five                131 mpfi              140
##  7 E-0007      ohc        five                131 mpfi              160
##  8 E-0008      ohc        four                108 mpfi              101
##  9 E-0009      ohc        six                 164 mpfi              121
## 10 E-0010      ohc        six                 164 mpfi              121
## # ℹ 78 more rows
## # ℹ 3 more variables: FuelTypes <fct>, Aspiration <chr>, EngineSize_Group
<fct>
```

```
# Viasualize horsepower distribution across the groups of engine sizes using
histogram
ggplot(data = data_with_groups) +
  geom_histogram(mapping = aes(x = Horsepower), binwidth = 20, col = "black",
fill="steelblue", na.rm = TRUE) +
  facet_wrap(~ EngineSize_Group) + # tách theo enginesize group
  labs(title = "Horsepower distribution across groups of the EngineSizes",
       x = "Horsepower",
       y = "Frequency")
```



2.3 Explain findings

- Engine type: The horsepower distribution is very different. Ohc engines have the widest distribution, ranging from low to medium horsepower. More complex engine types such as ohcv and dohcv tend to concentrate at significantly higher horsepower levels. Rotor engines have a unique horsepower distribution, concentrated around the 100-135 horsepower.

- Engine size: There is a clear positive relationship as engine size increases (from 60-90 to 300+), the horsepower distribution shifts significantly to the right (higher horsepower). The '300+' group has a distinctly high horsepower.

Question 3

3.1 Do diesel cars have higher average CityMpg than gasoline cars? Provide statistical evidence.

```
# Handle duplicates before joining
engine_df_clean <- engine_df %>%
  distinct(EngineModel, .keep_all = TRUE) # Remove duplicate rows while
keeping other columns

# Join auto_df and engine_df_clean using EngineModel
auto_engine_df <- left_join(auto_df, engine_df_clean, by = "EngineModel")


# Observe means CityMpg divided by FuelTypes before plotting
aggregate(CityMpg~FuelTypes,auto_engine_df, mean)

##   FuelTypes  CityMpg
## 1    diesel 30.30000
## 2       gas 24.67935

# Use T-test
# Hypothesis: H0: Diesel < Gas ; H1:Diesel > Gas
t_test_result <- t.test(CityMpg ~ FuelTypes,
                        data = auto_engine_df,
                        var.equal = FALSE,
                        alternative = "greater")

print(t_test_result)

##
##  Welch Two Sample t-test
##
## data:  CityMpg by FuelTypes
## t = 3.6237, df = 23.015, p-value = 0.0007118
## alternative hypothesis: true difference in means between group diesel and
```

