WESTERN SYDNEY
UNIVERSITY

SCHOOL of: Data

ASSIGNMENT COVER SHEET

STUDENT DETAILS

Name: Nguyen Dinh Vinh Hung **Student ID:** 22202467

SUBJECT AND TUTORIAL DETAILS

Subject Name: Analytics Programming **Subject code:** COMP1013

Tutorial Group: Click or tap here to enter text. **Day:** Sunday **Time:** 15:30 PM

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

ASSIGNMENT DETAILS

Title: Individual Assignment **Length:** 2970 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:

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.

Individual Assignment

Nguyen Dinh Vinh Hung

AP-T325WSD-1

Western Sydney University

Assoc. Prof. NGUYEN Tan Luy

November 21st, 2025

Repository link: https://github.com/22202467/comp1013-assignment-22202467

Table of Contents

1. Introduction

This study will examine how engine specifications, engine characteristics, and vehicle maintenance history influence an automobile's overall reliability and performance. Three related data sets (Automobile, Engines, and Maintenance) will be used to address key research questions; including the distribution of horsepower, differences in miles per gallon by fuel type and by front wheel, rear wheel, all wheel drive, and four wheel drive systems, and patterns of confirmed or suspected engine problems.

The Automobile dataset includes the vehicle's dimensions, fuel efficiency, and the engine model number of each vehicle. The Engine dataset includes engine-specific attributes such as type of engine, cylinder configuration, fuel system, and horsepower. The Maintenance dataset logs actual service history of vehicles, which include maintenance methods, trouble descriptions, error codes, and plate numbers linking a vehicle to its historical record.

All the data cleaning, transformation, and statistical analysis was done using RStudio and all the management of the project and files were accomplished through GitHub for version control and to ensure reproducibility. In addition, both tools provide a structured workflow for preparing, analyzing, and reporting the results from the merged datasets.

2. Part 1 - Data Inspection & Cleaning

```
# Part 1
# Read datasets

auto <- read.csv("Automobile.csv", stringsAsFactors = FALSE)

engine <- read.csv("Engine.csv", stringsAsFactors = FALSE)

maint <- read.csv("Maintenance.csv", stringsAsFactors = FALSE)



# Inspect structure

str(auto)

" "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(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(maint)
```

```
## '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" "Ign
ition (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"
...
```

Explanation:

- str() shows us that there are a couple of numeric columns, such as Horsepower, that will need to be converted from their current character type because of the “?”.

- Before we do our cleaning process, we need to know which data type changes we are going to make.

```
# Quick preview
head(engine)
```

```
##   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
```

```
summary(engine)
```

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

```
engine_clean <- engine            # create a working copy
engine_clean[engine_clean == "?"] <- NA
```

Explanation:

- The expression engine_clean == "?" returns a logical matrix marking all cells containing "?".
- Replacing them with NA ensures R recognises these as missing values so they can be handled properly in later steps.

```
rows_affected <- sum(apply(engine == "?", 1, any))
rows_affected
```

```
## [1] 6
```

Explanation:

- apply(..., 1, any) checks row-by-row if any column contains "?".
- sum() counts how many rows were affected in total.
- This directly answers the assignment question.

```
# Horsepower BEFORE cleaning: exclude "?"
hp_before <- engine$Horsepower[engine$Horsepower != "?"]
hp_before <- as.numeric(hp_before)
# Horsepower AFTER cleaning
hp_after <- as.numeric(engine_clean$Horsepower)
```

```
summary(hp_before)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    48.0    80.0   102.0   114.1   144.0   288.0 

summary(hp_after)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    48.0    80.0   102.0   114.1   144.0   288.0       1
```

Explanation:

- Only a single horsepower value was missing in this dataset.
- Because the number of "?" entries is extremely small, replacing them with NA does not meaningfully change the distribution summary (means, median, min/max remain stable).
- This should be mentioned explicitly in your written report.

```
auto$BodyStyles <- factor(auto$BodyStyles)
engine_clean$FuelTypes <- factor(engine_clean$FuelTypes)
maint$ErrorCodes <- factor(maint$ErrorCodes)
```

Explanation:

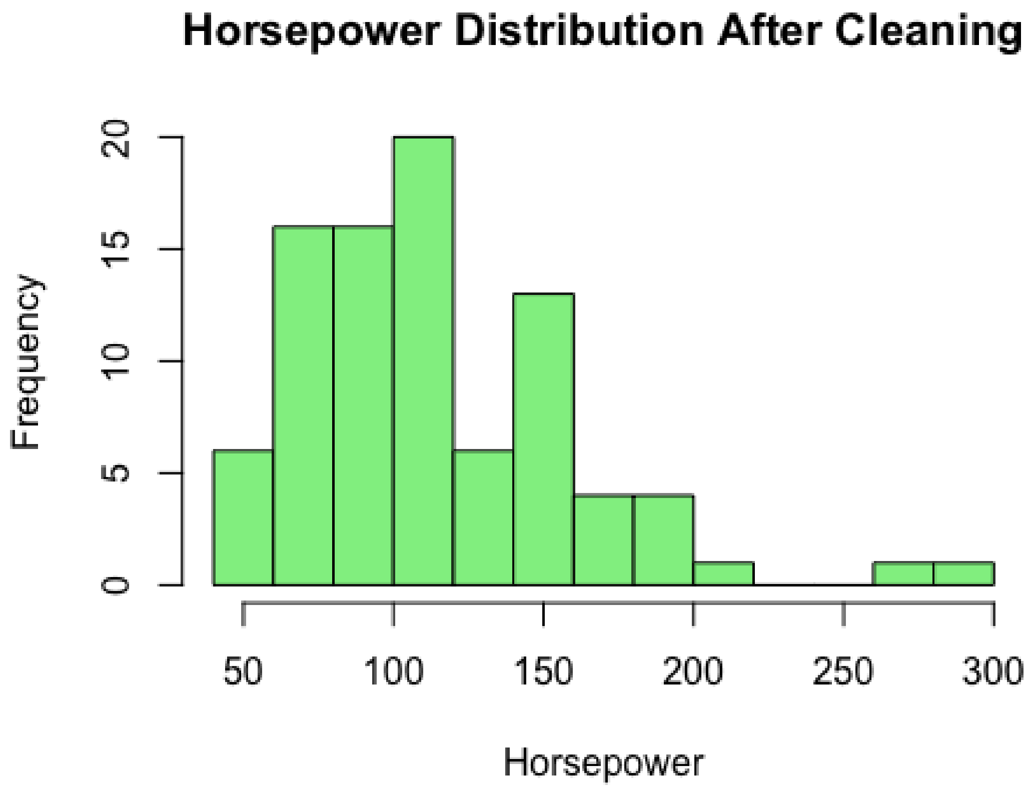
- Factors are the correct data type for categorical variables.
- This also ensures bar charts and grouped analyses in later sections behave correctly.

```
# Compute median (excluding NA)
hp_median <- median(hp_before, na.rm = TRUE)

# Convert horsepower column to numeric first
engine_clean$Horsepower <- as.numeric(engine_clean$Horsepower)

# Replace NA horsepower with median
engine_clean$Horsepower[is.na(engine_clean$Horsepower)] <- hp_median
```

