COMP1013 Analytics Programming

Huynh gia Bao - 22219215

2025-11-20

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 mayretain 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.

Repository link: https://github.com/huynzabao/Huynh-Gia-Bao__22219215__Assignment__AP-T325WSD-1.git

Task 1: Check and Structure data

The first step of analysis process is to understand the structure and quality of raw data. I will check check the number of rows affected by missing values, convert categorical variables to factor format, and deal with missing Horsepower values by replacing them with the median to reduce the impact of outliers. Then, selecting the chart type to display horsepower distribution.

```
# Read data from file csv, automatically converts "?" values to NA.
engines <- read.csv("Engine.csv", na.strings = "?")
automobile <- read.csv("Automobile.csv", na.strings = "?")
maintenance <- read.csv("Maintenance.csv", na.strings = "?")

# Display the first few rows and structure of each dataset to inspect them
head(automobile)
```

```
##   PlateNumber Manufactures  BodyStyles DriveWheels EngineLocation WheelBase
## 1     53N-001  Alfa-romero convertible         rwd          front      88.6
## 2     53N-002  Alfa-romero   hatchback         rwd          front      94.5
## 3     53N-003         Audi       sedan         fwd          front      99.8
## 4     53N-004         Audi       sedan         4wd          front      99.4
## 5     53N-005         Audi       sedan         fwd          front      99.8
## 6     53N-006         Audi       sedan         fwd          front     105.8
##   Length Width Height CurbWeight EngineModel CityMpg HighwayMpg
## 1  168.8  64.1   48.8       2548      E-0001      21         27
## 2  171.2  65.5   52.4       2823      E-0002      19         26
## 3  176.6  66.2   54.3       2337      E-0003      24         30
## 4  176.6  66.4   54.3       2824      E-0004      18         22
## 5  177.3  66.3   53.1       2507      E-0005      19         25
## 6  192.7  71.4   55.7       2844      E-0005      19         25
```

```
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 ...
```

```
head(engines)
```

```
##   EngineModel EngineType NumCylinders EngineSize FuelSystem Horsepower
## 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
##   FuelTypes Aspiration
## 1       gas        std
## 2       gas        std
## 3       gas        std
## 4       gas        std
## 5       gas        std
## 6       gas      turbo
```

```
str(engines)
```

```
## '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  : int  111 154 102 115 110 140 160 101 121 121 ...
##  $ FuelTypes   : chr  "gas" "gas" "gas" "gas" ...
##  $ Aspiration  : chr  "std" "std" "std" "std" ...
```

```r
head(maintenance)
```

```
##   ID PlateNumber       Date           Troubles ErrorCodes Price     Methods
## 1  1     53N-001 15/02/2024       Break system         -1   110 Replacement
## 2  2     53N-001 16/03/2024       Transmission         -1   175 Replacement
## 3  3     53N-001 15/04/2024   Suspected clutch         -1   175  Adjustment
## 4  4     53N-001 15/05/2024 Ignition (finding)          1   180  Adjustment
## 5  5     53N-001 14/06/2024            Chassis         -1    85 Replacement
## 6  6     53N-002 15/02/2024          Cylinders          1  1000 Replacement
```

```r
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" ...
```

```r
# Check for Affected Rows
# We use complete.cases() to find rows that have no NA values.
# By subtracting this from the total number of rows, we find how many rows are affected.
affected_rows_auto <- nrow(automobile) - sum(complete.cases(automobile))
affected_rows_eng <- nrow(engines) - sum(complete.cases(engines))
affected_rows_maint <- nrow(maintenance) - sum(complete.cases(maintenance))

# Print the results
cat("Number of rows affected in Automobile data:", affected_rows_auto, "\n")
```

```
## Number of rows affected in Automobile data: 0
```

```r
cat("Number of rows affected in Engines data:", affected_rows_eng, "\n")
```

```
## Number of rows affected in Engines data: 6
```

```r
cat("Number of rows affected in Maintenance data:", affected_rows_maint, "\n")
```

```
## Number of rows affected in Maintenance data: 0
```

```r
# Convert to Factors
automobile$BodyStyles <- as.factor(automobile$BodyStyles)
engines$FuelTypes <- as.factor(engines$FuelTypes)
maintenance$ErrorCodes <- as.factor(maintenance$ErrorCodes)
```

