WESTERN SYDNEY
UNIVERSITY

SCHOOL OF COMPUTER, DATA & MATH SCIENCES

ASSIGNMENT COVER SHEET

STUDENT DETAILS

Student name:	Le Hong Thanh Tu	Student ID number:	22166855

UNIT AND TUTORIAL DETAILS

Unit name:	Analytics Programming	Unit number:	AP-T325WSD-1
Tutorial group:		Tutorial day and time:	Sunday, 15h30-18h45
Lecturer or Tutor name:	Assoc. Prof. Nguyen Tan Luy		

ASSIGNMENT DETAILS

Title:	Assignment 2				
Length:		Due date:	21/11/2025	Date submitted:	21/11/2025
Home campus (where you are enrolled):	Vietnam				

DECLARATION

- ☑ I hold a copy of this assignment if the original is lost or damaged.
- ☑ I hereby certify that no part of this assignment or product has been copied from any other student's work or from any other source except where due acknowledgement is made in the assignment.
- ☑ I hereby certify that no part of this assignment or product has been submitted by me in another (previous or current) assessment, except where appropriately referenced, and with prior permission from the Lecturer / Tutor / Unit Coordinator for this unit.
- ☑ No part of the assignment/product has been written/produced for me by any other person except where collaboration has been authorised by the Lecturer / Tutor /Unit Coordinator concerned.
- ☑ I am aware that this work will 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).**

Student's signature: Le Hong Thanh Tu

Note: An examiner or lecturer / tutor has the right to not mark this assignment if the above declaration has not been signed.

Github link: https://github.com/22002418-thanhtu/22166855_Le-Hong-Thanh-Tu_AP1_Ass2

Project Trimester 2 2025

Le Hong Thanh Tu

November 21st, 2025

Vehicle Data Analysis Report

Part 1

1.a. Write the code to inspect the data structure and present the data: The missing values in the dataset were written as '?', replace any '?' with 'NA'.

Rationale

Method to replace the missing values, written as "?" in the data frames: First, the missing value should be identified using <engines == "?">, then replace those identified values with the 'NA' value.

The R code is shown below:

```
engines[engines == "?"] <- NA
automobile[automobile == "?"] <- NA
maintenance[maintenance == "?"] <- NA
```

Testing the R code

To test if the code replaces all '?' with 'NA', the testing code was run.

Firstly, it is essential to find the total remaining value of '?'.

Secondly, an 'if' statement was run to show the result. If there is no remaining 'NA' after replacement, it shows that all '?' characters are replaced with 'NA'. If not, it shows how many '?' value was left.

```
remain_NA <- sum(engines == "?", na.rm = TRUE)
           +sum(automobile == "?", na.rm = TRUE)
```

```
## [1] 0
```

```
           +sum(maintenance == "?", na.rm = TRUE)
```

```
## [1] 0
```

```
   if(remain_NA == 0) {
   print("All '?' characters are replaced with NA.")
 } else {
   print(paste("There are", remain_NA , "'?' characters still remain!"))
 }
```

```
## [1] "All '?' characters are replaced with NA."
```

Output Results: The top ten rows of the three data frames after the replacements are presented below

```
# Load libraries for table formatting
library(knitr)
library(kableExtra)
# 1. Display Engines Data
kable(head(engines, 10), caption = "Top 10 Rows: Engines Data")
```

Table 1: Top 10 Rows: Engines Data

EngineModel	EngineType	NumCylinders	EngineSize	FuelSystem	Horsepower	FuelTypes	Aspiration
E-0001	dohc	four	130	mpfi	111	gas	std
E-0002	ohcv	six	152	mpfi	154	gas	std
E-0003	ohc	four	109	mpfi	102	gas	std
E-0004	ohc	five	136	mpfi	115	gas	std
E-0005	ohc	five	136	mpfi	110	gas	std
E-0006	ohc	five	131	mpfi	140	gas	turbo
E-0007	ohc	five	131	mpfi	160	gas	turbo
E-0008	ohc	four	108	mpfi	101	gas	std
E-0009	ohc	six	164	mpfi	121	gas	std
E-0010	ohc	six	164	mpfi	121	gas	std

```
# 2. Display Automobile Data
kable(head(automobile, 10), caption = "Top 10 Rows: Automobile Data")%>%
  kable_styling(latex_options = c("scale_down", "hold_position"))
```

Table 2: Top 10 Rows: Automobile Data

PlateNumber	Manufactures	BodyStyles	DriveWheels	EngineLocation	WheelBase	Length	Width	Height	C
53N-001	Alfa-romero	convertible	rwd	front	88.6	168.8	64.1	48.8	
53N-002	Alfa-romero	hatchback	rwd	front	94.5	171.2	65.5	52.4	
53N-003	Audi	sedan	fwd	front	99.8	176.6	66.2	54.3	
53N-004	Audi	sedan	4wd	front	99.4	176.6	66.4	54.3	
53N-005	Audi	sedan	fwd	front	99.8	177.3	66.3	53.1	
53N-006	Audi	sedan	fwd	front	105.8	192.7	71.4	55.7	
53N-007	Audi	wagon	fwd	front	105.8	192.7	71.4	55.7	
53N-008	Audi	sedan	fwd	front	105.8	192.7	71.4	55.9	
53S-001	Audi	hatchback	4wd	front	99.5	178.2	67.9	52.0	
53S-002	Bmw	sedan	rwd	front	101.2	176.8	64.8	54.3	

```
# 3. Display Maintenance Data
kable(head(maintenance, 10), caption = "Top 10 Rows: Maintenance Data")
```

