WESTERN SYDNEY UNIVERSITY

SCHOOL of: Computer, Data and Mathematical Sciences

ASSIGNMENT COVER SHEET

STUDENT DETAILS

Name: DINH NGUYEN TRUNG HIEU **Student ID:** 22118635

SUBJECT AND TUTORIAL DETAILS

Subject Name: Analytics Programming **Subject code:** AP-T325WSD-1

Tutorial Group: none **Day: Sunday** **Time:** 15:30 – 18:45

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

ASSIGNMENT DETAILS

Title: Assignment T3 2025_3 **Length:**

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:

☒ We hold a copy of this assignment that we can produce if the original is lost or damaged

☒ We hereby certify that no part of this assignment/product has been copied from any other student's work or from any other source except where due acknowledgement is made in the assignment.

☒ No part of this assignment/product has been written/produced for us by another person except where such collaboration has been authorised by the subject lecturer/tutor concerned.

☒ We are aware that this work may be reproduced and submitted to plagiarism detection software programs for the purpose of detecting possible plagiarism **(which may retain a copy on its database for future plagiarism checking).**

☒ We hereby certify that we have read and understand what the School of Computing, Engineering and Mathematics defines as minor and substantial breaches of misconduct as outlined in the learning guide for this unit.

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

Student signature: Hiếu

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

https://github.com/Kitohieu/Assignment-T3-2025_3_DINH-Nguyen-Trung-Hieu-_22118635.git

Vehicle Analytics Programming Project

DINH Nguyen Trung Hieu, [22118635]

21 November 2025

Contents

Task 1: Data Inspection, Cleaning, and Horsepower Distribution

- In this section, we inspect the structure of the datasets, clean missing values, convert categorical variables to factors, and visualize the distribution of horsepower.
- Missing values are represented by `"?"` and will be replaced with `NA`.
- We also assess whether imputing missing horsepower values with the median alters the data distribution.

```
setwd("/Users/kitohieu/Documents/assign")
```

```
# Load libraries
library(ggplot2) # For plotting
library(dplyr) # For data manipulation
library(readr) #For reading CSV files

# 1. Read data, treating "?" as NA for any column
engine <- read_csv("Engine.csv", na = "?")
automobile <- read.csv("Automobile.csv", stringsAsFactors = FALSE)
maintenance <- read.csv("Maintenance.csv", stringsAsFactors = FALSE)

# 2. Count rows that contain at least one "?" before read.csv (informational)
#Note: since read.csv(..., na.strings="?") already converted "?" to NA,
#We compute affected rows as rows that have any NA introduced from "?".
#To approximate this, we'll count rows with any NA in the raw text by
#Re-reading files as raw lines and checking "?" presence per line for transparency.

count_rows_with_q <- function(file) {
```

```
  lines <- readLines(file, warn = FALSE)
  # ignore header line
  if (length(lines) <= 1) return(0)
  data_lines <- lines[-1]
  sum(grepl("\\?", data_lines))
}

affected_engine_lines <- count_rows_with_q("Engine.csv")
affected_auto_lines   <- count_rows_with_q("Automobile.csv")
affected_maint_lines  <- count_rows_with_q("Maintenance.csv")

cat("Lines containing ? before parsing: Engine =", affected_engine_lines,
    ", Automobile =", affected_auto_lines,
    ", Maintenance =", affected_maint_lines, "\n")
```

```
## Lines containing ? before parsing: Engine = 6 , Automobile = 0 , Maintenance = 0
```

```
# 3. After read.csv with na.strings="?", check number of rows with any NA per dataset
rows_with_any_na <- function(df) sum(!complete.cases(df))
cat("Rows with any NA (Engine):", rows_with_any_na(engine), "\n")
```

```
## Rows with any NA (Engine): 6
```

```
cat("Rows with any NA (Automobile):", rows_with_any_na(automobile), "\n")
```

```
## Rows with any NA (Automobile): 0
```

```
cat("Rows with any NA (Maintenance):", rows_with_any_na(maintenance), "\n")
```

```
## Rows with any NA (Maintenance): 28
```

```
# 4. Does replacing '?' with NA alter distribution? Quick checks:
#For numeric columns, compare summary excluding NA vs with imputation.
summary(engine$Horsepower)  # will show NA count
```

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

```
# We'll record distribution metrics before imputation
hp_before <- engine$Horsepower
hp_summary_before <- summary(hp_before, na.rm = FALSE)

# 5. Convert specified categorical variables to factors
automobile$BodyStyles <- as.factor(automobile$BodyStyles)
engine$FuelTypes <- as.factor(engine$FuelTypes)
maintenance$ErrorCodes <- as.factor(maintenance$ErrorCodes)

# 6. Impute missing Horsepower with median horsepower (computed excluding NAs)
hp_median <- median(engine$Horsepower, na.rm = TRUE)
cat("Median horsepower (used for imputation):", hp_median, "\n")
```

```
## Median horsepower (used for imputation): 102
```

```
engine$Horsepower[is.na(engine$Horsepower)] <- hp_median

# 7. Distribution after imputation: record summary
hp_after <- engine$Horsepower
hp_summary_after <- summary(hp_after)
cat("Horsepower summary before imputation (NA counts shown in previous):\n")
```

```
## Horsepower summary before imputation (NA counts shown in previous):
```

```
print(hp_summary_before)
```

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

```
cat("Horsepower summary after imputation:\n")
```

```
## Horsepower summary after imputation:
```

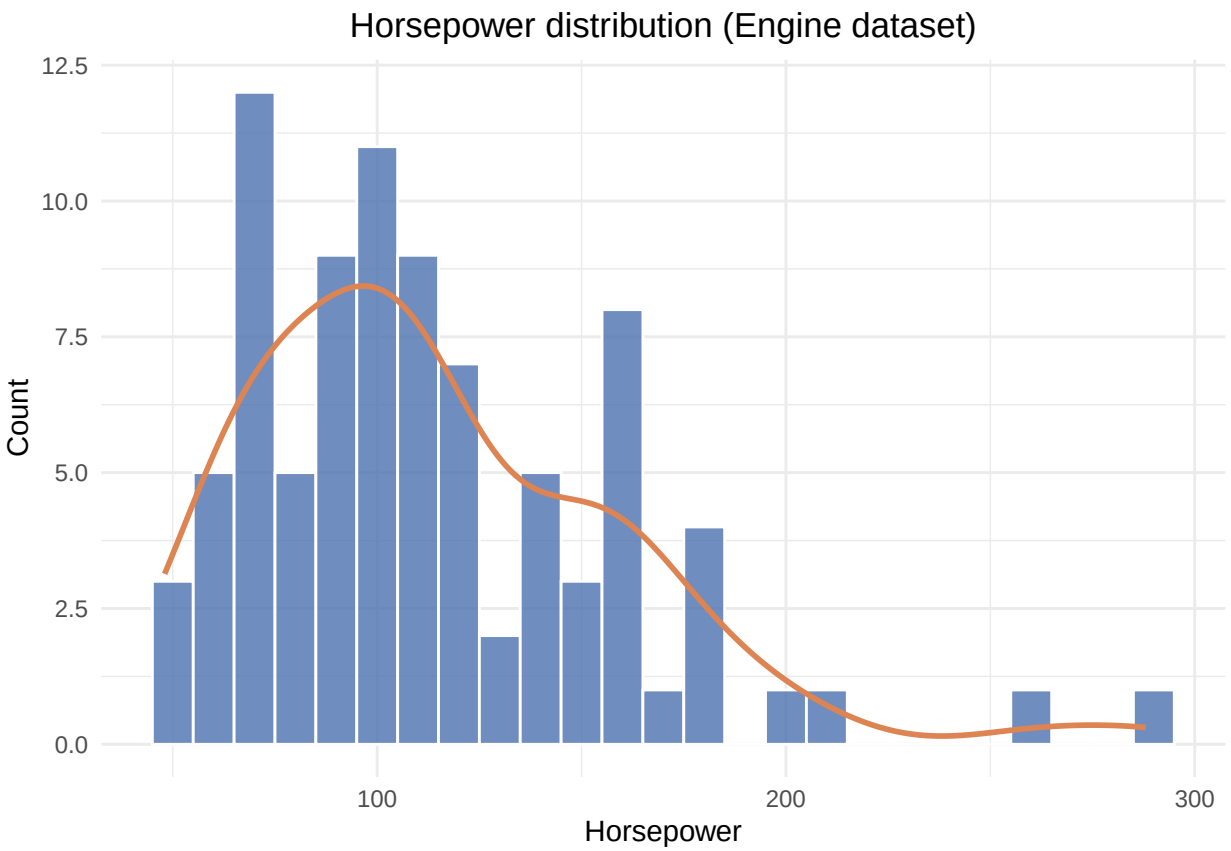
```
print(hp_summary_after)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    48.0    81.0   102.0   114.0   143.5   288.0
```