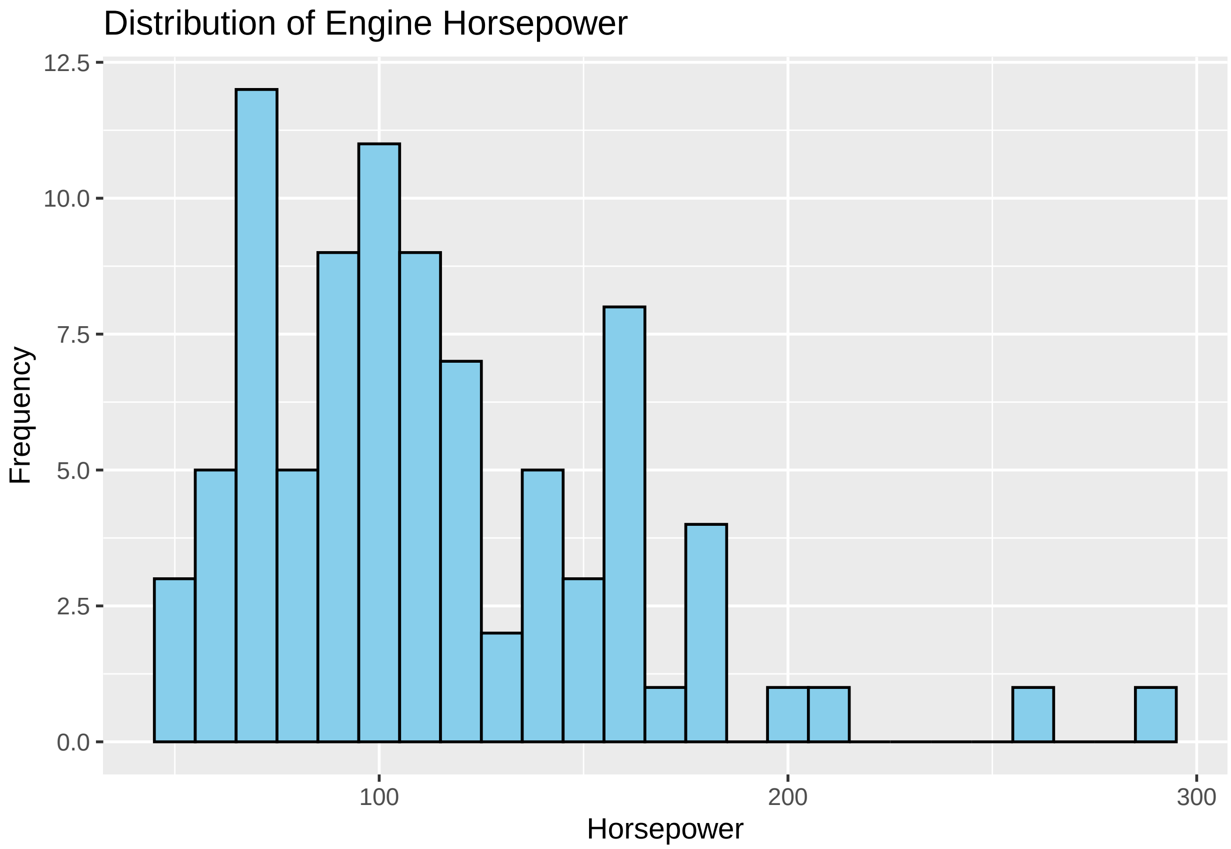
```r
# Impute Missing Horsepower with Median
# First, calculate the median of the Horsepower column, ignoring NA values.
median_horsepower <- median(engines$Horsepower, na.rm = TRUE)
# Find NA values in the Horsepower column and replace them with the median.
engines$Horsepower[is.na(engines$Horsepower)] <- median_horsepower
```

Initial data inspection and preparation revealed that the dataset had several quality issues that needed to be addressed before analysis. Specifically, missing values is denoted by the ? character. It affected a total of 0 rows in the Automobile table, 6 rows in the Engine table, and 0 rows in the Maintenance table. After converting these values to R standard NA format, we addressed the missing data in the Horsepower column by replacing NA values with the median. This approach was chosen to minimize the impact of outliers. Finally, important categorical variables such as BodyStyles, FuelTypes, and ErrorCodes were converted to factor format to ensure that R interpreted them correctly in subsequent statistical analysis and visualization.