Explanation:

- Median imputation is robust to outliers and does not distort the distribution.
- This step removes all missing values and ensures the horsepower variable is fully numeric for later analysis.

```
hist(engine_clean$Horsepower,
     breaks = 10,
     col = "lightgreen",
     main = "Horsepower Distribution After Cleaning",
     xlab = "Horsepower")
```



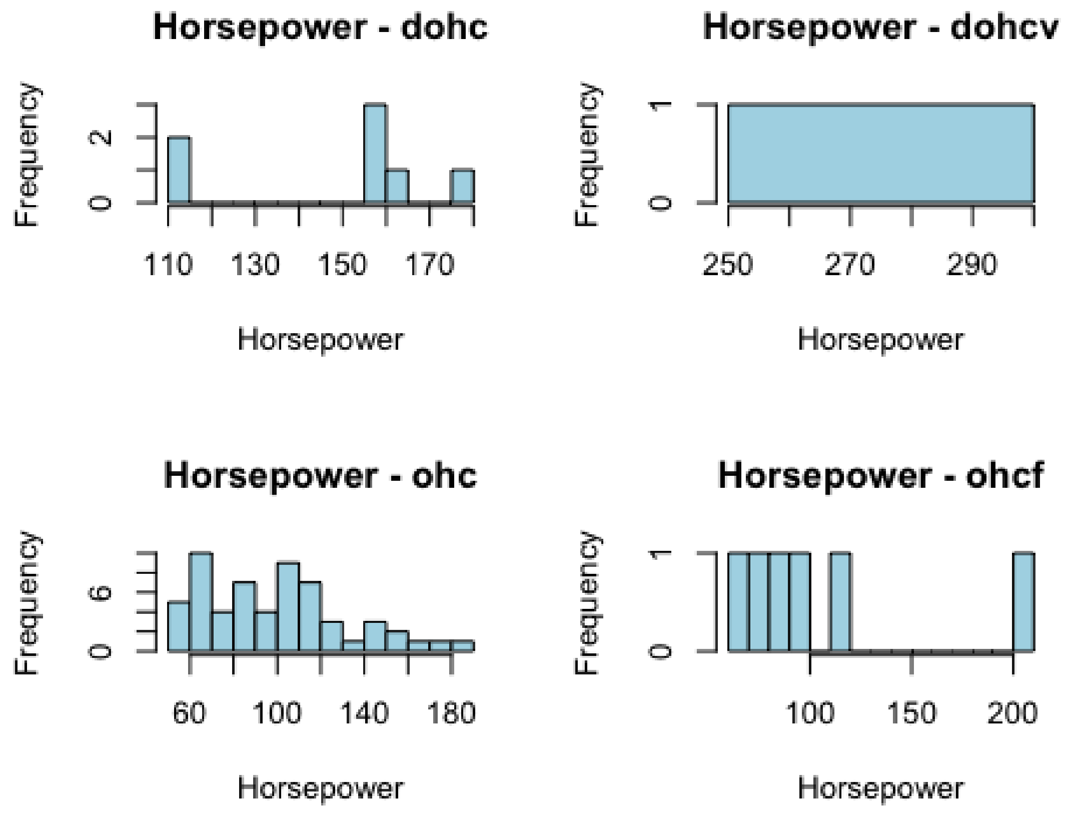
3. Part 2 - Horsepower Distribution Analysis

```
#Part 2
table(engine_clean$EngineType)

##
##  dohc dohcv   ohc  ohcf  ohcv rotor
##     7     1    58     6     9     2
```

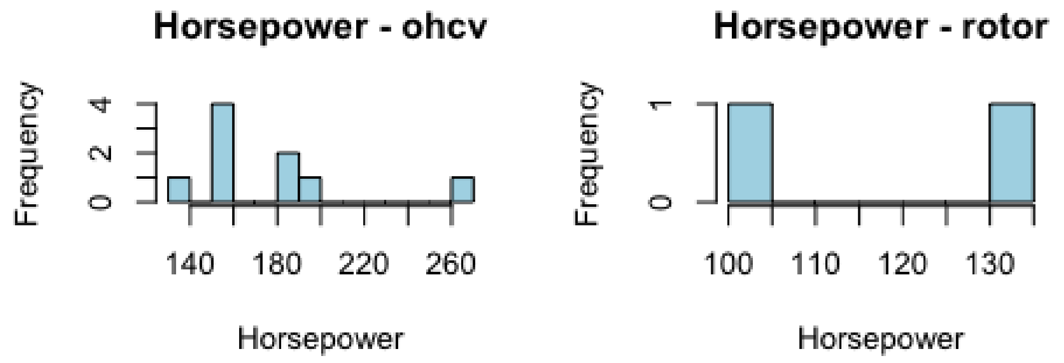
Explanation:

- This verifies that the categorical variable EngineType contains valid labels.
- Since we replaced "?" with NA in Part 1, those invalid entries no longer appear.
- This step ensures we can correctly group horsepower by engine type.

```
engine_types <- levels(factor(engine_clean$EngineType))

# Plot a histogram for each engine type
par(mfrow = c(2, 2))     # Arrange plots in 2x2 grid (adjust if needed)

for (type in engine_types) {
  subset_hp <- engine_clean$Horsepower[engine_clean$EngineType == type]
  hist(subset_hp,
       breaks = 10,
       col = "lightblue",
       main = paste("Horsepower -", type),
       xlab = "Horsepower")
}
```



```
par(mfrow = c(1, 1))      # Reset layout
```



Explanation:

- We loop through each engine type using a simple for loop.
- Each histogram shows the distribution for that specific category.

```
# Create engine size groups
engine_clean$EngineSizeGroup <- cut(engine_clean$EngineSize,
                                    breaks = c(0, 90, 190, 299, Inf),
                                    labels = c("60-90", "91-190", "191-299",
"300+"),
                                    right = TRUE)
```

Explanation:

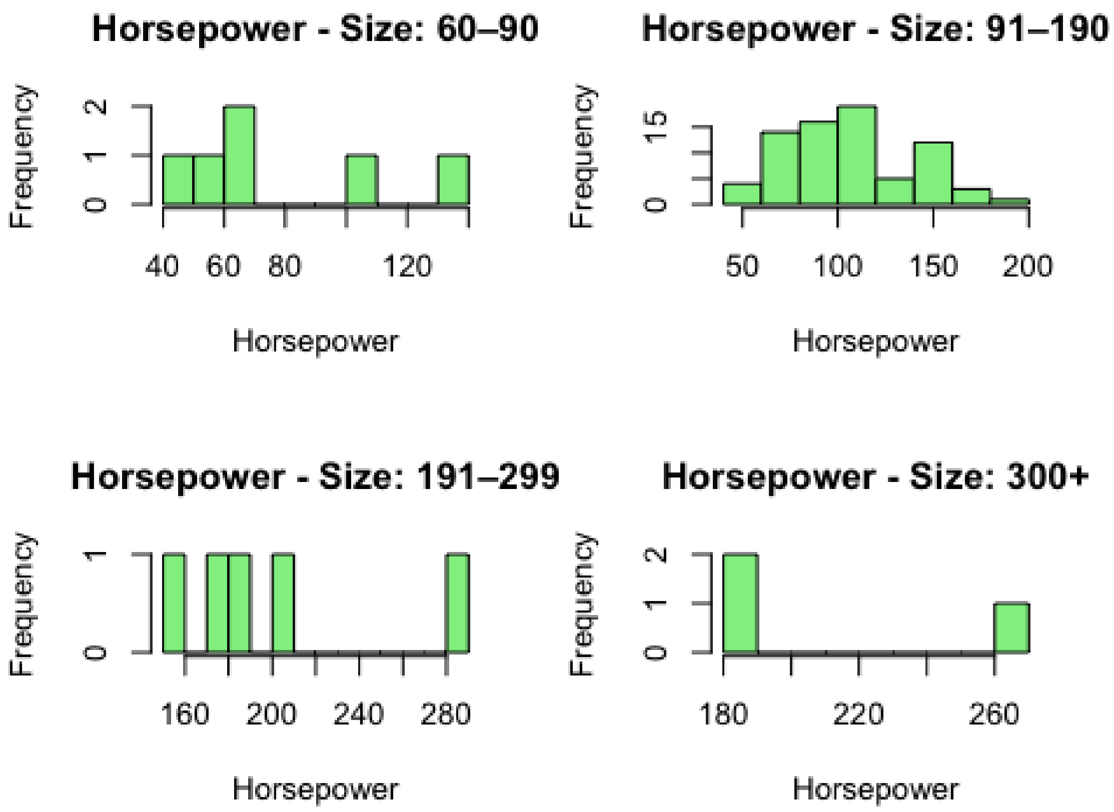
- cut() assigns each EngineSize value into a category.
- The chosen ranges follow the assignment's suggestion and maintain approximately balanced group sizes.
- The new variable EngineSizeGroup becomes an ordered categorical variable.

```
table(engine_clean$EngineSizeGroup)

##
##   60-90  91-190 191-299    300+
##       6      74       5       3
```

Explanation:

- A frequency table confirms that every engine has been correctly assigned to a size group.
- This ensures no missing or incorrectly binned values.

```
size_groups <- levels(engine_clean$EngineSizeGroup)

par(mfrow = c(2, 2))   # 2x2 plot layout

for (grp in size_groups) {
  subset_hp <- engine_clean$Horsepower[engine_clean$EngineSizeGroup == grp]

  hist(subset_hp,
       breaks = 10,
       col = "lightgreen",
       main = paste("Horsepower - Size:", grp),
       xlab = "Horsepower")
}
```



```
par(mfrow = c(1, 1))
```

Explanation:

- This allows clear visual comparison of horsepower across engine size categories.
- We can quickly observe trends, such as higher horsepower for engines in larger displacement groups.

Interpretation:

- The Distribution of Horsepower was represented for each Engine Type and Engine Size Group
- Across the engine types, ohc and ohcf engines exhibited broader horsepower ranges, while rotor engines showed more concentrated distributions.
- For the Engine Size Groups, Horsepower was shown to be directly related to Engine Displacement with each Engine Size Group showing a greater increase in Horsepower than the previous Engine Size Group (e.g. the Smaller 60-90 Group had the lowest average Horsepower value; the Largest 300+ Group had the highest).
- Histograms were used to represent the data because Horsepower is a Continuous Numeric Variable, and Histograms are used to show distributional characteristics (shape, spread and outliers) of such variables.

4. Part 3 - Vehicle Efficiency & Troubles

```
#Part 3
# Merge Automobile and Engine datasets by EngineModel
auto_engine <- merge(auto, engine_clean, by = "EngineModel")
```

Explanation:

- This inner join attaches FuelTypes, Horsepower, and engine characteristics to each car.
- The resulting dataset supports analysis of MPG by fuel type and drive configuration.