```
group gas is greater than 0
## 95 percent confidence interval:
##  2.962395      Inf
## sample estimates:
## mean in group diesel    mean in group gas
##             30.30000             24.67935
```

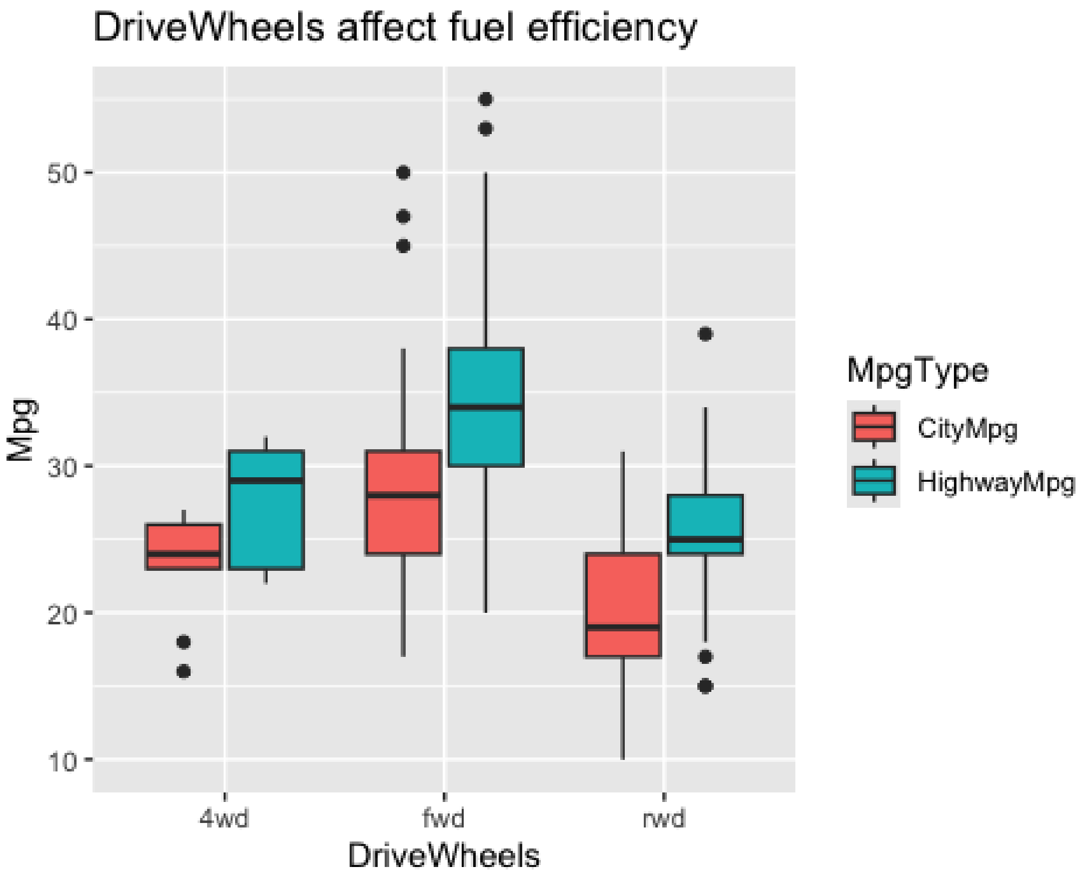
Because p-value = 0.0007118 < 0.05, we reject H0. Therefore, diesel cars have higher average CityMpg than gasoline cars.

3.2 How does DriveWheels affect fuel efficiency (CityMpg and HighwayMpg)? Elaborate on the findings

```
# Observe means of fuel efficiency divided by drivewheel before plotting
city_mean =aggregate(CityMpg~DriveWheels,auto_engine_df, mean)
high_mean =aggregate(HighwayMpg~DriveWheels,auto_engine_df, mean)

# Group mpg for plotting
mpg_group <- auto_df %>%
  select(DriveWheels, CityMpg, HighwayMpg) %>% # Select a set of columns
  pivot_longer(cols = c(CityMpg, HighwayMpg), # Specify which columns should
be merged
               names_to = "MpgType",
               values_to = "Mpg")

# Viasualize using boxplot
ggplot(data = mpg_group ) +
  geom_boxplot(mapping = aes(x = DriveWheels, y = Mpg,fill = MpgType )) +
  labs(title = "DriveWheels affect fuel efficiency",
       x = "DriveWheels",
       y ="Mpg")
```



Based on the statistics and boxplot, front-wheel drive (fwd) is the most fuel-efficient because the average fuel consumption of CityMpg (miles per gallon in the city) and HighwayMpg (miles per gallon on the highway) is significantly higher than the other two. In addition, the dispersion of fwd is also the widest, showing that there are many different types of front-wheel drive vehicles with similar fuel consumption. Rear-wheel drive (rwd) is the least fuel-efficient because the average distance traveled per gallon is the lowest. 4-wheel drive (4wd) is in the middle because the fuel consumption is higher than rwd but lower than fwd. However, this may not be accurate in reality because the number of 4wd is small (based on the height of the boxplot). Also, HighwayMpg is always greater than CityMpg because on the highway cars can run faster and do not have to stop suddenly.

3.3 Filter out those engines in the dataset that have trouble or are suspected of having trouble; what are the top 5 most common troubles related to the engines? Do the troubles differ between engine types?

