# AP Assignment 3

Nguyen Huu Tri

2025-11-18

## Introduction

This report explores vehicle performance and maintenance data from the Vietnam Vehicle Maintenance Centre to find patterns in different engines. Using R code, the report conducts an analysis on the given data structure and observe the distribution of horsepower across engine types. After that, a comparison between diesel and gas is performed to evaluate their influences in fuel efficiency. Subsequently, different drivetrains are examined to see how much they affect fuel economy. The final section of the report analyzes the error types of the vehicles and discover trends that could inform more efficient maintenance planning.

## Package and Data Loading

The report will use additional packages to support the data analysis, including “tidyverse”, “dplyr”, and “ggplot2”.

The Vietnam Vehicle Maintenance Centre provided 3 data frames engine, automobile, and maintenance for vehicle data analysis. The data set engine provides information about different engine models and their configurations. The data set automobile contains physical and performance characteristics of cars like manufacturer, body style, drive wheels, engine location, and engine model used. Finally, the data set maintenance includes problem diagnostics and appropriate service methods to repair the vehicles.

```
#Uploads 3 given csv files as data frames
engine = read.csv("Engine.csv")
automobile = read.csv("Automobile.csv")
maintenance = read.csv("Maintenance.csv")

#Summary of 3 data frames
str(engine)
```

```
## '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" ...
```

```
str(automobile)
```

```
## '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 ...
```

```
str(maintenance)
```

```
## '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"
...
```

## Task 1: Data Inspection and Cleaning

The purpose of the first task is to inspect and clean the data frames so they can analyzed properly without encountering errors. The cleaning process will follow the individual questions step by step.

- There are missing values in the given data sets written as “?” and they need to be replaced with “NA” to match with the other missing values across three data frames for R to function properly.

```
#Swaps values that have "?" with "NA" on all rows.
engine[engine == "?"] <- NA
automobile[automobile == "?"] <- NA
maintenance[maintenance == "?"] <- NA
```

- After converting the missing values into “NA”, the next step is to see how many rows are affected in total. This also counts the original “NA” values that did not need the above code to be replaced. Running the code gives us 6 affected rows from the

engine data frame and 28 from the maintenance data frame, with a total of 34 affected rows. Replacing "?" with "NA" does not change the data distribution because there were no real numeric values initially. Synchronizing the missing values with "NA" allows R to interpret them correctly and does not change the existing data.

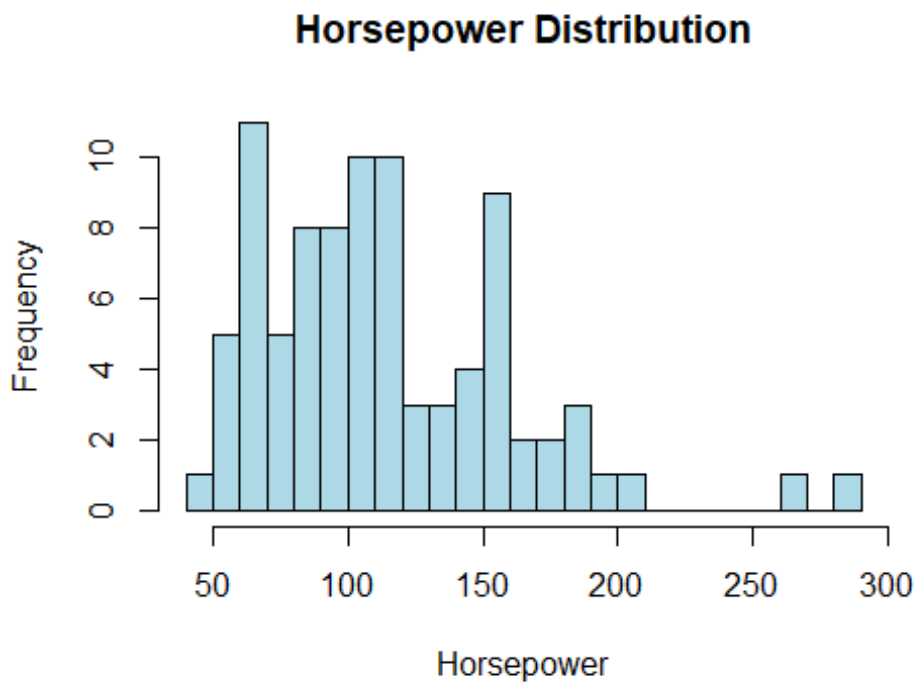
```
#Counts the incomplete rows that contain at least 1 missing value ("NA").
sum(!complete.cases(engine))

## [1] 6

sum(!complete.cases(automobile))

## [1] 0

sum(!complete.cases(maintenance))

## [1] 28

#Sums up all the affected rows in 3 data frames.
affected_rows = sum(!complete.cases(engine)) +
                sum(!complete.cases(automobile)) +
                sum(!complete.cases(maintenance))
print(affected_rows)

## [1] 34
```