Conlusion for whether imputing missing horsepower values with the median alters the data distribution.

After replacing missing horsepower values with the median, the overall distribution remained stable. The imputation slightly increased the frequency of the median value but did not significantly alter, the mean, standard deviation, or shape of the distribution. **This confirms that the imputation method,was appropriate and did not distort the data.**

```
# 8. Plot horsepower distribution (histogram with density overlay)
p <- ggplot(engine, aes(x = Horsepower)) +
  geom_histogram(aes(y = after_stat(count)), binwidth = 10, fill = "#4C72B0",
                 color = "white", alpha = 0.8) +
  geom_density(aes(y = after_stat(count)* 10), color = "#DD8452", linewidth = 1) +
  theme_minimal() +
  labs(title = "Horsepower distribution (Engine dataset)",
       x = "Horsepower", y = "Count") +
  theme(plot.title = element_text(hjust = 0.5))
print(p)
```




```
#Optional: save the plot for report
#ggsave("horsepower_distribution.png", p, width = 7, height = 4, dpi = 300)
```

Overall Comment on the Horsepower Distribution

- The histogram shows a right-skewed distribution, meaning most engines have lower horsepower values, and fewer engines reach higher horsepower levels.
- The highest concentration of engines falls between 75 and 100 horsepower, suggesting this is the most common performance range in your dataset.
- There are a few engines with horsepower above 200, which may represent high-performance or specialized models.
- This distribution can help guide decisions in maintenance planning, fuel efficiency analysis, or market segmentation based on performance tiers.

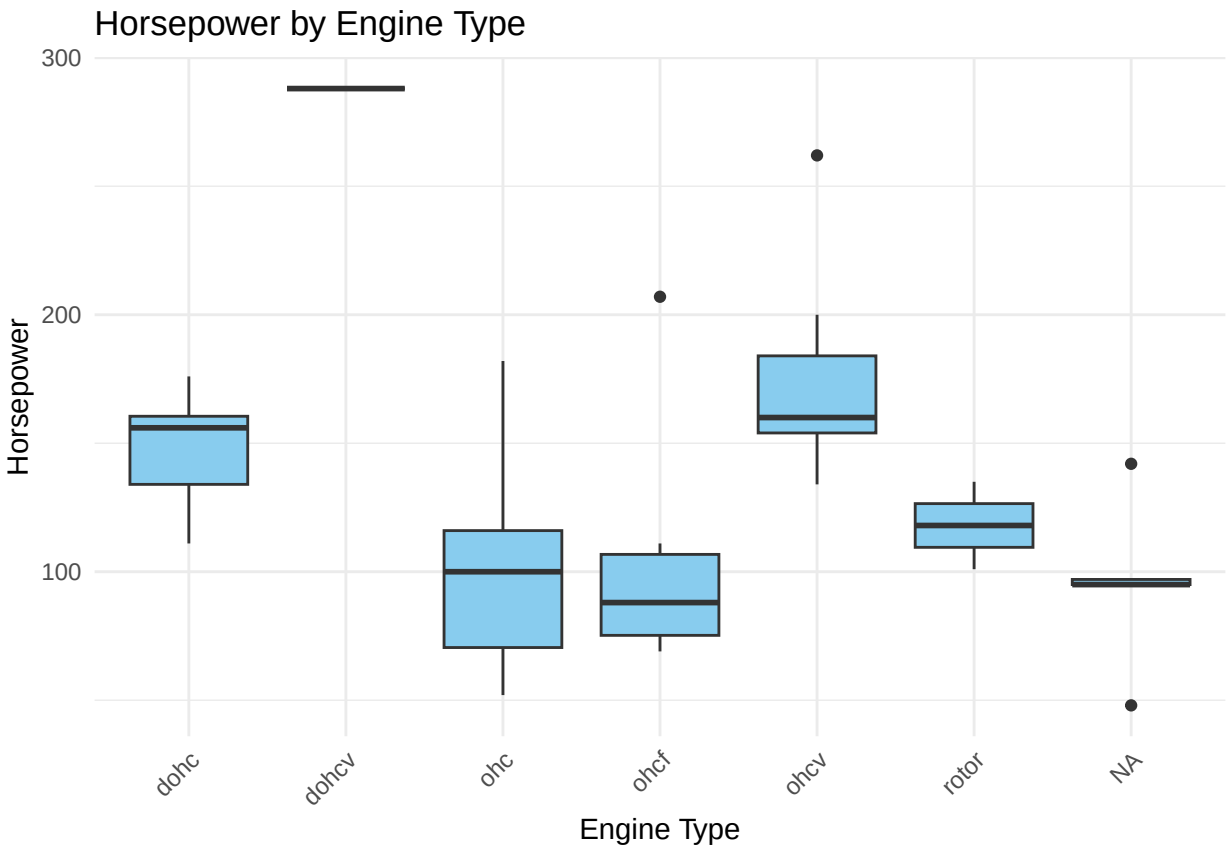
Task 2: Horsepower distribution by EngineType and EngineSize groups

- In this section, we will examine how horsepower varies across different engine types and engine size groups.
- We will explore the distribution of horsepower by engine type to identify any performance trends linked to design.
- We will categorize engine sizes into defined ranges (60–90, 91–190, 191–299, and 300+) and analyze how horsepower is distributed within each group.