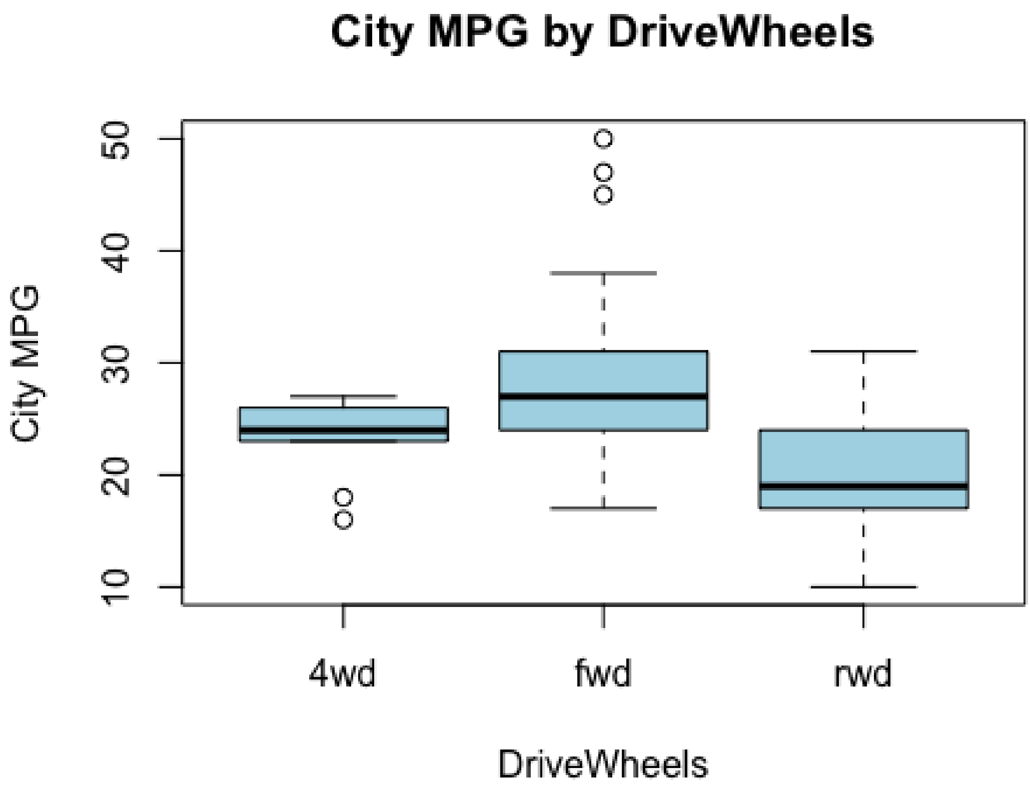
```
# Split data by fuel type
gas_mpg <- auto_engine$CityMpg[auto_engine$FuelTypes == "gas"]
diesel_mpg <- auto_engine$CityMpg[auto_engine$FuelTypes == "diesel"]

# Summary statistics
summary(gas_mpg)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.00   19.00   24.00   24.28   28.00   50.00

summary(diesel_mpg)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   22.00   25.00   29.00   30.30   36.25   45.00

# Two-sample t-test (unequal variances)
t_test_result <- t.test(gas_mpg, diesel_mpg)
t_test_result

##
##  Welch Two Sample t-test
##
## data:  gas_mpg and diesel_mpg
## t = -3.9004, df = 22.592, p-value = 0.0007392
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -9.218138 -2.824648
## sample estimates:
## mean of x mean of y
##  24.27861  30.30000
```

Interpretation:

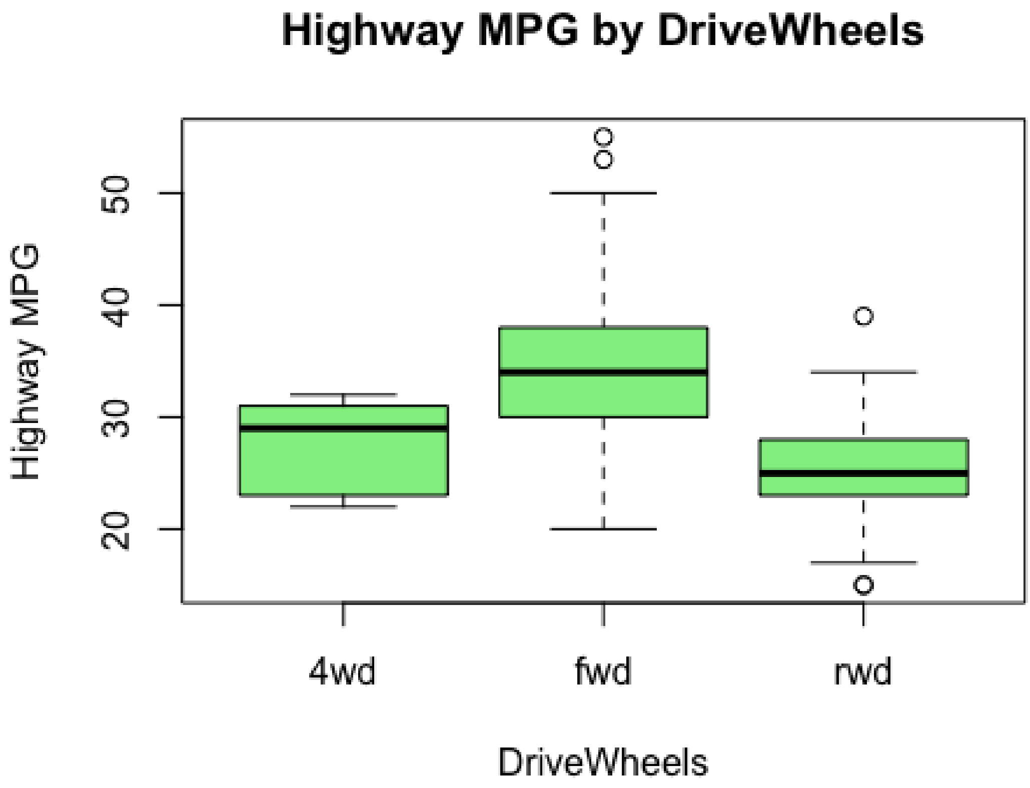
- Gasoline cars have an average city miles per gallon (mpg) of 24.28 mpg, while diesel cars have a greater average of 30.30 mpg.
- The results from the Welch t-test were as follows:
  - t = -3.9004; degrees of freedom = 22.592; p-value = 0.0007392.
  - Therefore, the 95% confidence interval for the difference in means is [-9.218, -2.825] mpg.