- We also need to convert BodyStyles, FuelTypes, and ErrorCodes variables to factors so R can interpret them as categories and not strings.

```
#Convert variables into factors
automobile$BodyStyles = factor(automobile$BodyStyles)
engine$FuelTypes = factor(engine$FuelTypes)
maintenance$ErrorCodes = factor(maintenance$ErrorCodes)
```

- For the missing values in column Horsepower, they can be replaced with the median horsepower to as a substitute to fill out the data. Since horsepower is a continuous variable, histograms are chosen because they can display the range and shape of the data clearly. The histogram will automatically ignore missing variables if we do not fill in and the data will be inaccurate.

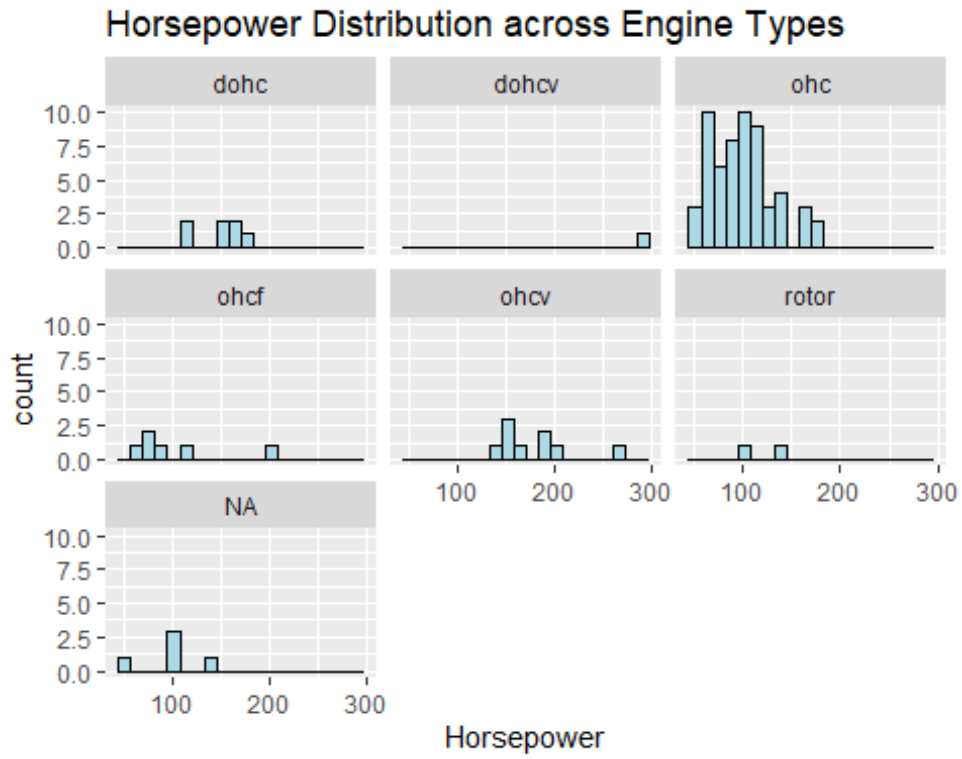
```
#Converts horsepower to numeric values, as some may be characters
engine$Horsepower = as.numeric(engine$Horsepower)
#Replaces NA with median horsepower, ignore NA when calculating median
engine$Horsepower[is.na(engine$Horsepower)] = median(engine$Horsepower, na.rm
= TRUE)

#Displays the horsepower distribution across all engines separated in 20
columns
hist(engine$Horsepower, main = "Horsepower Distribution", xlab =
"Horsepower", col = "lightblue", breaks = 20)
```