Table 3: Top 10 Rows: Maintenance Data

ID	PlateNumber	Date	Troubles	ErrorCodes	Price	Methods
1	53N-001	15/02/2024	Break system	-1	110	Replacement
2	53N-001	16/03/2024	Transmission	-1	175	Replacement
3	53N-001	15/04/2024	Suspected clutch	-1	175	Adjustment
4	53N-001	15/05/2024	Ignition (finding)	1	180	Adjustment
5	53N-001	14/06/2024	Chassis	-1	85	Replacement
6	53N-002	15/02/2024	Cylinders	1	1000	Replacement
7	53N-002	16/03/2024	Ignition (finding)	1	180	Adjustment
8	53N-002	15/04/2024	No error	0	0	NA
9	53N-003	16/04/2024	Loss of driving ability	-1	180	Urgent care
10	53N-004	17/04/2024	Loss of driving ability	-1	180	Urgent care

1.b. Write code to check: after replacing, how many rows were affected in total?

Rationale Calculating the number of rows that were affected after the replacement meant finding the number of rows that contain at least one 'NA' value. Steps to calculate are shown below:

- Step 1: Calculate the number of rows with NA for each data frame.
  - The code 'is.na()' was used to identify the 'NA' value in each data frame.
  - The code 'rowSums(is.na())' was used to find the total number of 'NA' values in each row.
  - The code 'sum(rowSums(is.na()))' was used to find the total number of 'NA' values in each data frame.
- Step 2: finding the total number of rows with 'NA' across all three data frames.
- Step 3: Presenting the number of 'NA' values for each data frame, and across three data frames.

The R code is shown below:

```
#Step 1. Calculate rows with NA for each data frame
  # Finding the number of <NA> in engines
  na_rows_engines <- sum(rowSums(is.na(engines)))
  # Finding the number of <NA> in automobile
  na_rows_automobile <- sum(rowSums(is.na(automobile)))
  # Finding the number of <NA> in maintenance
  na_rows_maintenance <- sum(rowSums(is.na(maintenance)))

#Step 2. The total number of rows with NA across all three data frames
  total_na_rows <- na_rows_engines + na_rows_automobile + na_rows_maintenance

#step 3. Present the result
  cat("engines has", na_rows_engines, "rows with at least one NA.\n")
  cat("automobile has", na_rows_automobile, "rows with at least one NA.\n")
  cat("maintenance has", na_rows_maintenance, "rows with at least one NA.\n")
  cat("Total rows with at least one NA across the three data frames:",
      total_na_rows,".\n")
  cat("There are",total_na_rows,"were affected in total after replacement.\n")
```

Output Results:

```
## engines has 6 rows with at least one NA.

## automobile has 0 rows with at least one NA.

## maintenance has 28 rows with at least one NA.

## Total rows with at least one NA across the three data frames: 34 .

## There are 34 were affected in total after replacement.
```

1.c. Does this change alter the data distribution?

To identify whether there are any changes after replacing '?' with 'NA', I will compare the before and after data using the 'identical()' to test If the two versions are exactly equal. The answer follows the steps below:

- Step 1: Input data before change. The data before the change is input again with another name
- Step 2: To analyse if the change alters the data distribution, the The column 'Horsepower' of the engines data frame was chosen to explain, as this variable contains missing values.
- Step 3: Convert the value in the original 'Horsepower' to a numeric value, and remove all '?' values.
- Step 4: Convert the value in the changed 'Horsepower' to a numeric, and remove all 'NA' values.
- Step 5: Apply 'if' statement and 'identical()' to have the result. The R code is shown below:

The R code is shown below:

```r
#Step 1: Input data before change
  engines_bef<-read.csv("Engine.csv")
  automobile_bef<-read.csv("Automobile.csv")
  maintenance_bef<-read.csv("Maintenance.csv")

#Step 2: Select 'Horsepower' as a specific column to analyze
  after_value <- engines$Horsepower
  bef_value <- engines_bef$Horsepower

#Step 3: Prepare the "Before" data for comparison
  #Taking the raw column, remove "?", and convert the rest to numbers


  bef_value <- engines_bef$Horsepower
  bef_value <- bef_value[bef_value != "?"] # Remove rows with "?"
  bef_value <- as.numeric(bef_value) # Convert text to number

  #Step 4: Prepare the "After" data for comparison
  #Taking the after changed column, remove <NA>, and convert the rest to numbers
  after_value <- engines$Horsepower
  after_value <- after_value[!is.na(after_value)] # Remove rows with <NA>
  after_value <- as.numeric(after_value) # Convert text to number

  #Step 5: Compare the Valid Data
  if(identical(after_value, bef_value)) {
```

```
    print("1. CONCLUSION: The data distribution remains UNCHANGED after replacing '?' with NA.\n")
      } else {
    print("1. CONCLUSION: The data distribution has CHANGED after replacing '?' with NA.\n")
      }
```

Output Results:

```
## [1] "1. CONCLUSION: The data distribution remains UNCHANGED after replacing '?' with NA.\n"
```

1.d. Convert categorical variables BodyStyles, FuelTypes, ErrorCodes to factors:

Rationale Using the code 'as.factor()' to transfer values in BodyStyles, FuelTypes, and ErrorCodes are factors.