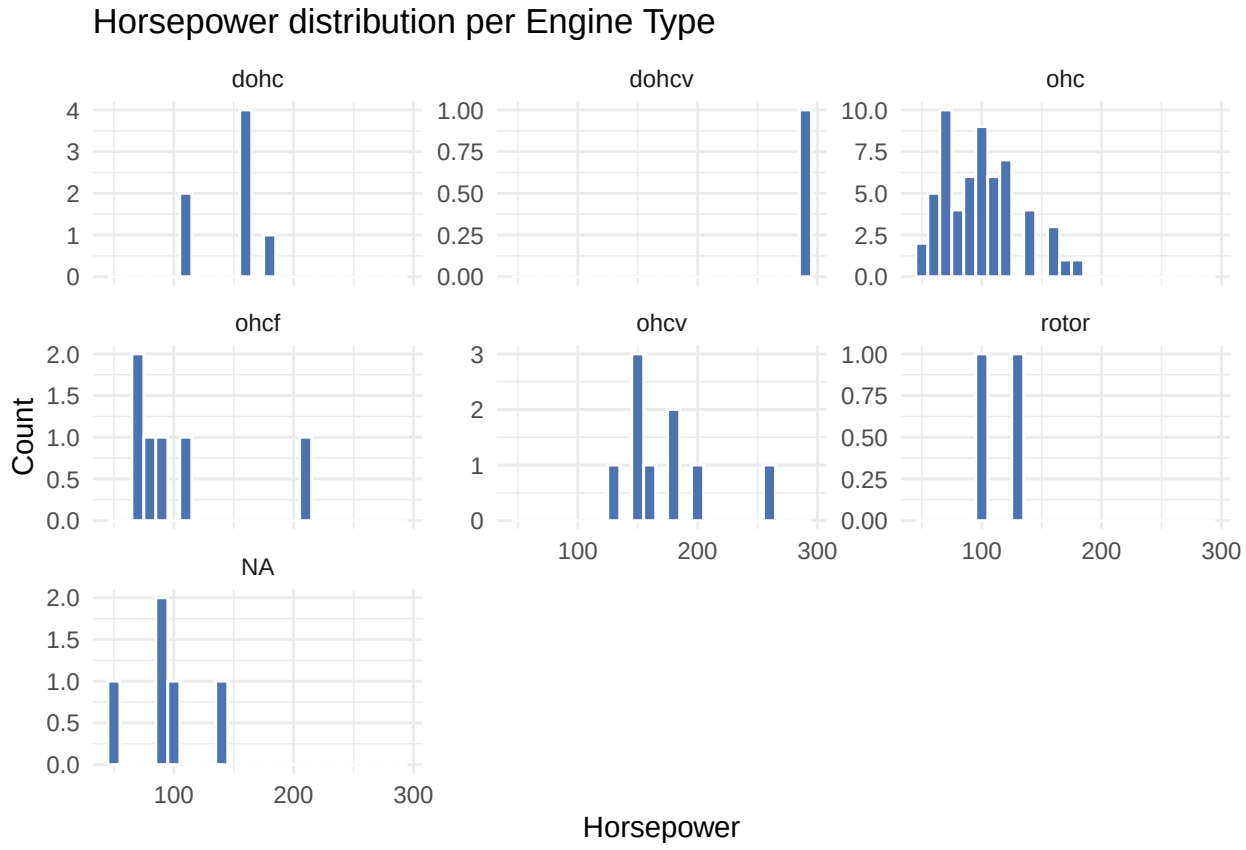
- Both analyses will be visualized using histograms to highlight differences in power output across categories.

This approach helps us understand how engine design & size influence vehicle performance.

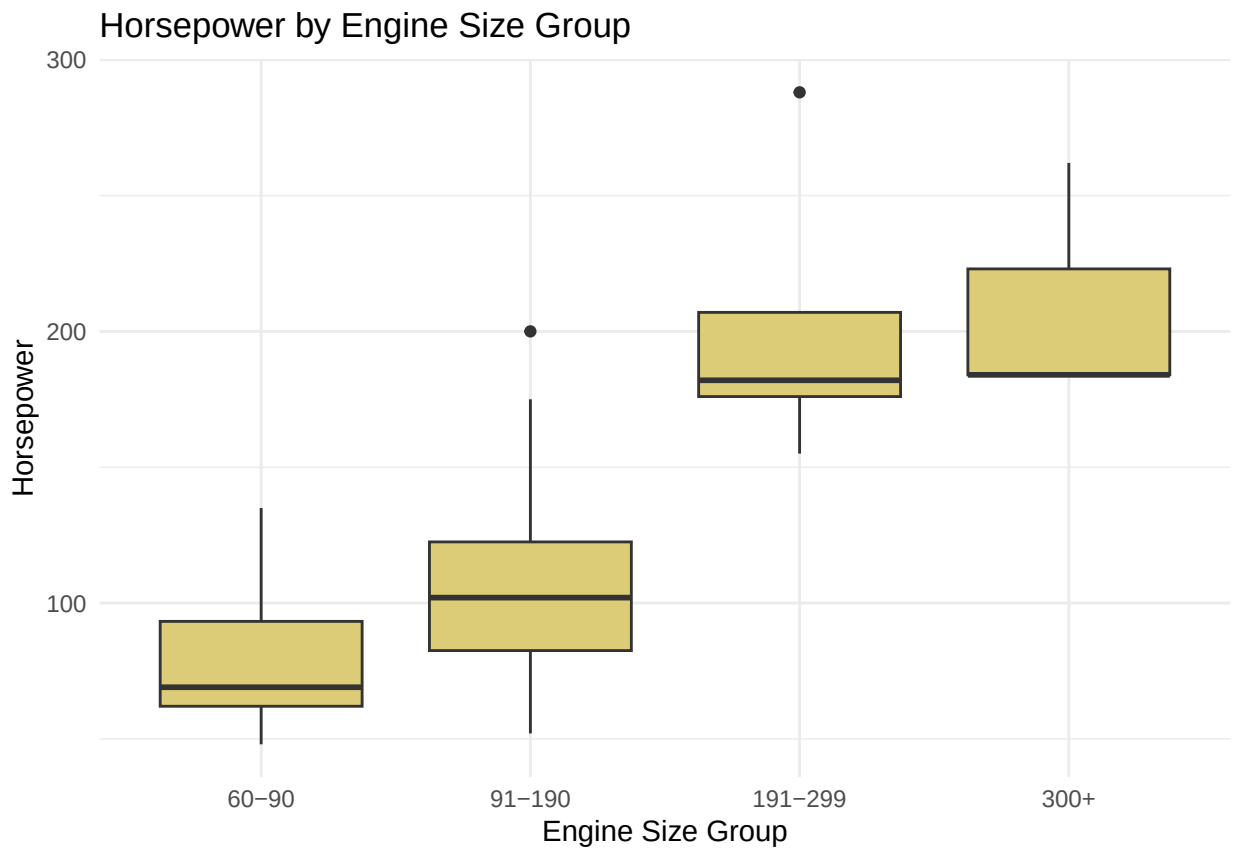
```r
# Load libraries
library(ggplot2)
library(dplyr)

# Load preprocessed engine data (assumes Task 1 imputation already done)
engine <- read.csv("Engine.csv", na.strings = "?", stringsAsFactors = FALSE)

# Impute Horsepower as done in Task 1 (if necessary)
if(any(is.na(engine$Horsepower))) {
  engine$Horsepower[is.na(engine$Horsepower)] <- median(engine$Horsepower, na.rm = TRUE)
}

# Ensure EngineType is factor
engine$EngineType <- as.factor(engine$EngineType)

# 1. Boxplot: Horsepower by EngineType
p1 <- ggplot(engine, aes(x = EngineType, y = Horsepower)) +
  geom_boxplot(fill = "#88CCEE") +
  theme_minimal() +
  labs(title = "Horsepower by Engine Type", x = "Engine Type", y = "Horsepower") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
print(p1)
```