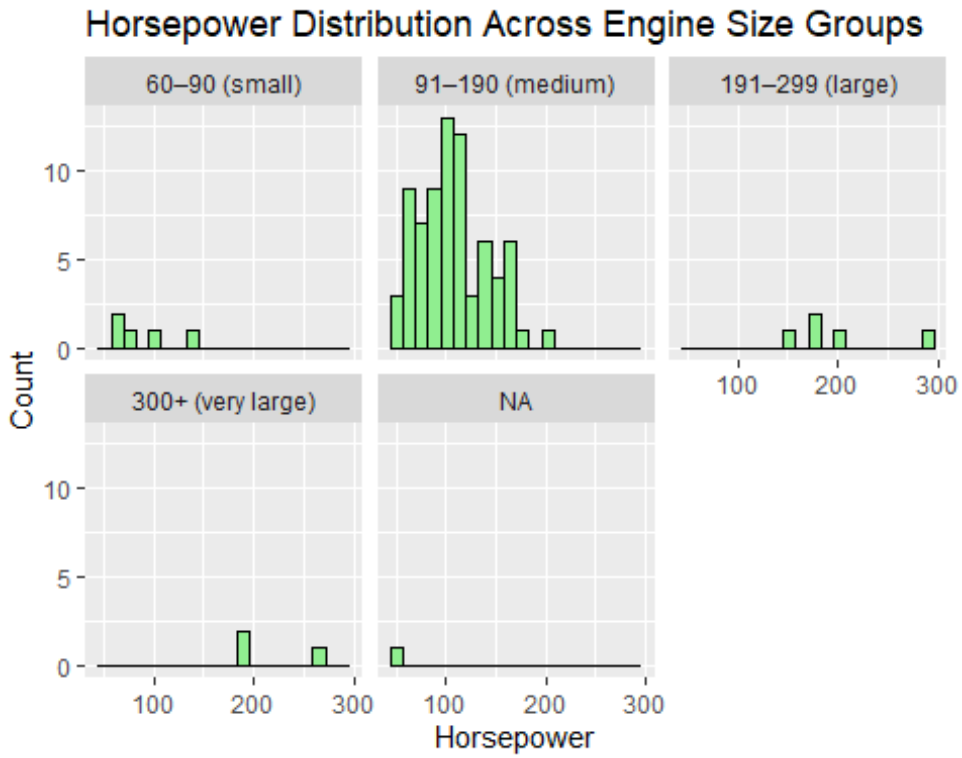

## Task 2: Horsepower Distributions

In this task, horsepower distribution is examined to investigate how it varies across different engine types and sizes. The engine sizes are grouped into 4 categories: small (60-90), medium (91-190), large (191-299), and very large (300+). Comparing these distributions helps determine whether the engine design affects horsepower output.

```
#Histogram of hp across engine types
ggplot(data = engine) +
  geom_histogram(mapping = aes(x = Horsepower),
                 bins = 20, color = "black", fill = "lightblue") +
  facet_wrap(~ EngineType) +
  labs(title = "Horsepower Distribution across Engine Types", x =
"Horsepower", y = "count")
```




```r
#Create engine size groups
engine = engine %>%
  mutate(EngineSizeGroup = cut(
    EngineSize,
    breaks = c(60, 90, 190, 299, Inf),
    labels = c("60-90 (small)", "91-190 (medium)", "191-299 (large)", "300+
(very large)"),
    right = TRUE
  ))

#Histogram of hp across engine sizes
ggplot(data = engine) +
  geom_histogram(mapping = aes(x = Horsepower),
                 bins = 20, color = "black", fill = "lightgreen") +
  facet_wrap(~ EngineSizeGroup) +
  labs(title = "Horsepower Distribution Across Engine Size Groups", x =
"Horsepower", y = "Count"
  )
```