The R code is shown below:

```
  automobile$BodyStyles <- as.factor(automobile$BodyStyles)
  engines$FuelTypes <- as.factor(engines$FuelTypes)
  maintenance$ErrorCodes <- as.factor(maintenance$ErrorCodes)
```

Code testing

The 'is.factor()' was utilized to determine if the required column is a factor. Then the 'if' statement was used to show the result.

```
#Code testing
  if(is.factor(automobile$BodyStyles)
     & is.factor(engines$FuelTypes)
     & is.factor(maintenance$ErrorCodes)) {
  # the condition to test if all columns are formatted as factors
    print("All variables are converted to Factors.")
  } else {
    print("One or more variables are not factors.")
  }
```

```
## [1] "One or more variables are not factors."
```

1.e.Replace the missing values in column Horsepower with the median horsepower

Rationale

To replace the 'NA' value in column Horsepower with its median, I followed the following steps:

- Step 1: Finding the Median of Horsepower, using median()
- Step 2: Identify the 'NA' value using the code 'is.na()', then replacing them with Hosrepower median

The R code is shown below:

```
#Step 1. Calculate the Median of Horsepower
  median_hp <- median(engines$Horsepower, na.rm = TRUE)

#Step 2. Replace <NA> with the Median
  engines$Horsepower[is.na(engines$Horsepower)] <- median_hp
```

Code testing

To test if any 'NA' value was not replaced, I count the number of 'NA' values in Horsepower after replacement. Then, the 'if' statement is used to show the result.

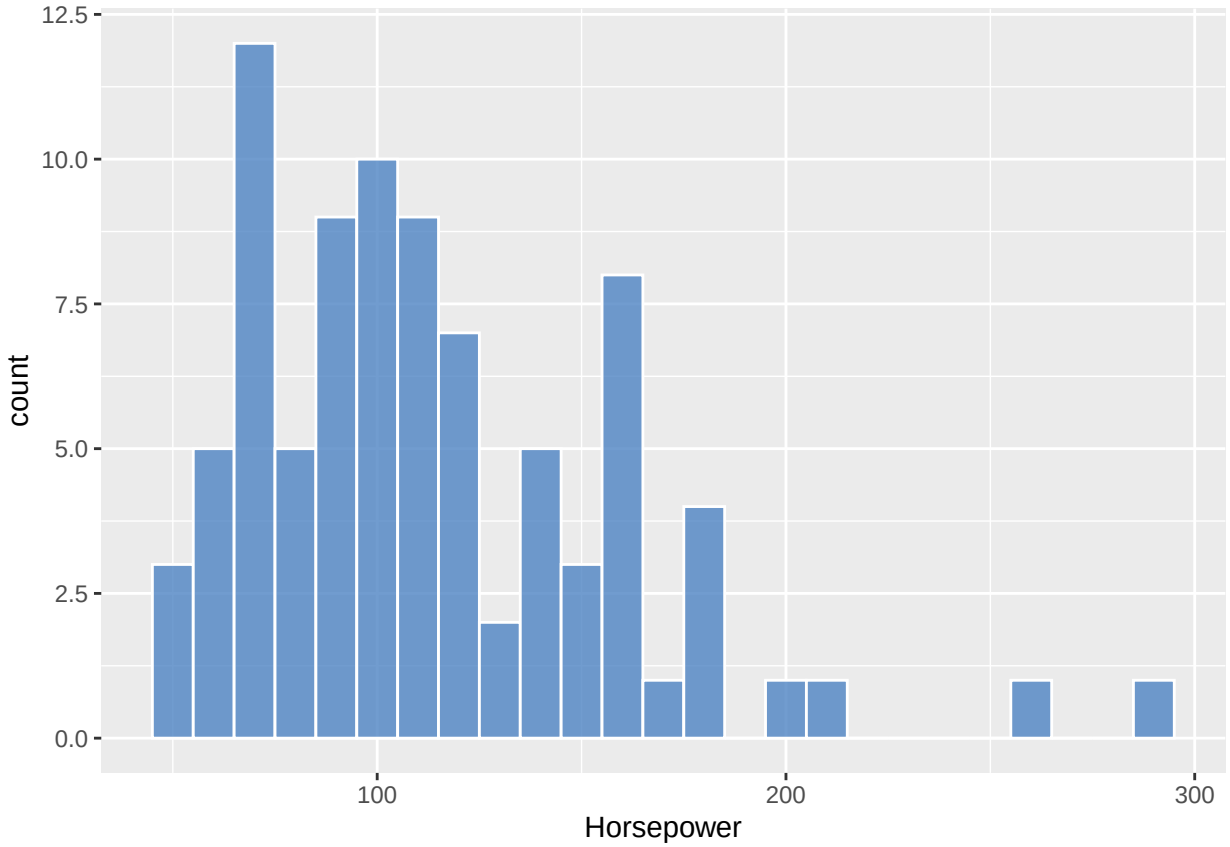
```
# calculating the 'NA' that did not replace
remaining_nas <- sum(is.na(engines$Horsepower))
  if(remaining_nas == 0) {
  print(paste("All missing values are replaced. Median used:", median_hp))
} else {
  print(paste("There are", remaining_nas, "missing values still remain!"))
}
```

```
## [1] "There are 1 missing values still remain!"
```

1.f. Select the appropriate chart type and display: horsepower distribution

The histogram was chosen to display the Horsepower after replacing the 'NA' value with its median.

```
library(ggplot2)
engines$Horsepower <- as.numeric(engines$Horsepower)
ggplot(engines, aes(x = Horsepower)) +
# The bin width is set to be 10.
# The fill = "#4E84C4" is the code to identify the color for the bin in the graph
# The alpha = 0.8 is the transparency of the graph, making the color lighter
geom_histogram(binwidth = 10, fill = "#4E84C4", color = "white", alpha = 0.8)
```