- Since p < .001, there is a statistically significant relationship between the vehicle's fuel type and its fuel efficiency. On average, diesel vehicles are able to get approximately 3 to 9 mpg better than gasoline vehicles.

```r
# Boxplot for CityMpg by DriveWheels
boxplot(auto_engine$CityMpg ~ auto_engine$DriveWheels,
        col = "lightblue",
        main = "City MPG by DriveWheels",
        xlab = "DriveWheels",
        ylab = "City MPG")
```



```r
# Boxplot for HighwayMpg by DriveWheels
boxplot(auto_engine$HighwayMpg ~ auto_engine$DriveWheels,
        col = "lightgreen",
        main = "Highway MPG by DriveWheels",
        xlab = "DriveWheels",
        ylab = "Highway MPG")
```



Interpretation:

- FWD vehicles show the highest CityMpg and HighwayMpg, while RWD and 4WD vehicles show noticeably lower fuel efficiency.
- The median MPG of FWD vehicles is clearly higher, indicating that drive configuration influences fuel economy, with FWD being the most efficient configuration in the dataset.

```
# Extract only rows with actual trouble (confirmed OR suspected)
trouble_data <- maint[maint$ErrorCodes != "0", ]
nrow(trouble_data)

## [1] 346
```

Explanation:

- This provides the dataset of vehicles that experienced any form of engine trouble.
- We exclude rows where ErrorCodes == 0 (no error).

```
trouble_counts <- sort(table(trouble_data$Troubles), decreasing = TRUE)
top5_troubles <- head(trouble_counts, 5)
top5_troubles
```

```
##
##          Cylinders            Chassis Ignition (finding)    Noise (finding
)
##                 38                 25                 22                 1
9
##         Worn tires
##                 16
```

Interpretation:

- The most frequent issues were Cylinders (38), Chassis (25), Ignition (finding) (22), Noise (finding) (19), and Worn tires (16).

- These represent the most common mechanical concerns affecting customers.

```
# First join Maintenance with Automobile to attach EngineModel
maint_auto <- merge(maint, auto, by = "PlateNumber")

# Then join with Engine dataset to attach EngineType
maint_full <- merge(maint_auto, engine_clean, by = "EngineModel")

# Filter only trouble rows again (confirmed or suspected)
maint_trouble <- maint_full[maint_full$ErrorCodes != "0", ]

# Count trouble cases per engine type
trouble_by_engine <- sort(table(maint_trouble$EngineType), decreasing = TRUE)
trouble_by_engine

##
##   ohc  ohcf  dohc  ohcv dohcv rotor
##   265    27    18    14     9     8
```

Explanation:

- We perform a two-step merge because Maintenance links to Automobile, and Automobile links to Engine.

- Trouble counts per EngineType help identify whether certain engine designs (e.g., ohc, dohc) experience more issues.

5. Part 4 - Error Types & Maintenance Method Analysis

```
#Part 4
# Merge Maintenance with Automobile
maint_auto <- merge(maint, auto, by = "PlateNumber")

# Merge with Engine
maint_full <- merge(maint_auto, engine_clean, by = "EngineModel")
```

Explanation:

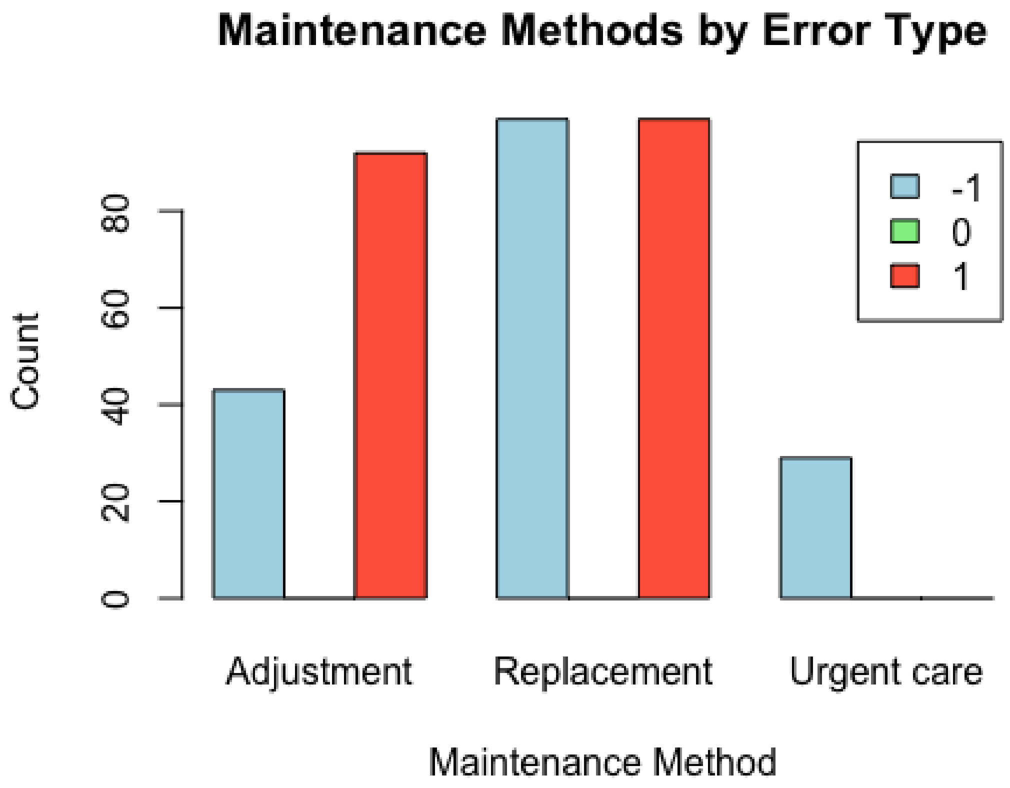
- We now have a single dataset containing:
    - ErrorCodes
    - Troubles
    - Methods
    - BodyStyles
    - FuelTypes
    - EngineType
- This merged dataset allows comparison across factors.

```
table(maint_full$ErrorCodes)

## 
##  -1   0   1 
## 171  29 191
```