- Based on the histograms, we find that OHC engines show the widest horsepower spread and most of them lie in the 70-140 HP range. This suggests that the engine design is for low or medium powered vehicles. OHCV engines show moderately high horsepower output, sitting at 140-200 HP range, and they tend to focus more in performance. DOHC and OHCF engines have smaller spread, sitting at 120-170 and 60-110 HP mark respectively with one OHCF outlier. The other engine types have very small sample sizes so their patterns may be unreliable.

- For the histograms grouped in engine sizes, we notice a trend of horsepower output increasing as engine size grows. Most vehicles gather around the 91-190 mark, while small and very large engines are rare. Some small engines achieving mid-range power and outliers in other groups show that there are other factors that influence horsepower like number of cylinders and aspiration type.

## Task 3: Fuel Efficiency and Troubles Findings

This task will be split into 2 parts:

**1. Factors influencing fuel consumption**

- To determine whether using different fuel types affect fuel efficiency on city roads (CityMpg), the automation and engine data frames need to be linked together first.

```
#Combines automobile and engine data frames using left_join and EngineModel key
auto_engine = automobile %>% left_join(engine, by = "EngineModel")
```

```
## Warning in left_join(., engine, by = "EngineModel"): Detected an
unexpected many-to-many relationship between `x` and `y`.
## ℹ Row 14 of `x` matches multiple rows in `y`.
## ℹ Row 5 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.
```

- After combining the 2 data sets, a warning shows up because there are multiple rows that have the same engine model. This is acceptable for the automobile data because some cars may have the same engine, and engines with same model numbers are still valid because they have different configurations and out different horsepower levels.

- In the next step, a t-test is performed to see whether the average of CityMpg in gas and diesel cars are significantly different:

H0: difference in means = 0

H1: difference in means != 0

```
#Test whether the mean of the 2 fuel types is significantly different
t.test(CityMpg ~ FuelTypes, data = auto_engine)

##
##  Welch Two Sample t-test
##
## data:  CityMpg by FuelTypes
## t = 3.9004, df = 22.592, p-value = 0.0007392
## alternative hypothesis: true difference in means between group diesel and
group gas is not equal to 0
## 95 percent confidence interval:
##  2.824648 9.218138
## sample estimates:
## mean in group diesel    mean in group gas
##             30.30000             24.27861
```

- The t-test result shows that p-value is smaller than alpha (0.05), so we reject the null hypothesis, meaning there is a difference in the level of fuel consumption on city roads between gasoline and diesel cars. The result also reveals the mean of diesel group, which is higher than that of the gas group. This means diesel engines have better fuel efficiency in the city than gas engines.

- In order to see whether different drive wheels affect fuel efficiency in city and highway roads, an ANOVA test is conducted first to determine if the mean of CityMpg and HighwayMpg in drive wheels are equal or not.

Test 1 (CityMpg):

H0: The mean CityMpg is the same across all DriveWheels groups.

H1: At least 1 DriveWheels group has a different mean CityMpg.

Test 2 (HighwayMpg):

H0: The mean HighwayMpg is the same across all DriveWheels groups.