```r
# Join 3 dataset
full_df <- maint_df %>%
  left_join(auto_df, by = "PlateNumber") %>%
  left_join(engine_df_clean, by = "EngineModel")

# Filter engine fault (ErrorCodes = 1)
```

```r
engine_troubles_df <- full_df %>%
  filter(ErrorCodes == "1")

# Recheck if it is equal to 1
print(engine_troubles_df)
```

```
## # A tibble: 182 × 26
##       ID PlateNumber Date       Troubles   ErrorCodes Price Methods
Manufactures
##    <dbl> <chr>       <chr>      <chr>      <fct>      <dbl> <chr>   <chr>
##  1     4 53N-001     15/05/2024 Ignition … 1            180 Adjust… Alfa-
romero
##  2     6 53N-002     15/02/2024 Cylinders  1           1000 Replac… Alfa-
romero
##  3     7 53N-002     16/03/2024 Ignition … 1            180 Adjust… Alfa-
romero
##  4    12 53N-004     18/06/2024 Ignition … 1            180 Adjust… Audi
##  5    14 53N-004     19/08/2024 Cylinders  1           1000 Replac… Audi
##  6    15 53N-005     20/08/2024 Noise (fi… 1             10 Adjust… Audi
##  7    18 53N-008     19/01/2024 Ignition   1            180 Adjust… Audi
##  8    20 53N-008     19/05/2024 Cylinders  1           1000 Replac… Audi
##  9    21 53N-008     19/07/2024 Noise (fi… 1             10 Adjust… Audi
## 10    26 53S-002     19/05/2024 Oil filter 1             90 Replac… Bmw
## # ℹ 172 more rows
## # ℹ 18 more variables: BodyStyles <fct>, DriveWheels <chr>,
## #   EngineLocation <chr>, WheelBase <dbl>, Length <dbl>, Width <dbl>,
## #   Height <dbl>, CurbWeight <dbl>, EngineModel <chr>, CityMpg <dbl>,
## #   HighwayMpg <dbl>, EngineType <chr>, NumCylinders <chr>, EngineSize
<dbl>,
## #   FuelSystem <chr>, Horsepower <dbl>, FuelTypes <fct>, Aspiration <chr>
```

```r
# Find the top 5 most common errors
top_5_troubles <- engine_troubles_df %>%
  count(Troubles) %>%             # Count the number of each trouble
  arrange(desc(n)) %>%            # Sort in descending order
  head(5) %>%
  pull(Troubles)