Part 2

2.a. Horsepower analysis on different engine types

Rationale

To investigate the distribution of the horsepower across the engine types, I follow the steps below:

- Step 1: Replace the 'NA' value in EngineTypes with 'Other Types'
- Step 2: Group Horsepower by engine type using 'group_by()'
- Step 3: Find the average, standard deviation, maximum, and minimum horsepower of each engine type.
- Step 4: Present the calculated statistic
- Step 5: Visualise Horsepower across Engine Types using the histogram

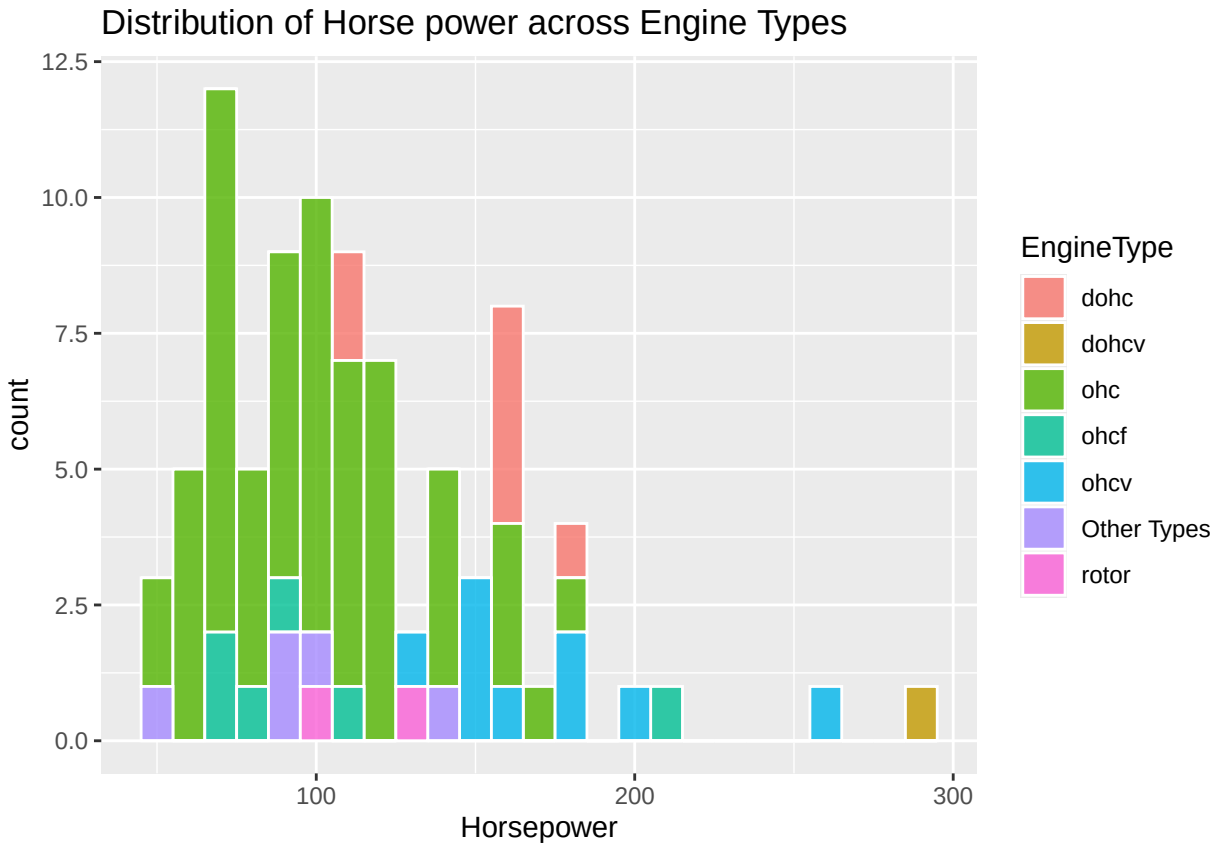
The R code is shown below:

```
 # Load the library for the 'summarise()'
 library(dplyr)
# Step 1: Replace the 'NA' value in EngineTypes with 'Other Types'
  engines$EngineType[is.na(engines$EngineType)]<-"Other Types"
# Step 2: Grouping the data
  Horsepower_EngTypes <- engines %>%
  group_by(EngineType) %>%
#step 3: Find the statistical values of horsepower for each engine type
  summarise(
    Count = n(), #finding the number of observations (Horsepower) by Engine Types
    Mean_HP = mean(Horsepower, na.rm = TRUE), #finding the average Horsepower by Engine Types
    SD_HP = sd(Horsepower, na.rm = TRUE), #finding the standard deviation of Horsepower by Engine Types
    Min_HP = min(Horsepower, na.rm = TRUE), #finding the minimum value of Horsepower by Engine Types
    Max_HP = max(Horsepower, na.rm = TRUE) #finding the maximum value Horsepower by Engine Types
    )
#Step 4: Present the calculated statistic
  # Load libraries for table formatting
  library(knitr)
  library(kableExtra)
    kable(Horsepower_EngTypes, caption = "Statistical Summary of Horsepower by Engine Type")
#Step 5: Visualise Horsepower across Engine Types using the histogram
  ggplot(engines, aes(x = Horsepower, fill=EngineType)) +
    geom_histogram(binwidth = 10, color = "white", alpha = 0.8)+
    labs( title = "Distribution of Horse power across Engine Types")
  # The bin width is set to be 10.
  # The color = "white" is the code to set the color of the line between bins
  # The alpha = 0.8 is the transparency of the graph, making the color lighter
```