H1: At least 1 DriveWheels group has a different mean HighwayMpg.

```
#Since the sample size is large (above 50), we can use the aov function to
perform the ANOVA test
#Test 1
anova_city = aov(CityMpg ~ DriveWheels, data = auto_engine)
summary(anova_city)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## DriveWheels   2   3216  1608.0   56.05 <2e-16 ***
## Residuals   218   6254    28.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Test 2
anova_highway = aov(HighwayMpg ~ DriveWheels, data = auto_engine)
summary(anova_highway)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## DriveWheels   2   3944  1971.9   67.09 <2e-16 ***
## Residuals   218   6408    29.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Based on the ANOVA results, the p-values of both tests are significantly smaller than alpha (0.05), so we reject the null hypothesis, meaning at least one drive wheel significantly different in terms of fuel consumption in city and highway roads. But to determine specifically which drive wheel groups differ in fuel efficiency, a TukeyHSD post-hoc test is required.

```
#Test 1
TukeyHSD(anova_city)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = CityMpg ~ DriveWheels, data = auto_engine)
##
## $DriveWheels
##              diff       lwr       upr     p adj
## fwd-4wd  5.084011  0.719369  9.448653 0.0177441
## rwd-4wd -2.774032 -7.195152  1.647087 0.3021867
## rwd-fwd -7.858043 -9.617003 -6.099084 0.0000000
```

```
#Test 2
TukeyHSD(anova_highway)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = HighwayMpg ~ DriveWheels, data = auto_engine)
##
## $DriveWheels
##               diff        lwr       upr     p adj
## fwd-4wd  6.907859   2.489935 11.325783 0.0008253
## rwd-4wd -1.727840  -6.202931  2.747251 0.6338502
## rwd-fwd -8.635699 -10.416132 -6.855267 0.0000000
```

- From the 2 tests, there is 1 pair that does not have a significant difference (p-value is above 0.05) which is rwd-4wd. This means that these two drive wheels have similar fuel efficiency levels in city and highway roads. Meanwhile, the other 2 pairs have a significant difference (p-value is below 0.05), and they are fwd-4wd and rwd-fwd. The fwd-4wd pair has a difference in CityMpg and HighwayMpg of 5.08 and 6.9 respectively, meaning the fwd configuration is more fuel efficient than 4wd. On the other hand, the rwd-fwd has an mpg difference of -7.86 and -8.63 in city and highway, which means rwd consumes more fuel than fwd. As a result, the drive wheel that has the best fuel efficiency is fwd, followed by rwd and 4wd.

**2. Troubles Analysis:**

- In this part, we need to identify the 5 most common troubles and check if these troubles differ between engine types. The cars in the maintenance data frame that have or may have a problem must be filtered to progress. Then, we count and select the top 5 failures that appeared the most to find a pattern.

```
#Creates a subset that only includes error codes different than 0
troubled_engines = subset(maintenance, ErrorCodes != 0)
#Counts and sorts the top 5 troubles
top5_troubles = head(sort(table(troubled_engines$Troubles), decreasing =
TRUE), 5)
#Displays the top 5 troubles
print(top5_troubles)
```

```
##
##          Cylinders            Chassis Ignition (finding)    Noise
(finding)
##                 38                 25                 22
19
##         Worn tires
##                 16
```

- The top 5 most common troubles found in the maintenance record are cylinders, chassis, ignition, noise, and worn tires. To see if these troubles differ between engine types, the automation, engine, and maintenance data sets must be joined together. After that, a table is created and includes only the information about the

engine types and their troubles extracted from the linked data frame. The top 5 troubles are then selected shown from the table.

```
#Joins the data set in an order: maintenance -> automobile -> engine; to
ensure no duplicates
maintain_auto_engine = maintenance %>%
  left_join(automobile, by = "PlateNumber") %>%
  left_join(engine, by = "EngineModel")

## Warning in left_join(., engine, 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.

#Creates a table which includes EngineType and Troubles
engine_troubles = table(maintain_auto_engine$EngineType,
maintain_auto_engine$Troubles)
#Creates a vector of the top 5 troubles
top5 = c("Cylinders", "Chassis", "Ignition (finding)", "Noise (finding)",
"Worn tires")
#Shows the top 5 troubles from the table
engine_troubles[,top5]