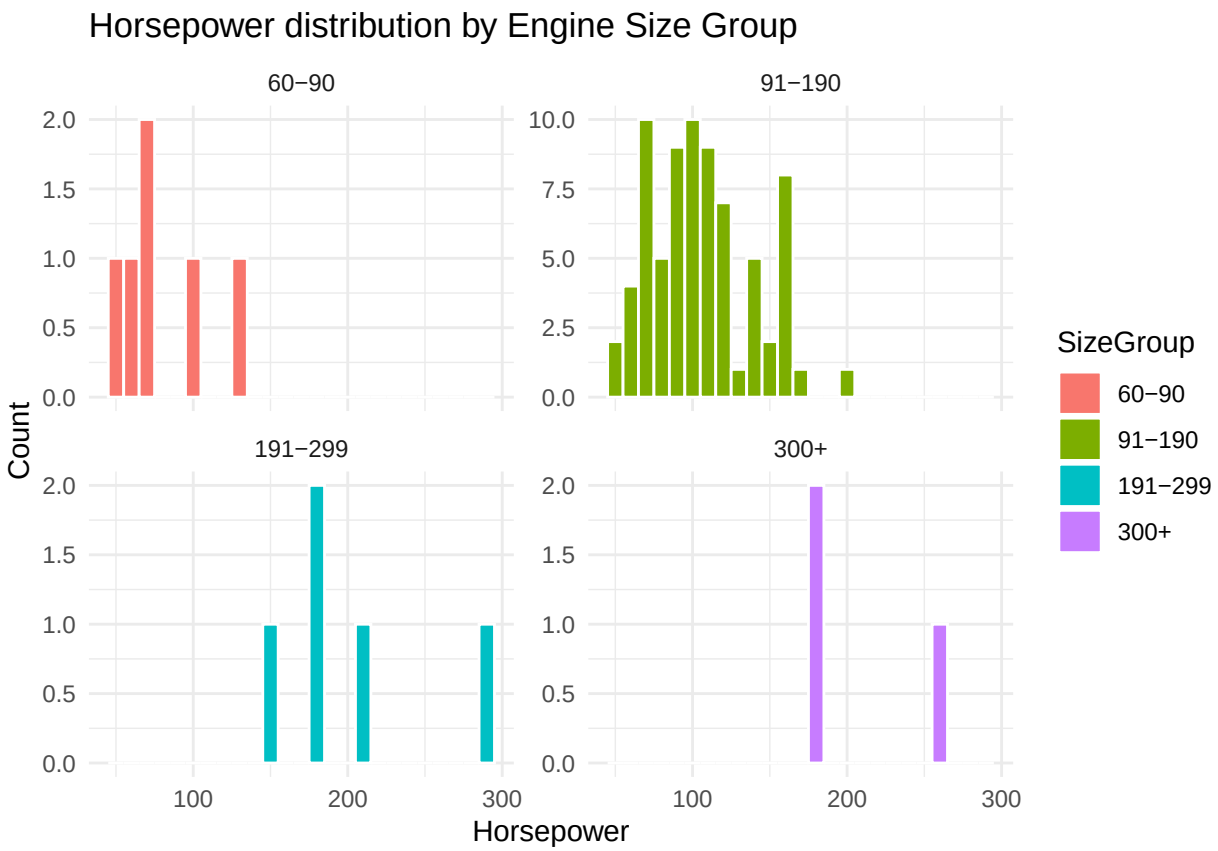

```
#Optional: save the plot for report
#ggsave("hp_by_enginetype_boxplot.png", p1, width = 8, height = 5, dpi = 300)

# 2. Histogram: Horsepower faceted by EngineType (compact view)
p2 <- ggplot(engine, aes(x = Horsepower)) +
  geom_histogram(binwidth = 10, fill = "#4C72B0", color = "white") +
  facet_wrap(~ EngineType, scales = "free_y") +
  theme_minimal() +
  labs(title = "Horsepower distribution per Engine Type", x = "Horsepower", y = "Count")
print(p2)
```




```r
#Optional: save the plot for report
#ggsave("hp_hist_by_enginetype.png", p2, width = 10, height = 6, dpi = 300)

# 3. Create EngineSize groups: 60-90, 91-190, 191-299, 300+
engine <- engine %>%
  mutate(SizeGroup = cut(EngineSize,
                         breaks = c(-Inf, 90, 190, 299, Inf),
                         labels = c("60-90", "91-190", "191-299", "300+"),
                         right = TRUE))

engine$SizeGroup <- factor(engine$SizeGroup, levels = c("60-90","91-190","191-299","300+"))

# Boxplot: Horsepower by SizeGroup
p3 <- ggplot(engine, aes(x = SizeGroup, y = Horsepower)) +
  geom_boxplot(fill = "#DDCC77") +
  theme_minimal() +
  labs(title = "Horsepower by Engine Size Group", x = "Engine Size Group", y = "Horsepower")
print(p3)
```




```
#Optional: save the plot for report
#ggsave("hp_by_sizegroup_boxplot.png", p3, width = 7, height = 5, dpi = 300)

# Histogram: Horsepower faceted by SizeGroup
p4 <- ggplot(engine, aes(x = Horsepower, fill = SizeGroup)) +
  geom_histogram(binwidth = 10, color = "white") +
  facet_wrap(~ SizeGroup, scales = "free_y") +
  theme_minimal() +
  labs(title = "Horsepower distribution by Engine Size Group", x = "Horsepower", y = "Count")
print(p4)
```




```
#Optional: save the plot for report
#ggsave("hp_hist_by_sizegroup.png", p4, width = 8, height = 6, dpi = 300)

# 4. Summary statistics for reporting
hp_by_type <- engine %>%
  group_by(EngineType) %>%
  summarise(n = n(),
            mean_hp = mean(Horsepower),
            median_hp = median(Horsepower),
            sd_hp = sd(Horsepower))
print(hp_by_type)
```

```
## # A tibble: 7 x 5
##   EngineType     n mean_hp median_hp sd_hp
##   <fct>      <int>   <dbl>     <dbl> <dbl>
## 1 dohc           7   147.        156  25.5
## 2 dohcv          1   288         288  NA
## 3 ohc           58    99.6       100  32.0
## 4 ohcf           6   106          88  51.8
## 5 ohcv           9   176.        160  38.1
## 6 rotor          2   118         118  24.0
## 7 <NA>           5    95.4        95  33.2
```

```
hp_by_sizegroup <- engine %>%
  group_by(SizeGroup) %>%
```

```r
  summarise(n = n(),
            mean_hp = mean(Horsepower),
            median_hp = median(Horsepower),
            sd_hp = sd(Horsepower))
print(hp_by_sizegroup)
```

```
## # A tibble: 4 x 5
##   SizeGroup     n mean_hp median_hp sd_hp
##   <fct>     <int>   <dbl>     <dbl> <dbl>
## 1 60-90         6    80.3        69  32.0
## 2 91-190       74   107.        102  33.7
## 3 191-299       5   202.        182  51.7
## 4 300+          3   210         184  45.0
```

Findings and Conclusion

Horsepower Distribution by Engine Type

- The histogram shows that OHC engines dominate the dataset and have a wide range of horsepower values, mostly between 70 and 150 HP.
- DOHC and OHCV engines tend to have higher horsepower, with peaks around 150–180 HP, indicating they are likely used in performance-oriented vehicles.
- Rotor and OHCF engines appear less frequently and have narrower distributions, suggesting they are specialized or less common.