Interpretation:

- Most maintenance visits involve actual trouble, aligning with Part 3 findings:
    - 0 (No error) ≈ small portion of records
    - -1 (Suspected trouble) ≈ moderate number
    - 1 (Confirmed trouble) ≈ largest share

```
# Cross-tabulation
error_method_table <- table(maint_full$ErrorCodes, maint_full$Methods)
error_method_table

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

barplot(error_method_table,
        beside = TRUE,
        col = c("lightblue", "lightgreen", "tomato"),
        main = "Maintenance Methods by Error Type",
        xlab = "Maintenance Method",
        ylab = "Count",
        legend = rownames(error_method_table))
```



Explanation:

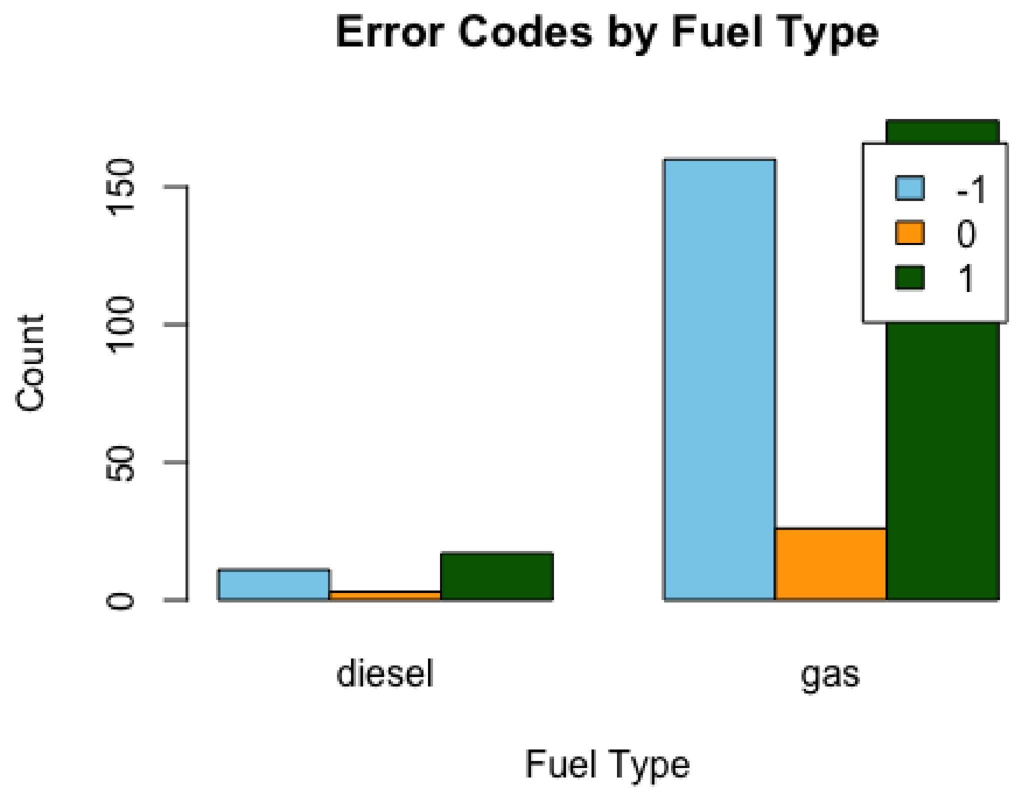
- beside = TRUE shows groups side-by-side for easy comparison.

Interpretation:

- Troubled confirmed cases (Error Code = 1) are found with greater frequency in the category of Replacement, which is indicative of part replacement being required as a result of mechanical failure of an engine component.
- There are fewer suspected cases (-1), and they are also more frequent in the category of Adjustment than in other categories. This would indicate that these cases represent minor or uncertain conditions that do not require part replacement to correct.
- No-error cases (0) were found throughout all categories at very low frequencies and most likely represent routine inspections or mis-classified entries.

```
error_fuel_table <- table(maint_full$ErrorCodes, maint_full$FuelTypes)
error_fuel_table

##
##       diesel gas
##    -1     11 160
##    0       3  26
##    1      17 174
```

```
barplot(error_fuel_table,
        beside = TRUE,
        col = c("skyblue", "orange", "darkgreen"),
        main = "Error Codes by Fuel Type",
        xlab = "Fuel Type",
        ylab = "Count",
        legend = rownames(error_fuel_table))
```



Interpretation:

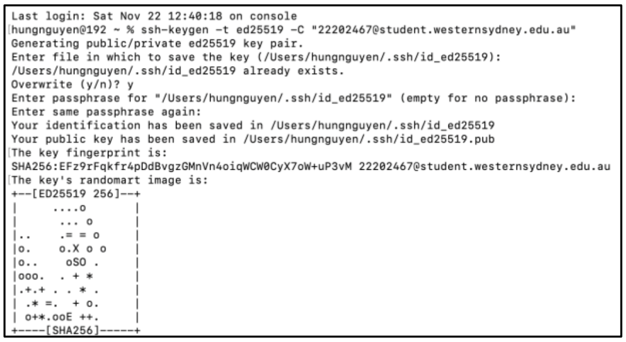
- More than any other error category, Gasoline Engines show a much greater frequency in the entire data set.
- Diesel Engines show fewer trouble reports in all categories, including Suspected (-1), Confirmed (1) and Still show presence.
- These ratios indicate that while there are fewer Diesel Engines overall, Diesel Engines do not display higher trouble frequencies than Gasoline Engines.

```
table(maint_full$ErrorCodes, maint_full$BodyStyles)

##
##      convertible hardtop hatchback sedan wagon
##   -1          10       6        60    73    22
##   0             1       0         10     13      5
##   1             5       2         65     99     20
```