Output Results:

Table 4: Statistical Summary of Horsepower by Engine Type

EngineType	Count	Mean_HP	SD_HP	Min_HP	Max_HP
Other Types	5	95.40000	33.24605	48	142
dohc	7	147.42857	25.45491	111	176
dohcv	1	288.00000	NA	288	288
ohc	58	99.59649	32.29046	52	182
ohcf	6	106.00000	51.77644	69	207
ohcv	9	176.11111	38.11314	134	262
rotor	2	118.00000	24.04163	101	135



Explain the findings

There are seven Engine Types. The 'Other Types' value is used to represent for missing value. From the result, the 'ohc' engine is the most popular engine type. Its average horsepower is 99.6, which is the second weakest engine after Other engine types have with average horsepower of 95.4. The engine type with the strongest horsepower is 'dohcv', with an average horsepower of 288. Because there is not much information to investigate for one observation, the mean value of 'dohcv' brings limited meaning. The second strong engine type is 'ohcv', with the average horsepower of 176.1. This engine type has nine observations, meaning that there is more insight to discover for the strength of 'ohcv' engine types.

2.b. Analyse Horsepower Across Engine Size Groups

Rationale

To investigate the distribution of the horsepower across the engine size, I follow the steps below:

- Step 1: Create the Groups or Bins for EngineSize by using the code 'cut()' to divide the engine size into 5 bins: 0-60, 61-90, 91-190, 191-299, and 300+
- Step 2: Group Horsepower by engine size using 'group_by()'
- Step 3: Find the average, standard deviation, maximum, and minimum horsepower of each engine size
- Step 4: Present the calculated statistic
- Step 5: Visualize Horsepower across Engine Size using the histogram

The R code is shown below:

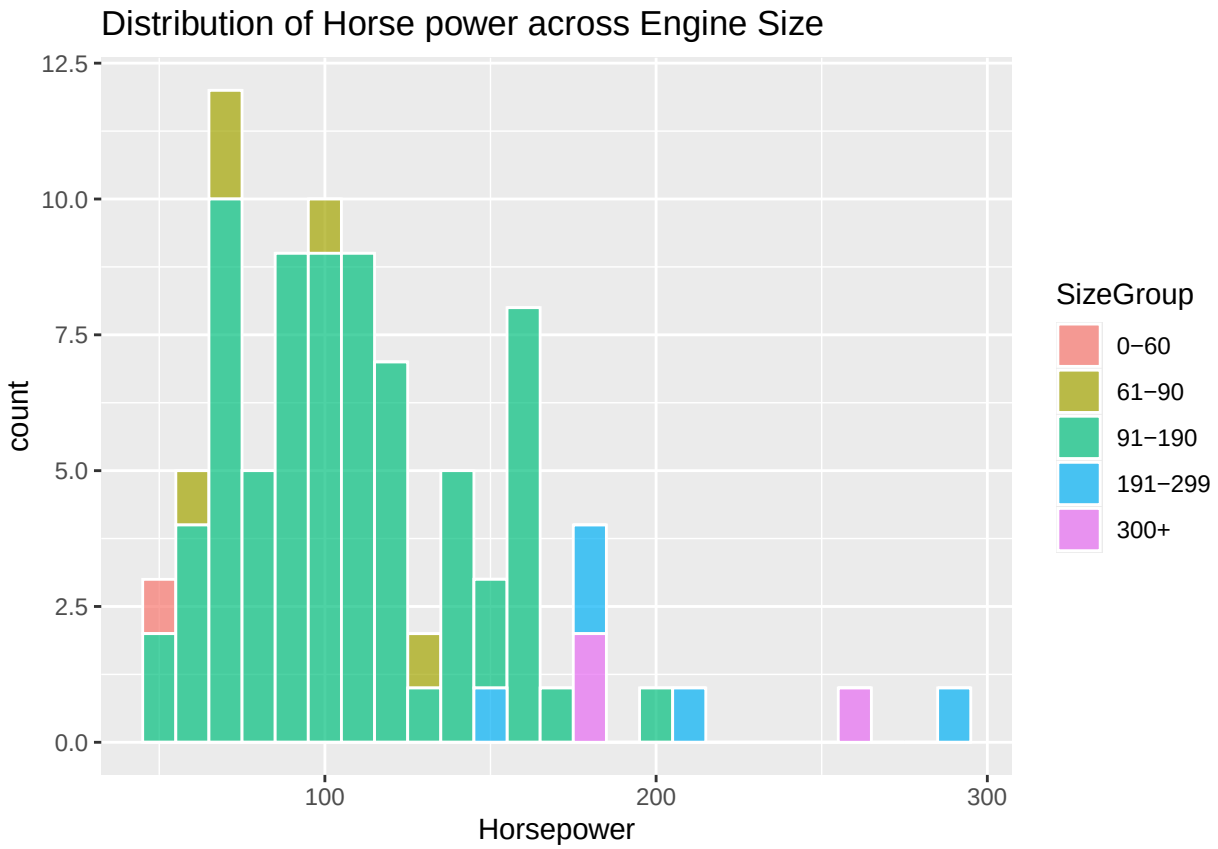
```
#Load the library for the 'summarise()'
library(dplyr)
  #Step 1: Create the Groups (Bins for EngineSize)
  engines$SizeGroup <- cut(engines$EngineSize,
                           breaks = c(0, 60, 90, 190, 299, Inf),
                           labels = c("0-60","61-90", "91-190", "191-299", "300+"),
                           right = TRUE)

  #Step 2: Group Horsepower by EngSize
  Horsepower_EngSize <- engines %>%
    group_by(SizeGroup) %>%
  #Step 3: Find the statistical values of horsepower for each engine size
    summarise(
      Count = n(),
      Mean_HP = mean(Horsepower, na.rm = TRUE),
      SD_HP = sd(Horsepower, na.rm = TRUE),
      Min_HP = min(Horsepower, na.rm = TRUE),
      Max_HP = max(Horsepower, na.rm = TRUE)
    )
#Step 4: Present the calculated statistic
  # Load libraries for table formatting
  library(knitr)
  library(kableExtra)
    kable(Horsepower_EngSize, caption = "Statistical Summary of Horsepower by Engine Size")
#Step 5: Visualise Horsepower across Engine Size using the histogram
  ggplot(engines, aes(x = Horsepower, fill=SizeGroup)) +
    geom_histogram(binwidth=10, color = "white", alpha = 0.7)+
    labs( title = "Distribution of Horse power across Engine Size")
```