Horsepower Distribution

The bar chart below shows the distribution of Horsepower across all engines.

```r
ggplot(engines, aes(x = Horsepower)) +
  geom_histogram(binwidth= 10, fill= "skyblue", color= "black") +
  labs(title = "Distribution of Engine Horsepower",
       x = "Horsepower",
       y = "Frequency")
```

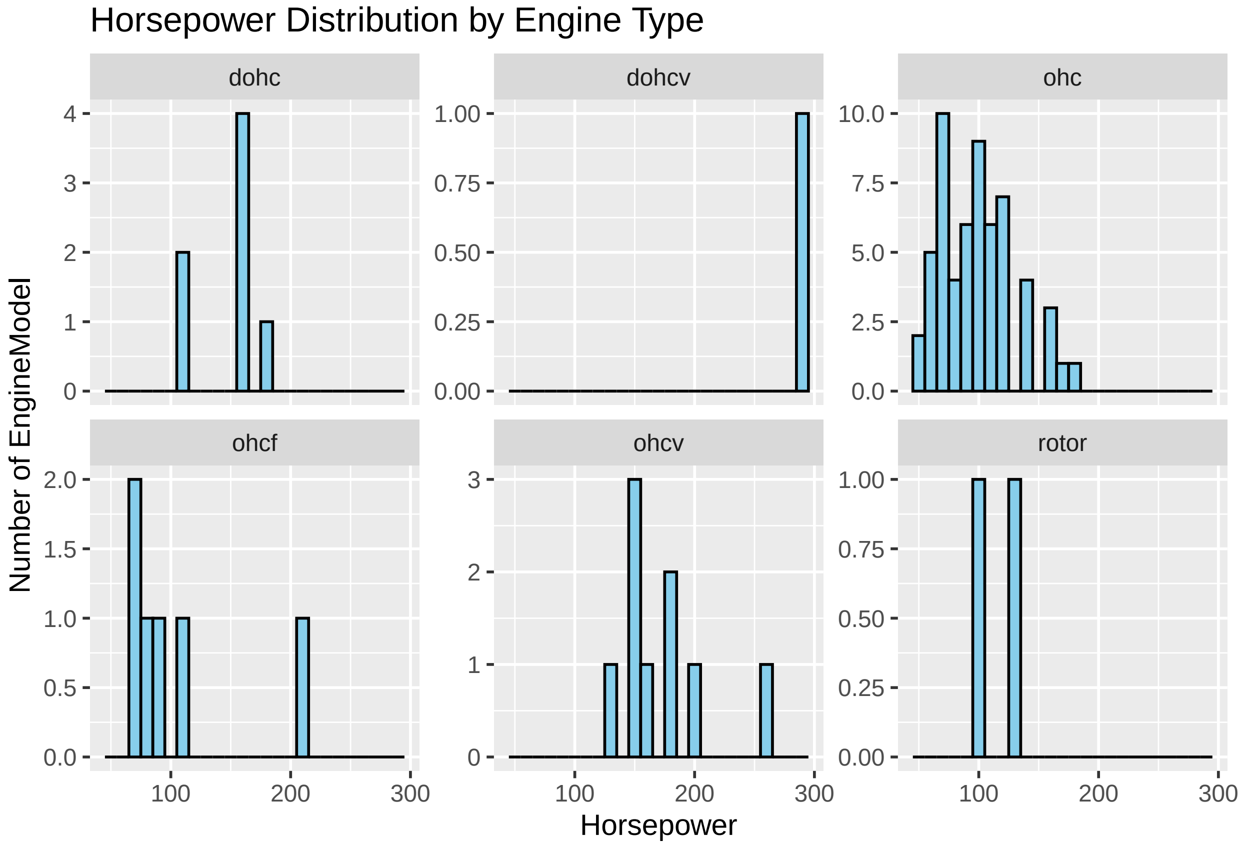


The histogram illustrates the horsepower distribution is skewed to the left. The majority of engines lies from 70 to 120 horsepower range, whereas there are some strong performance of engines with high horsepower on the right side.