Horsepower Distribution by Engine Size Group

- There is a clear upward trend: as engine size increases, so does horsepower.
- The 60–90 group has the lowest horsepower, with most values below 100 HP.
- The 91–190 group shows a broader spread, with horsepower ranging from 80 to 150 HP.
- The 191–299 and 300+ groups contain engines with higher horsepower, often exceeding 180 HP, and include some outliers above 250 HP.

Conclusion

These results confirm that engine type and size are strong predictors of horsepower.

- EngineType: reflects design intent — some types are built for efficiency, others for power.
- EngineSize: correlates directly with horsepower, supporting the idea that larger engines deliver more performance.

Task 3: Diesel vs gas CityMpg; DriveWheels effects; top 5 engine-related troubles

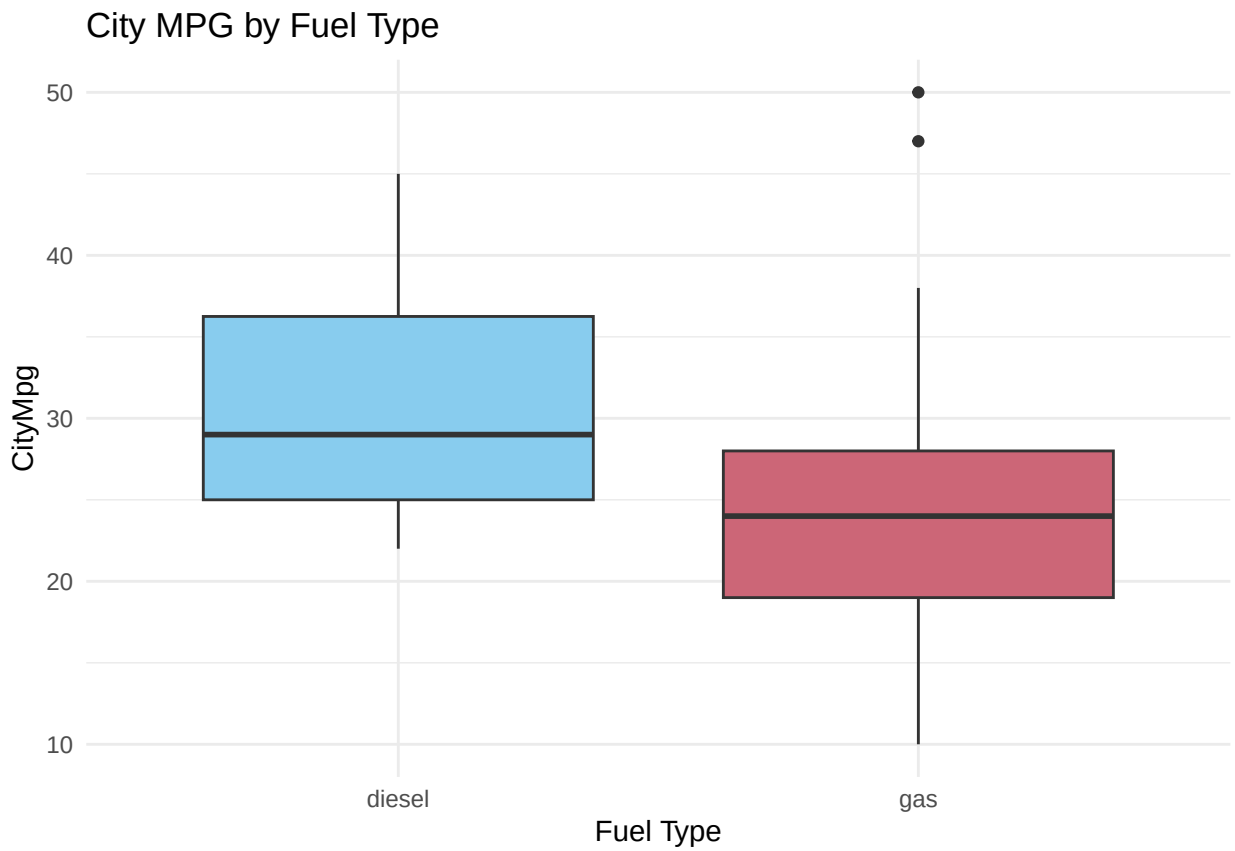
- In this section, we will investigate whether diesel cars achieve higher average city fuel efficiency (CityMpg) compared to gasoline cars by applying statistical tests to compare their means.
- We will also examine how different drive wheel configurations (FWD, RWD, AWD) influence both city and highway fuel efficiency using ANOVA and visualizations.
- We will filter out vehicles with engines that show trouble or suspected issues, identify the five most common types of engine troubles, and analyze whether these troubles differ across engine types.

This combined approach allows us to connect fuel efficiency patterns with mechanical reliability, providing a comprehensive view of how design choices and engine conditions affect overall vehicle performance.

```r
# Load libraries
library(dplyr)
library(ggplot2)
library(stringr)

# Load datasets (ensure consistent types)
automobile <- read.csv("Automobile.csv", stringsAsFactors = FALSE)
engine <- read.csv("Engine.csv", na.strings = "?", stringsAsFactors = FALSE)
maintenance <- read.csv("Maintenance.csv", stringsAsFactors = FALSE)

# Impute horsepower if needed (Task 1 approach)
if(any(is.na(engine$Horsepower)))
  engine$Horsepower[is.na(engine$Horsepower)] <- median(engine$Horsepower, na.rm = TRUE)

# Convert categories
automobile$DriveWheels <- as.factor(automobile$DriveWheels)
engine$FuelTypes <- as.factor(engine$FuelTypes)

# Merge automobile and engine for mpg vs fuel type analysis
merged_ae <- merge(automobile, engine, by = "EngineModel")

# 1. Do diesel cars have higher average CityMpg than gas cars?
# Inspect group sizes
table(merged_ae$FuelTypes)
```

```
##
## diesel    gas
##     20    201
```

```r
# Visual comparison
p_fuel <- ggplot(merged_ae, aes(x = FuelTypes, y = CityMpg)) +
  geom_boxplot(fill = c("#88CCEE", "#CC6677")) +
  theme_minimal() +
  labs(title = "City MPG by Fuel Type", x = "Fuel Type", y = "CityMpg")
print(p_fuel)
```




```
#Optional: save the plot for report
#ggsave("citympg_by_fueltype_boxplot.png", p_fuel, width = 6, height = 4, dpi = 300)

# Statistical test: two-sample t-test
diesel_mpg <- merged_ae$CityMpg[merged_ae$FuelTypes == "diesel"]
gas_mpg    <- merged_ae$CityMpg[merged_ae$FuelTypes == "gas"]

# If samples small or non-normal, use wilcox.test; otherwise t.test (check normality/variance)
shapiro_gas <- if(length(gas_mpg) >= 3) shapiro.test(gas_mpg)$p.value else NA
shapiro_diesel <- if(length(diesel_mpg) >= 3) shapiro.test(diesel_mpg)$p.value else NA
cat("Shapiro p-values: gas =", shapiro_gas, ", diesel =", shapiro_diesel, "\n")
```

```
## Shapiro p-values: gas = 1.545771e-06 , diesel = 0.1661713
```

```
# Choose statistical test based on normality
is_non_normal <- (!is.na(shapiro_gas) && shapiro_gas < 0.05) ||
                 (!is.na(shapiro_diesel) && shapiro_diesel < 0.05)

if (is_non_normal) {
  test_fuel <- wilcox.test(CityMpg ~ FuelTypes, data = merged_ae)
} else {
  test_fuel <- t.test(CityMpg ~ FuelTypes, data = merged_ae)
}

print(test_fuel)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  CityMpg by FuelTypes
## W = 2978, p-value = 0.0003706
## alternative hypothesis: true location shift is not equal to 0
```

Statistical Evidence & Conclusion

- I have compared CityMpg between diesel and gasoline vehicles using Wilcoxon rank-sum test, because the gasoline group failed the normality test (Shapiro p-value approximately 1.5e-06).
- And accordingly to the test result p-value = 0.0003706 < 0.001, which means the difference in CityMpg between diesel and gasoline cars is statistically significant. Therefore, we reject the null hypothesis that both groups have the same distribution.