Output Results:

Table 5: Statistical Summary of Horsepower by Engine Size

SizeGroup	Count	Mean_HP	SD_HP	Min_HP	Max_HP
0-60	1	48.0	NA	48	48
61-90	5	86.8	31.14001	60	135
91-190	74	107.0	33.91206	52	200
191-299	5	201.6	51.73297	155	288
300+	3	210.0	45.03332	184	262



Explain the findings

The results show that there is a positive relationship between engine size and horsepower. The size from 0 to 60 has the lowest average horse power of 48. The average horsepower of groups 61-90,91-190, and 191-299 are 86.8, 107, and 201.6, respectively. The strongest engine size is the largest size group (300+), with an average horsepower of 210.

Part 3

3.a. Do Diesel Cars Have Higher City MPG?

Rationale

The FuelTypes contain values about diesel and gasoline cars in the engine data frame. The CityMpg contains information about City Mpg in an automobile data frame. Because the two variables needed are located in different

data frames, it is necessary to merge these two datasets into one.

To determine if Diesel Cars Have Higher City MPG, I follow the steps below steps:

- Step 1: Merge the two data frames using 'merge()' into the new data frame frame called 'merged_data'
- Step 2: Categorize the CityMpg by Fuel Types using 'group_by'
- Step 3: Finding the mean of City Mpg for each Fuel type car to Compare between City MPG values for each type.

- Step 4: Present the result from step 3
- Step 5: Conclusion

The R code is shown below:

```
#Step 1: Merge the datasets
merged_data <- merge(automobile, engines, by = "EngineModel")
merged_data$CityMpg <- as.numeric(as.character(merged_data$CityMpg)) #Convert CityMpg to numeric
merged_data$FuelTypes <- as.factor(merged_data$FuelTypes) #Convert FuelTypes to factor
#Step 2: Group CityMpg by Fuel Types
mpg_fuel <- merged_data %>%
  group_by(FuelTypes) %>%
#Step 3: Calculating the mean of City MPG for each Fuel type car
  summarise(
    Count = n(),
    Average_CityMpg = mean(CityMpg, na.rm = TRUE),
  )
#Step 4: Present the result
# Load libraries for table formatting
library(knitr)
library(kableExtra)
  kable(mpg_fuel, caption = "Average City Mpg by Fuel Type Car")
#Step 5: Conclusion
#identify the mean City Mpg of diesel cars
CityMpg_diesel<- mpg_fuel$Average_CityMpg[mpg_fuel$FuelTypes == "diesel"]
# identify the mean City Mpg of gasoline cars
CityMpg_gas<- mpg_fuel$Average_CityMpg[mpg_fuel$FuelTypes == "gas"]
if(CityMpg_diesel>CityMpg_gas){
  print("Conclusion: The diesel cars have HIGHER average CityMpg the than gasoline cars")
  }else{
  print("Conclusion: The diesel cars have LOWER average CityMpg the than gasoline cars")
  }
```

Output Results:

Table 6: Average City Mpg by Fuel Type Car

FuelTypes	Count	Average_CityMpg
diesel	20	30.30000
gas	201	24.27861

```
## [1] "Conclusion: The diesel cars have HIGHER average CityMpg the than gasoline cars"
```

3.b. How does DriveWheels affect fuel efficiency (City Mpg and Highway Mpg)?

Rationale

To determine the effect of drive wheels on fuel efficiency , I follow the steps below:

- Step 1: Categorize the City MPG and Highway MPG by drive wheel using 'group_by'

- Step 2: To quantify the effect, I calculate the mean of City Mpg and Highway MPG for each drive wheel type
- Step 3: Present the result

The R code is shown below:

```
#Step 1: Categorize the City Mpg and Highway Mpg by drive wheel
  drive_stats <- automobile %>%
    group_by(DriveWheels) %>%
#Step 2: calculate the mean of City Mpg and Highway Mpg for each drive wheel
    summarise(
      Count = n(),
      Avg_City_Mpg = mean(CityMpg, na.rm = TRUE), # calculate the mean of City Mpg
      Avg_Hwy_Mpg  = mean(HighwayMpg, na.rm = TRUE), # calculate the mean of Highway Mpg
      )
#Step 3: Present the result
  # Load libraries for table formatting
  library(knitr)
  library(kableExtra)
    kable(drive_stats, caption = "Effect of Drive wheels on Fuel efficiency")
```

Output Results:

Table 7: Effects of Drive wheels on Fuel efficiency

DriveWheels	Count	Avg_City_Mpg	Avg_Hwy_Mpg
4wd	9	23.11111	27.22222
fwd	120	28.32500	34.23333
rwd	75	20.53333	25.64000

Explain the findings

From the above result, the front-wheel drive ('fwd') is the most fuel-efficient system, with the highest City and Highway MPG, 28.3 and 34.2, respectively. The least fuel-efficient system is rear-wheel drive ('rwd'), with the average City Mpg is 20.5 and the average Highway Mpg is 25.6. Moreover, the value of MPG on the highway is always higher than that in the city, which means driving on the highway is more fuel-efficient than in the city.