Task 2: Horsepower Analysis

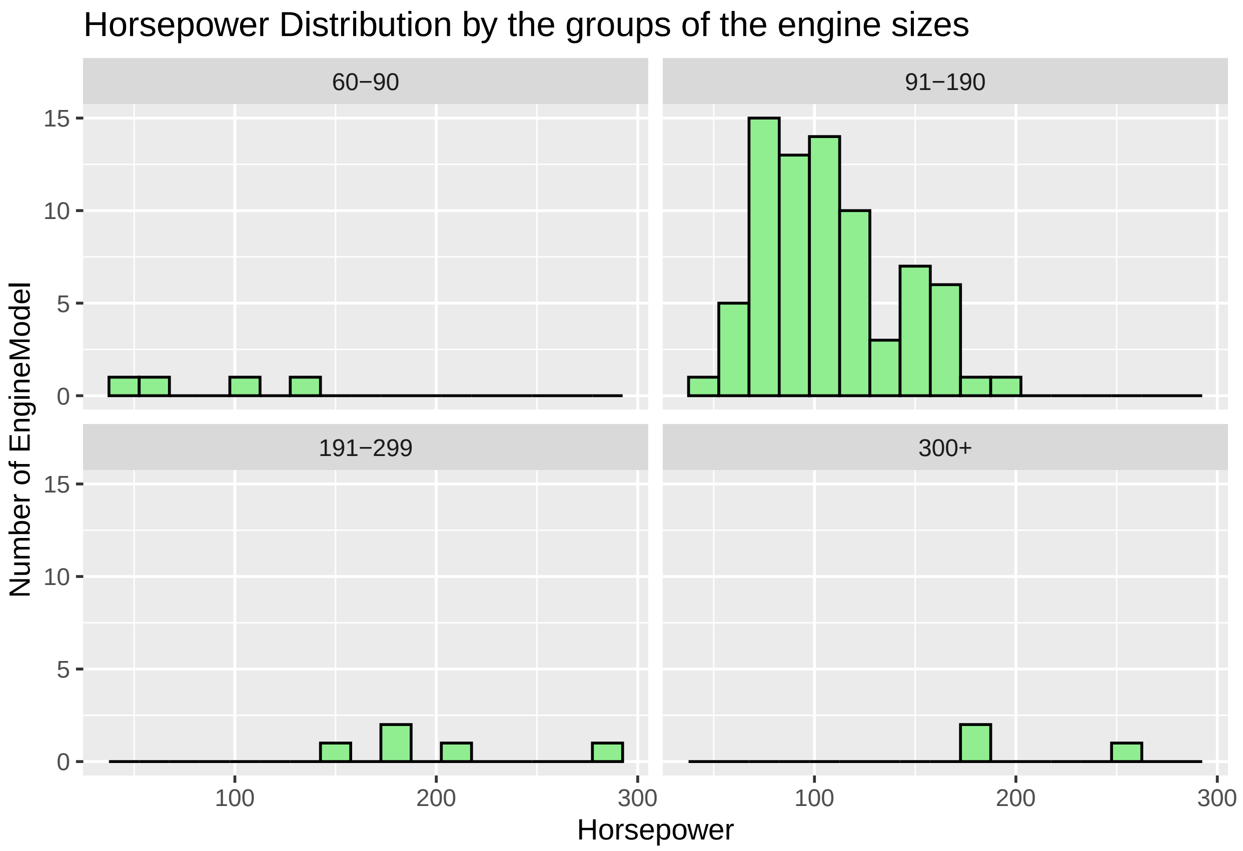
To better understand horsepower, I will analyze distribution by two factors: engine types (EngineTypes) and engine size (EngineSize). Using data has replaced the NA in horsepower by the median of the remaining horsepowers. The engine types (EngineType) had some missing value (NA) so making a clean engines data is important to analyse. Then, drawing chart for two factors will be easier to compare these distributions.

```
# Analysis by Engine Type
# First, we need to clean the NA in the engines data
engines_clean <- engines %>%
  filter(!is.na(EngineType))
ggplot(data = engines_clean, mapping = aes(x = Horsepower)) +
  geom_histogram(binwidth = 10, fill = "skyblue", color = "black") +
  facet_wrap(~ EngineType, scales = "free_y") +
  labs(title = "Horsepower Distribution by Engine Type",
       x = "Horsepower",
       y = "Number of EngineModel")
```