print(top_5_troubles)
```

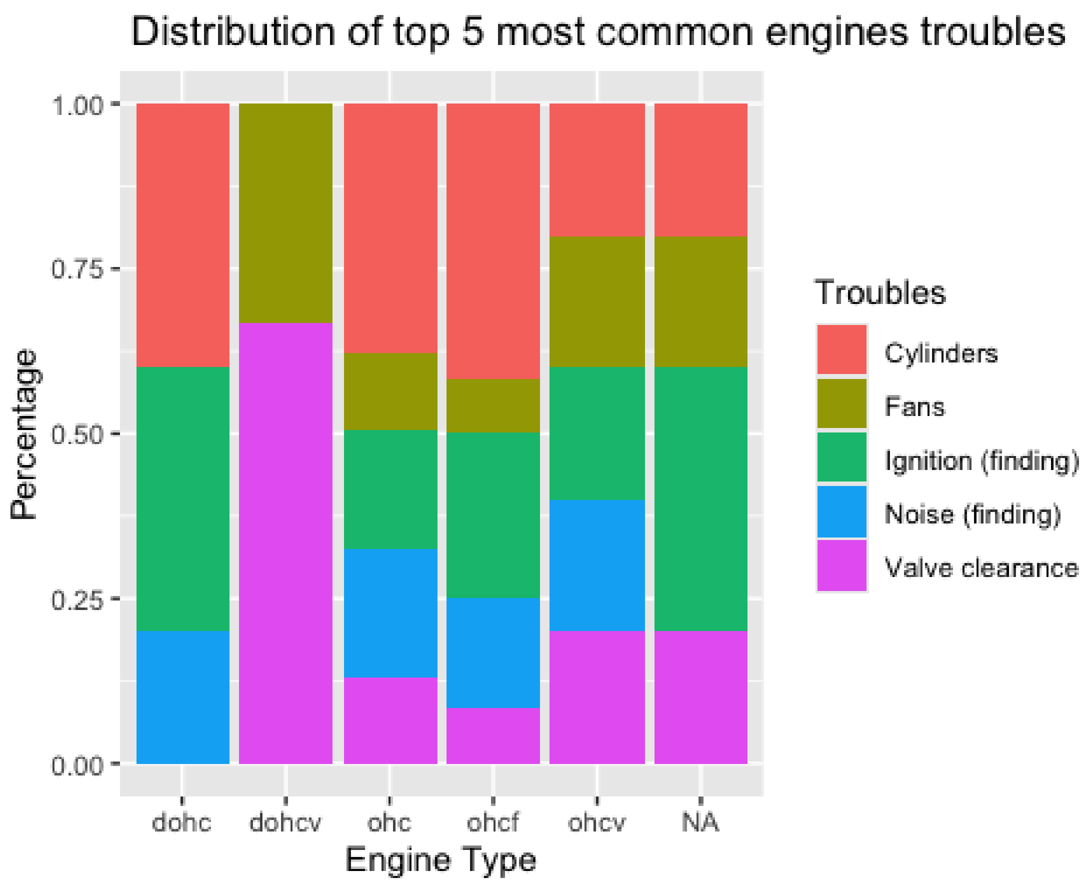
```
## [1] "Cylinders"          "Ignition (finding)" "Noise (finding)"
## [4] "Valve clearance"    "Fans"
```

```r
# Extract data of these 5 (Do 5 troubles differ between engine types?)
top_5_df <- engine_troubles_df %>%
  filter(Troubles %in% top_5_troubles)
print(top_5_df)
```

```
## # A tibble: 107 × 26
##       ID PlateNumber Date       Troubles   ErrorCodes Price Methods
Manufactures
```

```
##     <dbl> <chr>       <chr>      <chr>      <fct>      <dbl> <chr>    <chr>
##  1      4 53N-001     15/05/2024 Ignition … 1            180 Adjust… Alfa-
romero
##  2      6 53N-002     15/02/2024 Cylinders  1           1000 Replac… Alfa-
romero
##  3      7 53N-002     16/03/2024 Ignition … 1            180 Adjust… Alfa-
romero
##  4     12 53N-004     18/06/2024 Ignition … 1            180 Adjust… Audi
##  5     14 53N-004     19/08/2024 Cylinders  1           1000 Replac… Audi
##  6     15 53N-005     20/08/2024 Noise (fi… 1             10 Adjust… Audi
##  7     20 53N-008     19/05/2024 Cylinders  1           1000 Replac… Audi
##  8     21 53N-008     19/07/2024 Noise (fi… 1             10 Adjust… Audi
##  9     45 53S-015     03/11/2023 Valve cle… 1            175 Adjust… Dodge
## 10     48 53S-018     13/02/2024 Fans       1             55 Replac… Dodge
## # ℹ 97 more rows
## # ℹ 18 more variables: BodyStyles <fct>, DriveWheels <chr>,
## #   EngineLocation <chr>, WheelBase <dbl>, Length <dbl>, Width <dbl>,
## #   Height <dbl>, CurbWeight <dbl>, EngineModel <chr>, CityMpg <dbl>,
## #   HighwayMpg <dbl>, EngineType <chr>, NumCylinders <chr>, EngineSize
<dbl>,
## #   FuelSystem <chr>, Horsepower <dbl>, FuelTypes <fct>, Aspiration <chr>
```