3.c. Filter out those engines in the dataset that have trouble or are suspected of having trouble

Rationale

To filter out those engines in the dataset that have trouble or are suspected of having trouble, I will investigate what engine models have trouble or are suspected of having trouble.

I follow the steps below:

- Step 1:
  - I assign the 'No error' value to 'safe', because the value 'No error' in the column 'Trouble' of the 'automobile' data frame means that the car has no trouble, different from a car with Trouble or suspected of having trouble

  - Then, I remove the value 'No error' and the missing value from the Troubles column by using 'filter()', creating a new data frame, called 'troubled_maintenance'

- Step 2:
  - The 'EngineModel' is located in the 'engines' dataframe. The 'Trouble' is located in the 'maintenance' data frame. The 'engines' and 'maintenance' data frames linked by the date in the 'automobile' data frame. Therefore, to investigate which engine models have trouble or are suspected of having trouble, it is necessary to merge them into one.
  - Here I use 'right_join()' to join all the necessary data into one table. The column 'Troubles' in 'troubled_maintenance' acts as the framework to join the data.

- Step 3: Because a car has multiple troubles, there are many engines Model values were duplicated after the joining. So, I use 'distinct' to keep to one value of the Engine Model.

- Step 4: Present the data

The R code is shown below:

```
#Step 1. Data Preparation & Filtering
safe <- "No error"
troubled_maintenance <- maintenance %>%
  filter(Troubles != safe & !is.na(Troubles))
#Step 2. Linking the data frames (The Chain)
#a. Link Cars to the Troubled Maintenance records
# right_join keeps all rows from 'troubled_maintenance' and attaches car info.
troubled_cars<- automobile %>%
  right_join(troubled_maintenance, by = "PlateNumber")
#b. Link Engines to those Troubled Cars
#right_join keeps all rows from 'troubled_cars' and attaches engine info.
target_engines_merged <- engines %>%
  right_join(troubled_cars, by = "EngineModel")
#Step 3. Ensure Uniqueness
# Joins create duplicates if a car has multiple troubles.
# We use distinct() to get a unique list of engines, just like the original question asked.
  target_engines <- target_engines_merged %>%
  distinct(EngineModel, .keep_all = TRUE) %>%
  select(EngineModel, PlateNumber, Troubles) # Pick some columns to show

#Step 4. Present the result
  cat("Total Troubled/Suspected Engines Model Identified:", nrow(target_engines),".\n")
library(knitr)
library(kableExtra)
  kable(head(target_engines,10),
  caption = "Engine models have trouble or are suspected of having trouble")%>%
  kable_styling(latex_options = c("scale_down", "hold_position"))
```

Output Results:

```
## Total Troubled/Suspected Engines Model Identified: 83 .
```

EngineModel	EngineType	NumCylinders	EngineSize	FuelSystem	Horsepower	FuelTypes	Aspiration	SizeGr
E-0001	dohc	four	130	mpfi	111	gas	std	91-190
E-0002	ohcv	six	152	mpfi	154	gas	std	91-190
E-0003	ohc	four	109	mpfi	102	gas	std	91-190
E-0004	ohc	five	136	mpfi	115	gas	std	91-190
E-0005	ohc	five	136	mpfi	110	gas	std	91-190
E-0006	ohc	five	131	mpfi	140	gas	turbo	91-190
E-0007	ohc	five	131	mpfi	160	gas	turbo	91-190
E-0008	ohc	four	108	mpfi	101	gas	std	91-190
E-0009	ohc	six	164	mpfi	121	gas	std	91-190
E-0010	ohc	six	164	mpfi	121	gas	std	91-190

3.d. The top 5 most common troubles related to the engines

Rationale

To find the top 5 most common troubles related to the engines, I follow the steps below:

- Step 1: Because the troubles related to the engines have the value '1' in 'ErrorCodes' column, so I need to remove the other value except '1' in 'ErrorCodes' column using 'filter()', creating a new data frame, called 'engine_troubles_data'
- Step 2: I create a new data frame, called I count 'top_5_engine_issues', to contain the value from the top 5 most common troubles of the engine. I count the frequency of each value in the 'Troubles' column using 'count()', then sort the frequency values from highest to lowest using 'sort = TRUE'. When i use 'head(,5)', I have the top 5 most common engine issues.
- Step 3: Present the result

The R code and Output Results:

```
# What are the top 5 most common troubles related to the engines
  #Step 1. Data Filtering
  engine_troubles_data <- maintenance %>%
  filter(as.character(ErrorCodes) == "1") #only keep the value '1'
  #Step 2. Calculate frequency (The Top 5)
  top_5_engine_issues <- engine_troubles_data %>%
  count(Troubles, sort = TRUE)%>%
  head(5)
  #Step 3: Present the result
  library(knitr)
  library(kableExtra)
    kable(top_5_engine_issues, caption = "Top 5 engine troubles")
```

Table 9: Top 5 engine troubles

Troubles	n
Cylinders	38

Troubles	n
Ignition (finding)	22
Noise (finding)	19
Valve clearance	15
Fans	13

3.e. Do the troubles differ between engine types?