The bar chart shows the horsepower distribution in six types of engine technology. The distribution is different between each type of engines. Especially, the horsepower of ohc lies mainly on the left side. While engine technologies like dohc tend to achieve higher average horsepower than ohc.

```
# Analysis by the groups of the engine sizes
# Create the the groups of the engine sizes column on the engines data
engines_with_groups <- engines %>%
  mutate(
    EngineSizeGroup = cut(EngineSize,
                          breaks = c(60, 90, 190, 299, Inf),
                          labels = c("60-90", "91-190", "191-299", "300+"),
                          right = FALSE)
  )
ggplot(data = engines_with_groups, mapping = aes(x = Horsepower)) +
  geom_histogram(binwidth = 15, fill = "lightgreen", color = "black") +
  facet_wrap(~ EngineSizeGroup, nrow = 2) +
  labs(title = "Horsepower Distribution by the groups of the engine sizes",
       x = "Horsepower",
       y = "Number of EngineModel")
```



The chart shows a clear relationship between horsepower and engine size. The smaller of engine size, the lower the horsepower tends to be and vice versa.

Task 3: Fuel Performance and Trouble Analysis

This task compares fuel efficiency between diesel and petrol cars using a t-test, and explores the influence of a wheel of a motor vehicle to transmit force (DriveWheels) by using a boxplot. Finally, we analyze the frequency of Troubles to find the most common problems.

Do diesel cars have higher average CityMpg than gasoline cars?

```r
# Aggregation of the `engines` table creates a single lookup table for each EngineModel
engines_aggregated <- engines %>%
  group_by(EngineModel) %>%
  summarise(
    EngineType = first(na.omit(EngineType)),
    NumCylinders = first(na.omit(NumCylinders)),
    FuelSystem = first(na.omit(FuelSystem)),
    FuelTypes = first(na.omit(FuelTypes)),
    EngineSize = mean(EngineSize, na.rm = TRUE),
    Horsepower = mean(Horsepower, na.rm = TRUE)
  ) %>%
  ungroup()
# Create summary data table of automobile and engines_aggregated
full_data <- left_join(automobile, engines_aggregated, by = "EngineModel")

# Compare between Diesel and Gasoline
diesel_cars <- full_data %>% filter(FuelTypes == "diesel")
gas_cars <- full_data %>% filter(FuelTypes == "gas")

# Using T_test as a statical evidence
# Hypothesis H0: There is no difference of average CityMPG between 2 types
# Hypothesis H1: There is difference in CityMPG
ttest_result <- t.test(diesel_cars$CityMpg, gas_cars$CityMpg)
print(ttest_result)
```

```
##
##  Welch Two Sample t-test
##
## data:  diesel_cars$CityMpg and gas_cars$CityMpg
## t = 3.6237, df = 23.015, p-value = 0.001424
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  2.412141 8.829163
## sample estimates:
## mean of x mean of y
##  30.30000  24.67935
```

By using t-test, the mean of diesel cars (30.3 miles per gallon) is higher than gasoline cars (24.67935 miles per gallon). This means that diesel cars have statistically significantly better city fuel consumption than gas cars.