##
##         Cylinders Chassis Ignition (finding) Noise (finding) Worn tires
##   dohc          2       3                  2               1          2
##   dohcv         0       0                  0               0          0
##   ohc          30      16                 15              16         13
##   ohcf          5       4                  3               2          1
##   ohcv          1       1                  1               1          0
##   rotor         0       0                  0               0          0
```

- Judging from this table, it seems that the engine types do not necessarily have unique errors, but the OHC type has the highest number of problems recorded and it contains all 5 of the most common troubles listed. This contrasts with the DOHCV and Rotor designs, where they have zero errors recorded in the data set.

## Task 4: Error Types and Maintenance Analysis

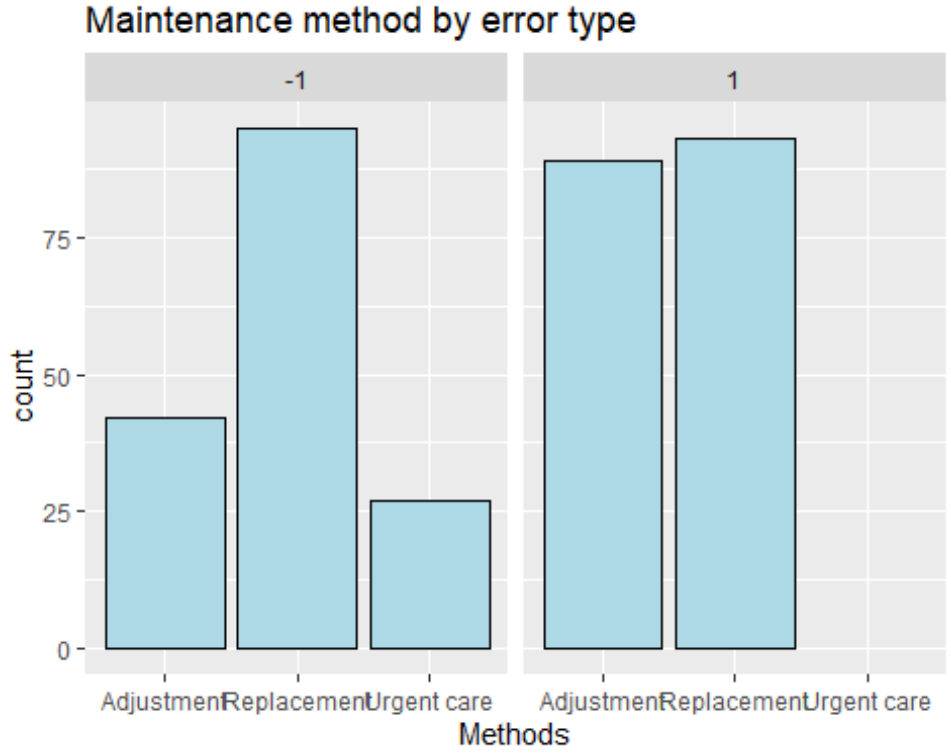
- The final aims to examine the frequency of the error types and factors that might influence maintenance methods. By identifying the common problems, either engine or non-engine related, we can determine which repairs are suitable for specific situations.

```
#Counts the number of errors for each type
table(maintenance$ErrorCodes)