```
# Plotting 5 trouble with different type of engine to see if it is different
ggplot(top_5_df) +
  geom_bar(aes(x = EngineType, fill = Troubles), position = "fill") + #
return % for comparison
  labs(title = " Distribution of top 5 most common engines troubles ",
       x = "Engine Type",
       y = "Percentage",
       fill = "Troubles")
```



```
# Chi-squared test
# H0: There is no difference; H1: There is differences
# Contigency table
table_troubles <- table(top_5_df$EngineType, top_5_df$Troubles)
print(table_troubles)

##
##         Cylinders Fans Ignition (finding) Noise (finding) Valve clearance
##   dohc          2    0                  2               1               0
##   dohcv         0    1                  0               0               2
##   ohc          29    9                 14              15              10
##   ohcf          5    1                  3               2               1
##   ohcv          1    1                  1               1               1

# Chi-squared test
test_result <- chisq.test(table_troubles)
print(test_result)

##
##  Pearson's Chi-squared test
##
## data:  table_troubles
## X-squared = 13.564, df = 16, p-value = 0.6312
```

Because p-value = 0.6312 > 0.05, we cannot reject H0. Therefore, troubles do not differ between engine types.

Question 4

4.1 Write the code to show which error type (ErrorCodes) occurs most frequently

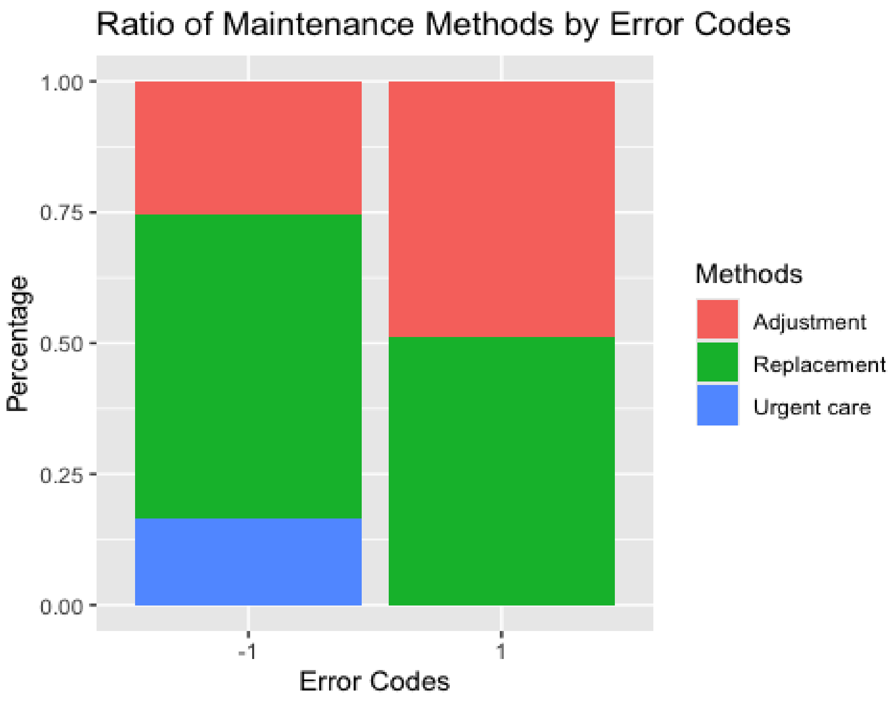
```
mostly_troubles <- engine_troubles_df %>%
  count(Troubles) %>%
  arrange(desc(n)) %>%
  head(1) %>%
  pull(Troubles)
print(mostly_troubles)