How does DriveWheels affect fuel efficiency (CityMpg and HighwayMpg)?

```r
# Calculate mean by using group_by() and summarise()
mpg_summary <- full_data %>%
  group_by(DriveWheels) %>%
  summarise(
    Avg_CityMpg = mean(CityMpg, na.rm = TRUE),
    Avg_HighwayMpg = mean(HighwayMpg, na.rm = TRUE)
  )
print(mpg_summary)
```

```
## # A tibble: 3 x 3
##   DriveWheels Avg_CityMpg Avg_HighwayMpg
##   <chr>             <dbl>          <dbl>
## 1 4wd                23.1           27.2
## 2 fwd                28.3           34.2
## 3 rwd                20.5           25.6
```

From the result table, the front-wheel drive (fwd) has the highest miles per gallon in both the City and Highway. Therefore, the front-wheel drive (fwd) save more fuel than the others. While the rear-wheel drive (rwd) is consumed more fuel than front-wheel drive (fwd), especially in city.

What are the top 5 most common troubles related to the engines?

```r
# Create summary data table of full_data and maintenance
full_trouble <- left_join(maintenance, full_data, by = "PlateNumber")
# Filter the trouble cars
trouble_vehicles <- full_trouble %>%
  filter(as.character(ErrorCodes) != "0") #Because ErrorCodes is factor, convert it to character
top_5_troubles <- trouble_vehicles %>%
  count(Troubles, sort = TRUE) %>%
  head(5)
print(top_5_troubles)
```

```
##             Troubles  n
## 1          Cylinders 38
## 2            Chassis 25
## 3 Ignition (finding) 22
## 4    Noise (finding) 19
## 5         Worn tires 16
```

After filtering the records of vehicle troubles, I conducted a frequency analysis to determine the most common problems. The results showed that the top five problems related to the engine and other components were Cylinders with the highest frequency (38), Chassis (25), Ignition (22), Noise (19), Worn tires (16).

Do the troubles differ between engine types?

```r
# Create the data set troubles with each engine type and filer NA for sure
most_common_trouble_per_engine_type <- trouble_vehicles %>%
  filter(!is.na(EngineType) & !is.na(Troubles)) %>%
  group_by(EngineType) %>% # Group by engine type
  count(Troubles, name = "count_per_trouble") %>%
  filter(count_per_trouble == max(count_per_trouble)) %>% #Keep rows with counts equal to the largest c
  arrange(EngineType) %>%
  ungroup()
# Print the results
print(most_common_trouble_per_engine_type,n= Inf)
```

```
## # A tibble: 26 x 3
##    EngineType Troubles            count_per_trouble
##    <chr>      <chr>                           <int>
##  1 dohc       Chassis                             3
##  2 dohc       Suspected clutch                    3
##  3 dohcv      Painting                            2
##  4 dohcv      Real axe                            2
##  5 dohcv      Side slip                           2
##  6 dohcv      Valve clearance                     2
##  7 ohc        Cylinders                          29
##  8 ohcf       Cylinders                           5
##  9 ohcv       Air conditioner                     1
## 10 ohcv       Brake fluild                        1
## 11 ohcv       Chassis                             1
## 12 ohcv       Cylinders                           1
## 13 ohcv       Fans                                1
## 14 ohcv       Ignition (finding)                  1
## 15 ohcv       Noise (finding)                     1
## 16 ohcv       Painting                            1
## 17 ohcv       Pedals                              1
## 18 ohcv       Real axe                            1
## 19 ohcv       Side slip                           1
## 20 ohcv       Suspected battery                   1
## 21 ohcv       Temperature sensors                 1
## 22 ohcv       Valve clearance                     1
## 23 rotor      Crank shaft                         1
## 24 rotor      Front axe                           1
## 25 rotor      Gear box (finding)                  1
## 26 rotor      Oil filter                          1
```

The troubles are different between each engine type. For example, the ohc engines such as ohc and ohcf faced a specific and common trouble which is Cylinders. On the other hand, high performance dohc engines often have problems with Chassis and Suspected clutch. Other engines such as dohcv and rotor do not have a single problem that dominates, but have many different problems that occur together.

Task 4: Errorcodes and Maintenance methods analysis

First step, I analyse the frequency of error codes. I choose two factors: BodyStyles and FuelTypes. I analyse the factors that can influence the maintenance method applied and use the 100% stacked bar chart to compare proportions between groups.

Which error type (ErrorCodes) occurs most frequently?