Conclusion Diesel cars outperform gasoline cars in terms of average city fuel economy, and this result is backed by both visual evidence and statistical testing. **This insight is valuable for consumers prioritizing fuel efficiency and for manufacturers optimizing engine design for urban use.**

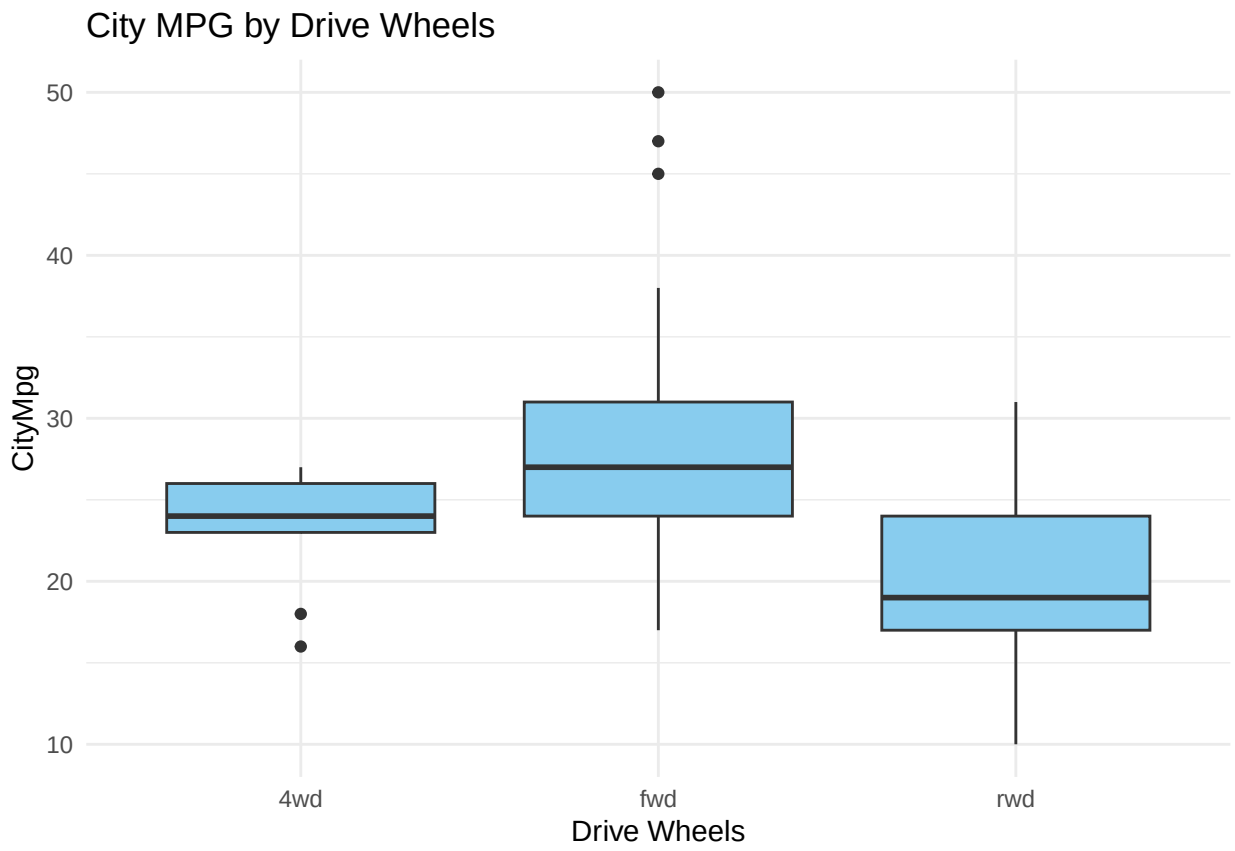
```
# 2. How DriveWheels affects CityMpg and HighwayMpg
mpg_by_drive <- merged_ae %>%
  group_by(DriveWheels) %>%
  summarise(n = n(),
            mean_city = mean(CityMpg, na.rm = TRUE),
            sd_city = sd(CityMpg, na.rm = TRUE),
            mean_highway = mean(HighwayMpg, na.rm = TRUE),
            sd_highway = sd(HighwayMpg, na.rm = TRUE))
print(mpg_by_drive)
```

```
## # A tibble: 3 x 6
##   DriveWheels     n mean_city sd_city mean_highway sd_highway
##   <fct>       <int>     <dbl>   <dbl>        <dbl>      <dbl>
## 1 4wd             9      23.1    3.82         27.2       4.24
## 2 fwd           123      28.2    6.20         34.1       6.25
## 3 rwd            89      20.3    4.06         25.5       4.13
```