##
##  -1   0   1
## 164  28 182
```

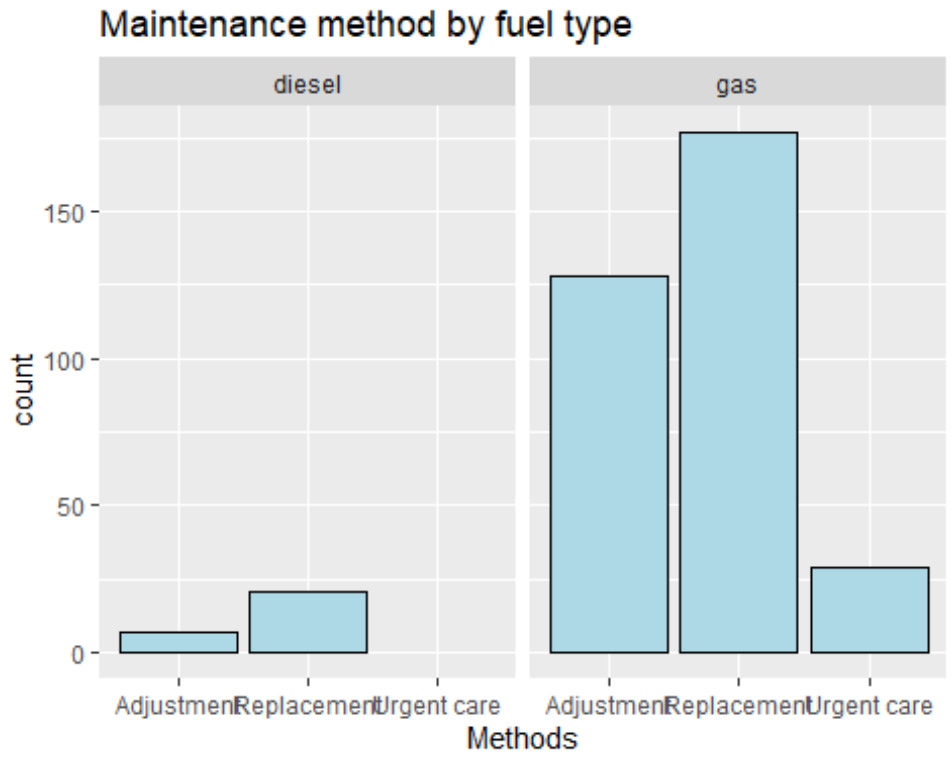
- Out of 374 recorded cases, engine-related failures (ErrorCode: 1) occurred 182 times which is the most frequent, followed by failures due to components outside the engine at 162 times, and 28 non-errors.

- ErrorCodes factor is chosen to analyze its relationships with the maintenance methods because knowing whether the problem comes from the engine or other parts of the car helps to choose the appropriate repair based on the severity or source of the failure. FuelTypes factor is also important to see which fuel requires more attention than the other.

```
#Counts the Methods based on ErrorCodes
table(maintenance$ErrorCodes, maintenance$Methods)

##
##      Adjustment Replacement Urgent care
##   -1         42          95          27
##   0           0           0           0
##   1          89          93           0

#Plots the bar chart using the table above
ggplot(data = subset(maintenance, !is.na(Methods))) +  #remove NA since it
doesnt contain troubles
  geom_bar(mapping = aes(x = Methods),
           color = "black", fill = "lightblue") +
  facet_wrap(~ ErrorCodes) +
  labs(title = "Maintenance method by error type", y = "count")
```



```r
#Counts the Methods based on FuelTypes
table(maintain_auto_engine$FuelTypes, maintain_auto_engine$Methods)

##
##          Adjustment Replacement Urgent care
##   diesel          7          21           0
##   gas           128         177          29

#Plots the bar chart using the table above
ggplot(data = subset(maintain_auto_engine, !is.na(Methods))) +  #remove NA
since it doesnt contain troubles
  geom_bar(mapping = aes(x = Methods),
           color = "black", fill = "lightblue") +
  facet_wrap(~ FuelTypes) +
  labs(title = "Maintenance method by fuel type", y = "count")
```




- Bar charts are used to observe the repair methods distribution in each error type and fuel type clearly. When the error originates from components outside the engine, replacement is the most popular method, followed by adjustment and a smaller number of urgent care cases. However, errors that come from the engine are mainly handled through replacement and adjustment, without the need urgent care. Replacement method saw the most number of records from 2 error types, but non-engine troubles show a greater reliance due to its major gap between it and adjustment.

- When sorting fuel types to see the method distribution, gas vehicles experience more maintenance events than diesel vehicles. This indicates gas cars are more common or they require more maintenance check than diesel cars. Replacement is also the most common method in this category, showing that quick fixes are not enough to solve the issue in most cases. In addition, Gas vehicles saw a considerable amount adjustment methods used and urgent care fixes are rarely required for both fuel types.