## [1] "Cylinders"
```

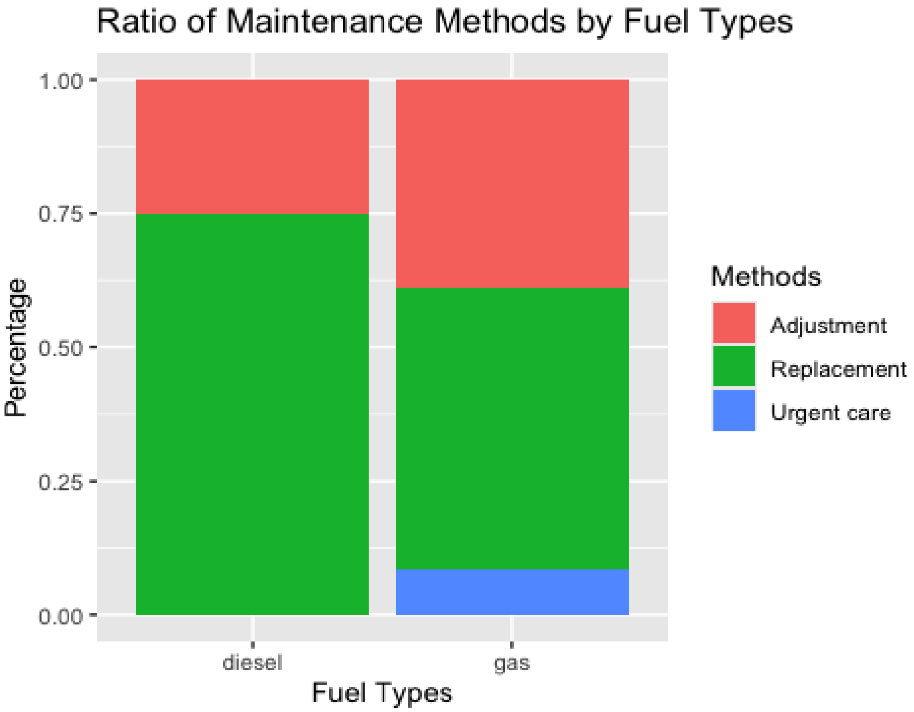
4.2 Write the code to analyze the factors that might influence the maintenance methods (Urgent care, Adjustment, Replacement) for the trouble vehicles (confirmed or suspected) in the dataset. Any factors in the dataset, such as BodyStyles, FuelTypes, and ErrorCodes, can be considered. Pick 2 of the factors and explain if there is a trend that explains the variation

```
# Filter out vehicles with ErrorCodes = 0
vehicles_trouble <- full_df %>%
  filter(ErrorCodes != "0")

# Factor 1: ErrorCodes
ggplot(vehicles_trouble) +
geom_bar(mapping = aes(x = ErrorCodes, fill = Methods), position = "fill") +
  labs(title = "Ratio of Maintenance Methods by Error Codes",
       x = "Error Codes",
       y = "Percentage")
```



```r
# Factor 2: FuelTypes
ggplot(vehicles_trouble) +
geom_bar(mapping = aes(x = FuelTypes, fill = Methods), position = "fill") + #
fill về tỉ lệ 1
  labs(title = "Ratio of Maintenance Methods by Fuel Types",
       x = "Fuel Types",
       y = "Percentage")
```



For ErrorCodes: Urgent care appears to only occur in other vehicle component fails (error -1) with 15%. While the engine failure group (error 1) is mainly handled by adjustment (50%) and replacement (50%). This indicates error -1 such as brake usually affect directly to the safety of driver, so it is classified as urgent, while engine errors (error 1) are often technical failures requiring component replacement.

For FuelTypes: Diesel engine has higher probability of replacement while gas engine has higher adjustment ratio. This may suggest that problems on diesel vehicles are generally more complicated or that diesel engine parts that fail are more difficult to repair than gas engines, thus requiring more frequent replacement.