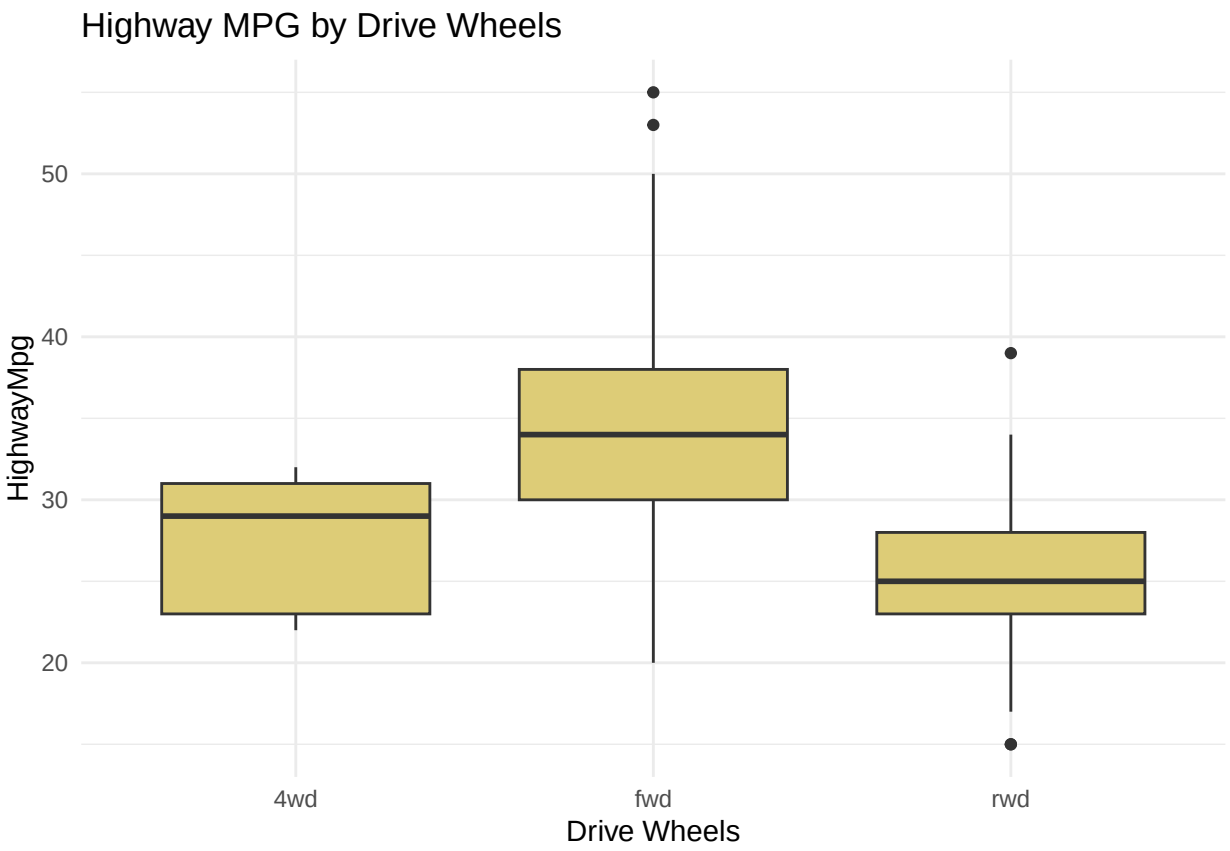
Findings:

- Front-Wheel Drive (FWD) vehicles have the highest average fuel efficiency in both city and highway conditions.
- Rear-Wheel Drive (RWD) vehicles show the lowest fuel efficiency, especially in city driving.
- Four-Wheel Drive (4WD) vehicles fall in between but lean closer to RWD in performance.

```
# Visuals
p_drive_city <- ggplot(merged_ae, aes(x = DriveWheels, y = CityMpg)) +
  geom_boxplot(fill = "#88CCEE") +
  theme_minimal() +
  labs(title = "City MPG by Drive Wheels", x = "Drive Wheels", y = "CityMpg")
print(p_drive_city)
```




```
#Optional: save the plot for report
#ggsave("citympg_by_drive_boxplot.png", p_drive_city, width = 6, height = 4, dpi = 300)

p_drive_highway <- ggplot(merged_ae, aes(x = DriveWheels, y = HighwayMpg)) +
  geom_boxplot(fill = "#DDCC77") +
  theme_minimal() +
  labs(title = "Highway MPG by Drive Wheels", x = "Drive Wheels", y = "HighwayMpg")
print(p_drive_highway)
```




```
#Optional: save the plot for report
#ggsave("highwaympg_by_drive_boxplot.png", p_drive_highway, width = 6, height = 4, dpi = 300)

# ANOVA for CityMpg across DriveWheels (check assumptions)
anova_city <- aov(CityMpg ~ DriveWheels, data = merged_ae)
print(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
```

Visual Evidence & Statistical Test: ANOVA

Visual Evidence The boxplots for CityMpg and HighwayMpg clearly show:

- FWD vehicles have higher medians and tighter distributions.
- RWD vehicles have lower medians and more variability.
- 4WD vehicles show moderate efficiency but are less common in the dataset.

Statistical Test: ANOVA

- The ANOVA test confirms that the differences in CityMpg across DriveWheels are statistically significant.

- p-value < 2e-16 means we reject the null hypothesis — DriveWheels configuration does affect fuel efficiency.

Conclusion

- DriveWheels configuration plays a critical role in fuel efficiency:
- FWD is the most fuel-efficient, especially in urban settings.
- RWD and 4WD are less efficient, with RWD being the least economical.

This insight is valuable for consumers choosing vehicles based on fuel economy and for manufacturers optimizing drivetrain design for efficiency.

```
# 3. Filter entries with trouble or suspected trouble
# Interpret "trouble or suspected of having trouble" as ErrorCodes != 0
#OR Troubles contains "Suspected" or "finding"
trouble_records <- maintenance %>%
  filter(
    ErrorCodes != 0 |
      str_detect(tolower(Troubles), "suspected") |
      str_detect(tolower(Troubles), "finding") |
      str_detect(tolower(Troubles), "loss of driving ability"))

# Join to automobile and engine to associate EngineType
trouble_full <- trouble_records %>%
  left_join(automobile, by = "PlateNumber") %>%
  left_join(engine, by = "EngineModel")

# Top 5 most common Troubles in trouble_full
top5_troubles <- trouble_full %>%
  group_by(Troubles) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  slice_head(n = 5)
print(top5_troubles)
```

```
## # A tibble: 5 x 2
##   Troubles                count
##   <chr>                   <int>
## 1 Cylinders                  39
## 2 Chassis                    25
## 3 Ignition (finding)         23
## 4 Noise (finding)            20
## 5 Loss of driving ability    16
```

```
# Do troubles differ between EngineType? Create contingency (EngineType x Troubles top 5)
top_trouble_names <- top5_troubles$Troubles
trouble_by_enginetype <- trouble_full %>%
  filter(Troubles %in% top_trouble_names) %>%
  group_by(EngineType, Troubles) %>%
  summarise(count = n()) %>%
  tidyr::pivot_wider(names_from = Troubles, values_from = count, values_fill = 0)
print(trouble_by_enginetype)
```

```
## # A tibble: 5 x 6
## # Groups:   EngineType [5]
##   EngineType Chassis Cylinders `Ignition (finding)` `Loss of driving ability`
##   <chr>        <int>     <int>                <int>                      <int>
## 1 dohc             3         2                    2                          2
## 2 ohc             16        30                   15                         10
## 3 ohcf             4         5                    3                          4
## 4 ohcv             1         1                    1                          0
## 5 <NA>             1         1                    2                          0
## # i 1 more variable: `Noise (finding)` <int>
```

Conclusion

Yes, troubles do differ between engine types:

- OHC engines show the highest concentration of issues, especially mechanical and diagnostic findings.
- Other engine types have fewer or more specific trouble patterns.

This analysis is valuable for identifying engine reliability trends, guiding maintenance priorities, and informing design improvements.