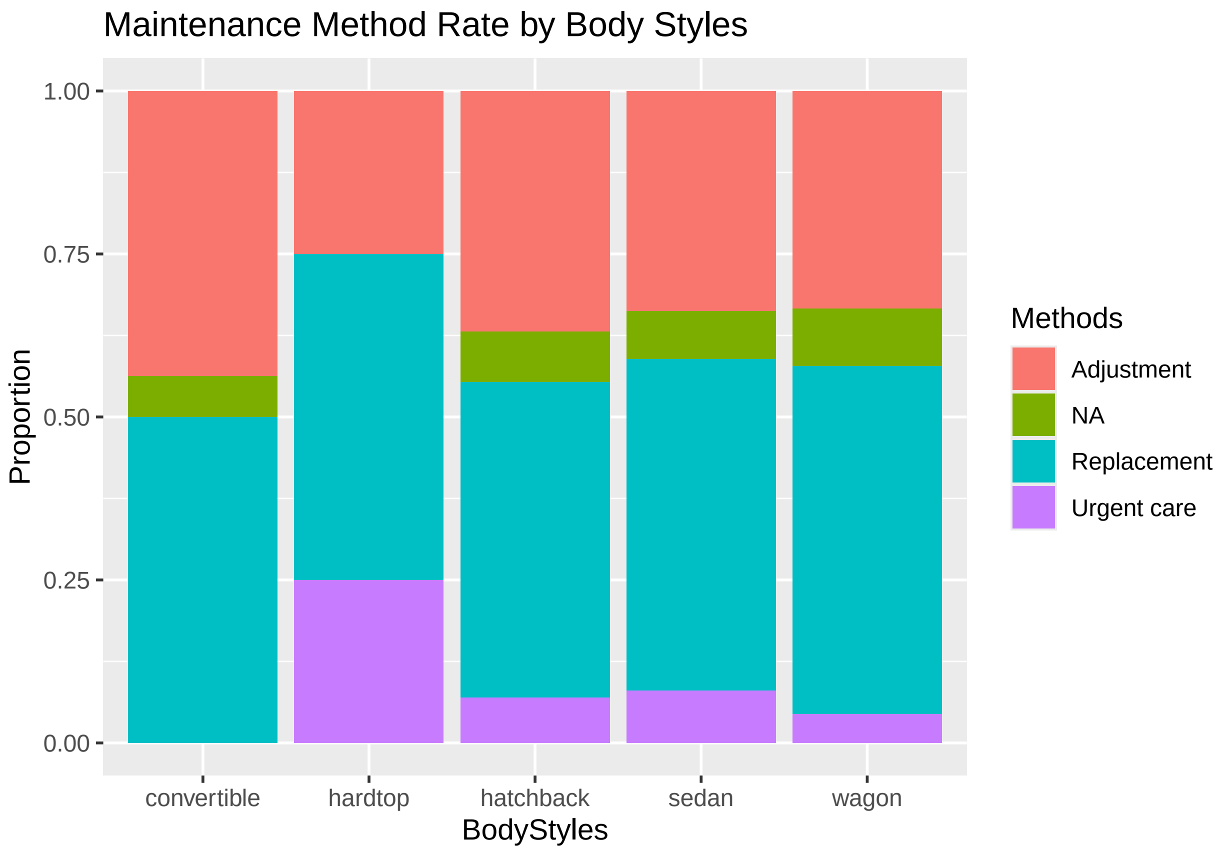
```r
# Create the data set of error codes
error_code_frequency <- full_trouble %>%
  count(ErrorCodes, sort = TRUE) # Count and sort
# Print the results
print(error_code_frequency)
```

```
##   ErrorCodes   n
## 1          1 182
## 2          -1 164
## 3           0  28
```