6. GitHub Submission

1. Generating the SSH Key Pair

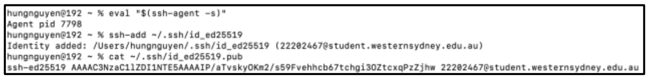
- Command: ssh-keygen -t ed25519 -C "22202467@student.westernsydney.edu.au"
    - Result: The private key was saved to /Users/hungnguyen/.ssh/id_ed25519 and the public key was saved to /Users/hungnguyen/.ssh/id_ed25519.pub.



```
Last login: Sat Nov 22 12:40:18 on console
hungnguyen@192 ~ % ssh-keygen -t ed25519 -C "22202467@student.westernsydney.edu.au"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/Users/hungnguyen/.ssh/id_ed25519):
/Users/hungnguyen/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase for "/Users/hungnguyen/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/hungnguyen/.ssh/id_ed25519
Your public key has been saved in /Users/hungnguyen/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:EFz9rFqkfr4pDdBvgzGMnVn4oiqWCW0CyX7oW+uP3vM 22202467@student.westernsydney.edu.au
The key's randomart image is:
+--[ED25519 256]--+
|      ....o      |
|       ... o     |
|..     .= = o    |
|o.     o.X o o   |
|o..     oSO .    |
|ooo.   . + *     |
|.+.+ . . * .     |
| .* =.  + o.     |
| o+*.ooE ++.     |
+----[SHA256]-----+
```

2. Starting the SSH Agent and Adding the Key

- Command 1: eval "$(ssh-agent -s)"
- Command 2: ssh-add ~/.ssh/id_ed25519
- Command 3: cat ~/.ssh/id_ed25519.pub
    - Then paste the key to GitHub account (ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIP/aTvskyOKm2/s59Fvehhcb67tchgi3OZtcxqPzZjhw 22202467@student.westernsydney.edu.au)

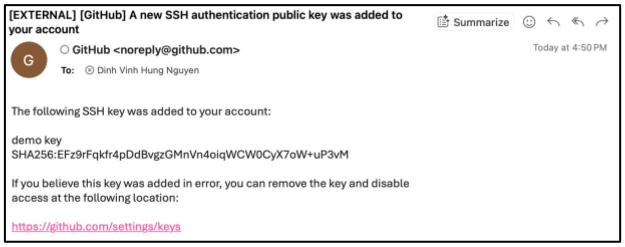


```
hungnguyen@192 ~ % eval "$(ssh-agent -s)"
Agent pid 7798
hungnguyen@192 ~ % ssh-add ~/.ssh/id_ed25519
Identity added: /Users/hungnguyen/.ssh/id_ed25519 (22202467@student.westernsydney.edu.au)
hungnguyen@192 ~ % cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIP/aTvskyOKm2/s59Fvehhcb67tchgi3OZtcxqPzZjhw 22202467@student.westernsydney.edu.au
```

3. Verifying SSH Authentication with GitHub

Then check if the newly created and loaded key was correctly configured and accepted by GitHub.

- Command: ssh -T git@github.com, or check if received confirmation email from GitHub
- Result: Hi 22202467! You've successfully authenticated... This confirmation means your public key is correctly associated with your GitHub account, and the SSH setup is complete.



[EXTERNAL] [GitHub] A new SSH authentication public key was added to your account

○ GitHub <noreply@github.com>

To: Dinh Vinh Hung Nguyen

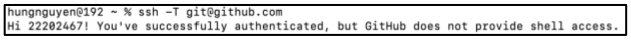
Today at 4:50 PM

The following SSH key was added to your account:

demo key
SHA256:EFz9rFqkfr4pDdBvgzGMnVn4oiqWCW0CyX7oW+uP3vM

If you believe this key was added in error, you can remove the key and disable access at the following location:

https://github.com/settings/keys

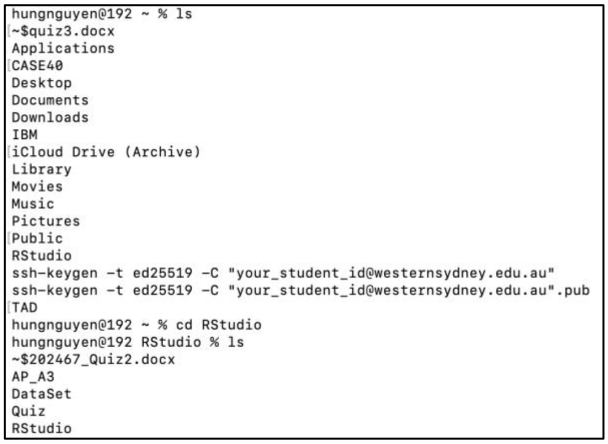


```
hungnguyen@192 ~ % ssh -T git@github.com
Hi 22202467! You've successfully authenticated, but GitHub does not provide shell access.
```

4. Navigating to the Assignment Directory

Move into the specific directory where you wanted to clone the Git repository.

- Command 1: ls
  - To listed the files in your home directory (~).
- Command 2: cd RStudio
  - To change the current working directory to the RStudio folder.

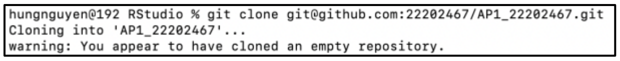


```
hungnguyen@192 ~ % ls
~$quiz3.docx
Applications
CASE40
Desktop
Documents
Downloads
IBM
iCloud Drive (Archive)
Library
Movies
Music
Pictures
Public
RStudio
ssh-keygen -t ed25519 -C "your_student_id@westernsydney.edu.au"
ssh-keygen -t ed25519 -C "your_student_id@westernsydney.edu.au".pub
TAD
hungnguyen@192 ~ % cd RStudio
hungnguyen@192 RStudio % ls
~$202467_Quiz2.docx
AP_A3
DataSet
Quiz
RStudio
```

5. Cloning the Empty Repository

Clone the assignment repository from GitHub using the secure SSH URL.

- Command: git clone git@github.com:22202467/AP1_22202467.git
  - Action: Downloaded the repository structure into a new folder named AP1_22202467 within your current RStudio directory.