Rationale To determine if the troubles differ between engine types, I follow the below steps:

- Step 1: I create a new data frame 'trouble_analysis_data' to contain all the values of Troubles, except the 'No error' value. Then, I use 'inner_join ()' to join the data from 'automobile' and 'engines' to 'trouble_analysis_data'. Now, I have a data frame that links the trouble and engine types.
- Step 2: I create a new data frame 'summary_table' that contains the list of Engine Types and the number of distinct troubles that they face. - Step 3: Present the result
- Step 4: Conclusion
  - Logic: If each engine type has the same number of troubles, there will be a high chance that the troubles are the same between engine types. If the number of troubles for each engine type is different, there is no chance that the troubles are the same between engine types.
  - Code choice: I use 'unique()' to removes duplicates in the 'number_troubles' column. Then I use 'length' to count the left item. If 'length()' equals 1, it means all the values are the same. Then, the troubles are the same between engine types. If 'length()' is higher than 1, then the troubles are different between engine types.

The R code and Output Results:

```
#Step 1. Data Preparation
safe_value <- "No error"
trouble_analysis_data <- maintenance %>%
filter(Troubles != safe_value & !is.na(Troubles)) %>%
inner_join(automobile, by = "PlateNumber") %>%
inner_join(engines, by = "EngineModel")

#Step 2. Create the summary
library(dplyr)
summary_table <- trouble_analysis_data %>%
group_by(EngineType) %>%
summarise(number_troubles = n_distinct(Troubles))

#Step 3. Present the result
library(knitr)
library(kableExtra)
  kable(summary_table, caption = "Number of trouble for each Engine Types")
```

Table 10: Number of trouble for each Engine Types

EngineType	number_troubles
Other Types	17
dohc	10
dohcv	5
ohc	32
ohcf	11
ohcv	14
rotor	4

```
  #Step 4. Conclusion
  if (length(unique(summary_table$number_troubles)) == 1) {
  print("The troubles are the same between engine types")
} else {
  print("The troubles are different between engine types")
}
```

```
## [1] "The troubles are different between engine types"
```

4

4.a. Error type (ErrorCodes) occurs most frequently

Rationale

To determine the error type that occurs most frequently, I follow the steps below:

- Step 1: I count the frequency each error code appears using 'count()'. Then, I use 'sort=TRUE' to sort the error code from the most frequent code to the least frequent code.
- Step 2: Extract the most frequent error code from the first row and draw a conclusion

The R code is shown below:

```
 #Step 1. Calculate Frequency
  error_counts <- maintenance %>% #count how many times each code appears.
  count(ErrorCodes, sort = TRUE)  # Sorts highest to lowest automatically
  library(knitr)
  library(kableExtra)
    kable(error_counts, caption = "Error Code Frequency Table")
  #Step 2. Conclusion
  # Extract the winner (the first row, since we sorted it)
  top_code <- as.character(error_counts$ErrorCodes[1])
  top_count <- error_counts$n[1]
  cat("The most frequent Error Type is Code:", top_code,".It appeared", top_count)
  if(top_code == "0") {
    Cat("(Meaning: 'No Error' is the most common state).")
  } else if (top_code == "1") {
    cat("(Meaning: 'Engine Failure' is the most common issue).")
  } else if (top_code == "-1") {
    cat("(Meaning: 'Other Component Failure' is the most common issue).")
  }
```

Output Results:

Table 11: Error Code Frequency Table

ErrorCodes	n
1	182
-1	164
0	28

```
## The most frequent Error Type is Code: 1 .It appeared 182
```

```
## (Meaning: 'Engine Failure' is the most common issue).
```

4.b. Factors that might influence the maintenance methods

- The Error Codes and Engine Types were considered as factors influencing the maintenance methods.
- The data frame for this part has already been created above, called 'trouble_analysis_data'
- Then, I will draw two tables to investigate the effect of the factor on maintenance methods and explain if there is a trend.

Table 12: Error Codes vs Methods

	Adjustment	Replacement	Urgent care
-1	43	99	29
1	92	99	0

In the error code, '1' means the engine fails, and '-1' means any other vehicle component fails. Overall, replacement is the most common maintenance method for both types of errors, equal 99 cases for each of the error types. In terms of vehicle component failures, it can be fixed using all three maintenance methods. The urgent care case is the least frequent method to fix errors related to other vehicle components, with 29 cases. The case of adjustment for other vehicle components fails is 43. Regarding the engine failing, they never experienced an urgent care case. This may be because the problem related to the engine can be just fixed with an adjustment and replacement method. This reason makes the number of adjustment methods for engine failures twice that of other vehicle component failures. Moreover, the total number of maintenance methods for engine failures is higher than for other vehicle component failures. This means the engine needs more maintenance than other components of the car.

Table 13: Engine Type vs Methods

	Adjustment	Replacement	Urgent care
dohc	7	9	2
dohcv	4	5	0
ohc	96	148	21
ohcf	10	13	4
ohcv	7	6	1
Other Types	11	9	1
rotor	0	8	0

Overall, the urgent care is the rarest case for all engine types. From the table, 'ohc' is the engine type that is the most common in the maintenance method, which means it can suffer more errors than other engine types. Its values for adjustment, replacement, and urgent care cases are 96, 148, and 21, respectively. Whereas 'rotor' might be a good engine type, because it has the smallest total number of maintenance method cases, only 8 cases. The engine with the second-lowest maintenance cases is 'dohcv' with only nine cases. The 'dohcv' and 'rotor' are good choices for Engine Type as they require little maintenance.