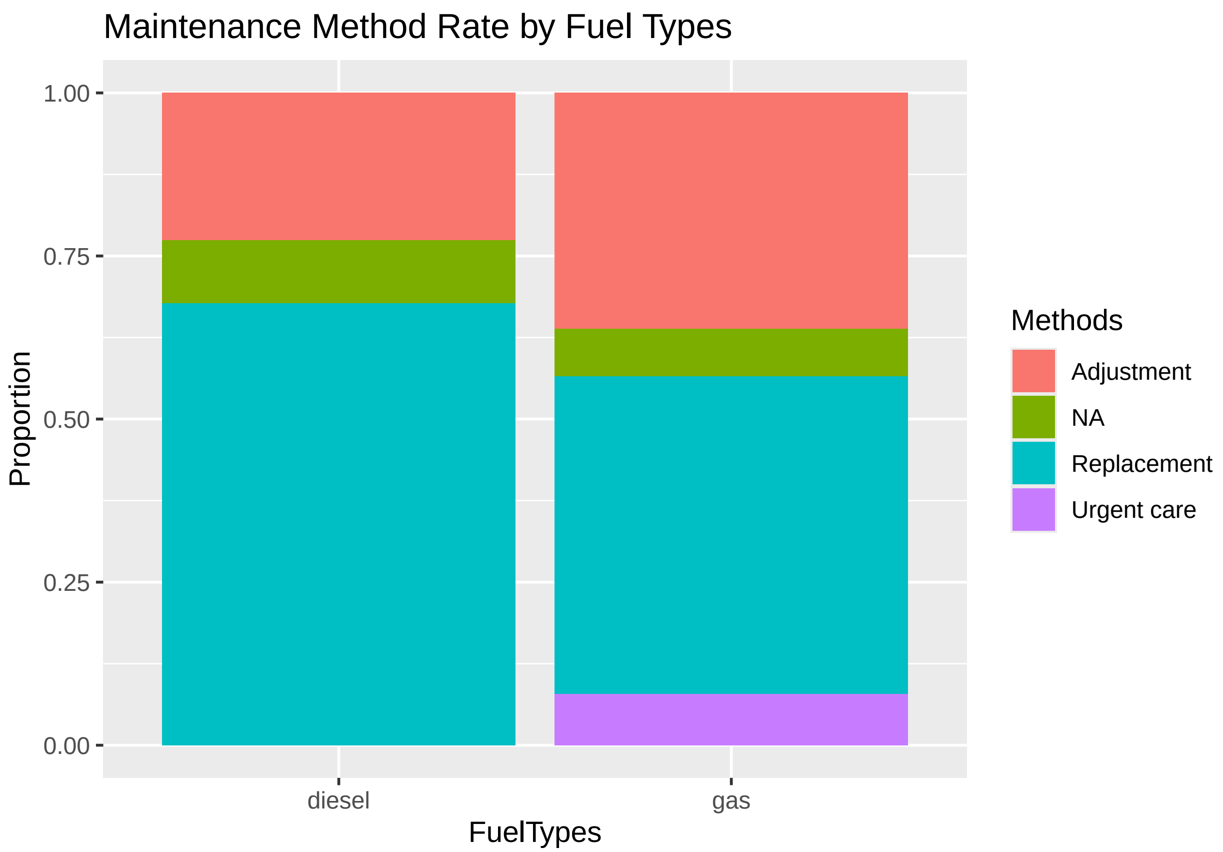
The most frequency of error codes is the engine fails (1), while other vehicle component fails (-1) is lower than engine fails (1) by 18 cases.

Analysis of factors affecting maintenance methods

```
# Factor 1: BodyStyles
ggplot(data = full_trouble, mapping = aes(x = BodyStyles, fill = Methods)) + geom_bar(position = "fill"
  labs(title = "Maintenance Method Rate by Body Styles",
       x = "BodyStyles",
       y = "Proportion")
```



```
# Factor 2: FuelTypes
ggplot(data = full_trouble, mapping = aes(x = FuelTypes, fill = Methods)) + geom_bar(position = "fill")
  labs(title = "Maintenance Method Rate by Fuel Types",
       x = "FuelTypes",
       y = "Proportion")
```



In the maintenance methods by body styles, there is a higher rate of urgent care in “hardtop” cars than the other, whereas “convertible” cars has a highest percentage of Adjustment method. Beside, the chart in fuel types shows a certain trend, with gasoline vehicles having a lower Replacement rate than diesel vehicles. Especially, the Urgent care methods appears only in gasoline car.