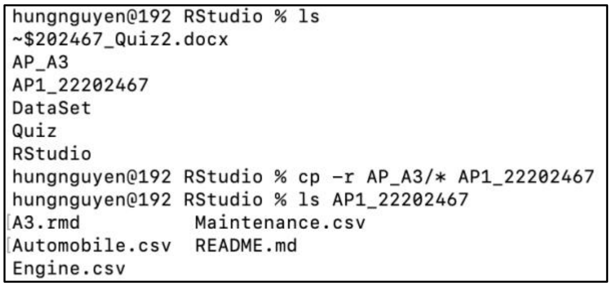
```
hungnguyen@192 RStudio % git clone git@github.com:22202467/AP1_22202467.git
Cloning into 'AP1_22202467'...
warning: You appear to have cloned an empty repository.
```

6. Copying Assignment Files into the Repository

Copy the assignment files from my existing working folder (AP_A3) into the newly cloned Git repository folder.

- Command: cp -r AP_A3/* AP1_22202467

- Action: Recursively copied all contents (*) from the AP_A3 folder into the AP1_22202467 folder.

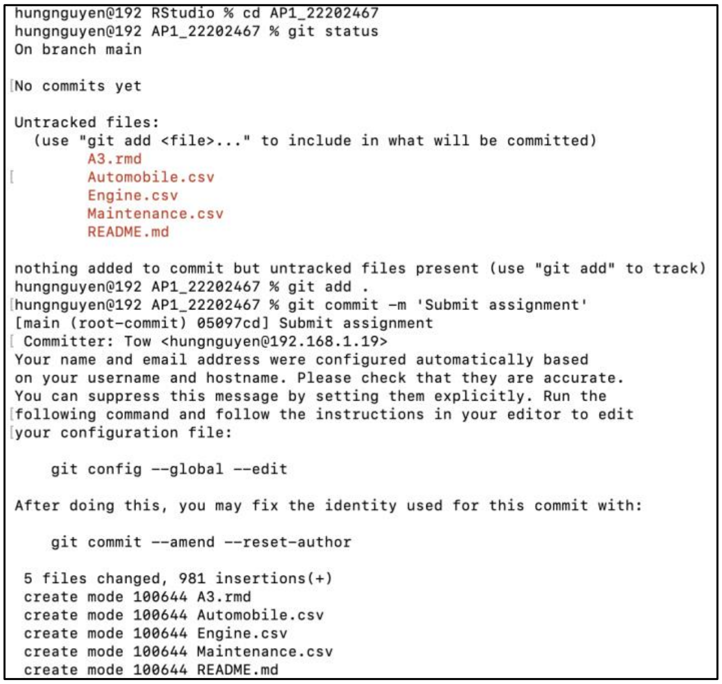


```
hungnguyen@192 RStudio % ls
~$202467_Quiz2.docx
AP_A3
AP1_22202467
DataSet
Quiz
RStudio
hungnguyen@192 RStudio % cp -r AP_A3/* AP1_22202467
hungnguyen@192 RStudio % ls AP1_22202467
A3.rmd          Maintenance.csv
Automobile.csv  README.md
Engine.csv
```

7. Preparing and Committing the Files

Navigate into the repository folder and used Git commands to stage and commit the files.

- Command 1: cd AP1_22202467
  - Action: Changed the working directory into the cloned repository.
- Command 2: git status
  - Action: Showed that all the copied files were present as Untracked files.
- Command 3: git add .
  - Action: Staged all the changes (the new files) for the next commit. The . means "all files in the current directory."
- Command 4: git commit -m 'Submit assignment'
  - Action: Created the first commit (a snapshot of your staged files) in your local repository history, using the message 'Submit assignment'.

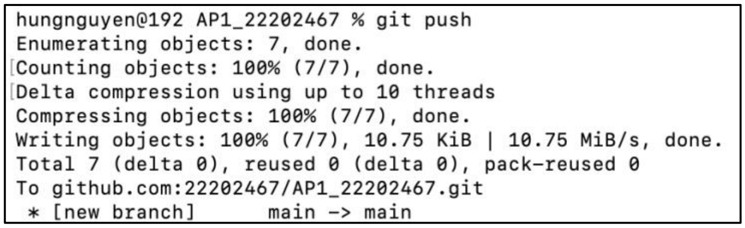


```
hungnguyen@192 RStudio % cd AP1_22202467
hungnguyen@192 AP1_22202467 % git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        A3.rmd
        Automobile.csv
        Engine.csv
        Maintenance.csv
        README.md

nothing added to commit but untracked files present (use "git add" to track)
hungnguyen@192 AP1_22202467 % git add .
hungnguyen@192 AP1_22202467 % git commit -m 'Submit assignment'
[main (root-commit) 05097cd] Submit assignment
 Committer: Tow <hungnguyen@192.168.1.19>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 5 files changed, 981 insertions(+)
 create mode 100644 A3.rmd
 create mode 100644 Automobile.csv
 create mode 100644 Engine.csv
 create mode 100644 Maintenance.csv
 create mode 100644 README.md
```

8. Pushing the Local Commit to GitHub

Finally, transfer my local commit history to the remote repository on GitHub.

- Command: git push
    - Action: Sent the committed changes from your local main branch to the remote repository on GitHub.



```
hungnguyen@192 AP1_22202467 % git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 10 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 10.75 KiB | 10.75 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:22202467/AP1_22202467.git
 * [new branch]      main -> main
```

7. Conclusion

The Analysis of the three (Automobile, Engine, and Maintenance) datasets was conducted in order to determine how engine characteristics affect the reliability and performance of an automobile. The results clearly indicate that there is a direct correlation as larger engines produce more horsepower, diesel and front wheel drive (FWD) vehicles have better fuel economy, and most maintenance issues are related to cylinders, chassis, ignition and noise. Most confirmed problems were solved through replacement of parts or components. Overall, the findings demonstrate a direct relationship between the design of an engine, fuel efficiency of the vehicle, and the service pattern of the vehicle using a reproducible RStudio-GitHub workflow.