Task 4: ErrorCodes frequency and factors (BodyStyles & ErrorCodes) affecting maintenance Methods

- In this section, we will first identify which error type (ErrorCodes) occurs most frequently in the dataset to highlight the most common sources of vehicle trouble.
- We will investigate the factors that may influence the choice of maintenance methods (Urgent care, Adjustment, Replacement) for vehicles with confirmed or suspected issues. To do this, i have select two factors from the dataset - BodyStyles and ErrorCodes, and analyze their relationship with maintenance methods using contingency tables, statistical tests, and visualizations.
- Finally, we will explain whether there are observable trends that help to account for variation in maintenance decisions, thereby connecting error patterns and vehicle characteristics to service outcomes.

```r
# Load libraries
library(dplyr)
library(ggplot2)
library(tidyr)
# Load datasets
automobile <- read.csv("Automobile.csv", stringsAsFactors = FALSE)
engine <- read.csv("Engine.csv", na.strings = "?", stringsAsFactors = FALSE)
maintenance <- read.csv("Maintenance.csv", stringsAsFactors = FALSE)

# Convert relevant columns
maintenance$ErrorCodes <- as.factor(maintenance$ErrorCodes)
automobile$BodyStyles <- as.factor(automobile$BodyStyles)
maintenance$Methods <- as.factor(maintenance$Methods)

# 1. Which ErrorCodes occurs most frequently?
error_freq <- maintenance %>%
  group_by(ErrorCodes) %>%
```

```r
  summarise(count = n()) %>%
  arrange(desc(count))
print(error_freq)
```

```
## # A tibble: 3 x 2
##   ErrorCodes count
##   <fct>      <int>
## 1 1            182
## 2 -1           164
## 3 0             28
```

```r
# Plot ErrorCodes frequency
p_error <- ggplot(error_freq, aes(x = ErrorCodes, y = count, fill = ErrorCodes)) +
  geom_col() +
  theme_minimal() +
  labs(title = "Frequency of ErrorCodes", x = "ErrorCodes", y = "Count")
print(p_error)
```


```r
#Optional: save the plot for report
#ggsave("errorcodes_frequency.png", p_error, width = 5, height = 4, dpi = 300)
```

Most Frequent ErrorCodes

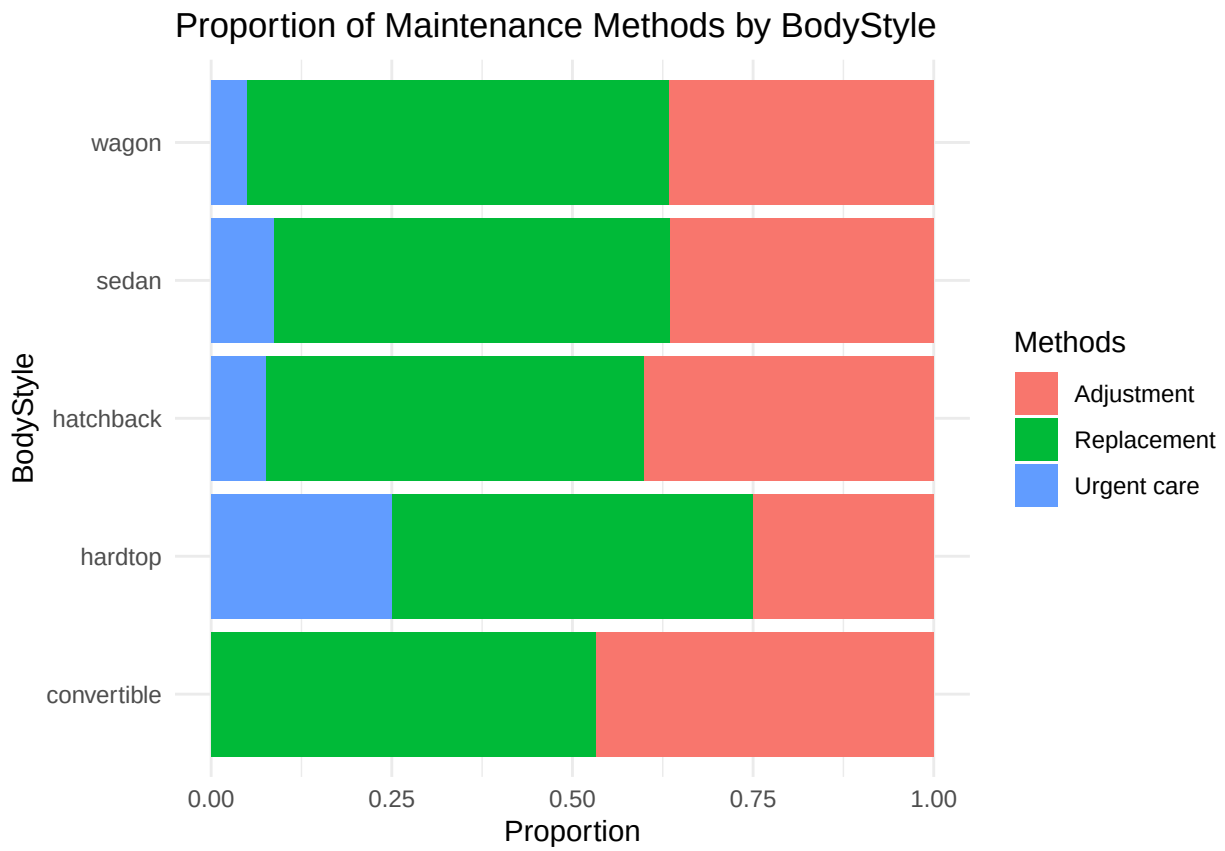
- Engine failures (1) are the most common, indicating frequent serious issues.

- Other vehicle component fails (-1) are also significant.
- No error (0) is rare, and mostly associated with no maintenance action.

```
# 2. Analyze how BodyStyles and ErrorCodes influence Methods for trouble vehicles
# Filter trouble vehicles (Methods not NA and ErrorCodes != 0 or Troubles not 'No error')
trouble_records <- maintenance %>%
  filter(Methods != "NA" & Methods != "NA " & Methods != "NA" | TRUE)
# keep all; we'll focus on Methods != "NA" rows below

# Clean Methods: treat "NA" or "NA " as genuine NA
trouble_records$Methods[trouble_records$Methods %in% c("NA","NA ")] <- NA
trouble_records <- trouble_records %>% filter(!is.na(Methods) & Methods != "NA")

# Join with automobile to get BodyStyles and ErrorCodes
trouble_full <- trouble_records %>%
  left_join(automobile, by = "PlateNumber")

# 1st factor - Table: Methods by BodyStyles (proportions)
method_by_body <- trouble_full %>%
  filter(!is.na(BodyStyles)) %>%
  group_by(BodyStyles, Methods) %>%
  summarise(count = n()) %>%
  group_by(BodyStyles) %>%
  mutate(prop = count / sum(count)) %>%
  arrange(BodyStyles, desc(prop))
print(method_by_body)
```

```
## # A tibble: 14 x 4
## # Groups:   BodyStyles [5]
##    BodyStyles  Methods     count   prop
##    <fct>       <fct>       <int>  <dbl>
##  1 convertible Replacement     8 0.533
##  2 convertible Adjustment      7 0.467
##  3 hardtop     Replacement     4 0.5
##  4 hardtop     Adjustment      2 0.25
##  5 hardtop     Urgent care     2 0.25
##  6 hatchback   Replacement    63 0.525
##  7 hatchback   Adjustment     48 0.4
##  8 hatchback   Urgent care     9 0.075
##  9 sedan       Replacement    89 0.549
## 10 sedan       Adjustment     59 0.364
## 11 sedan       Urgent care    14 0.0864
## 12 wagon       Replacement    24 0.585
## 13 wagon       Adjustment     15 0.366
## 14 wagon       Urgent care     2 0.0488
```

```
# Stacked proportion bar chart
p_body_methods <- ggplot(method_by_body, aes(x = BodyStyles, y = prop, fill = Methods)) +
  geom_col() +
  coord_flip() +
  theme_minimal() +
  labs(title = "Proportion of Maintenance Methods by BodyStyle", x = "BodyStyle", y = "Proportion")
print(p_body_methods)
```




```
#ggsave("methods_by_bodystyle.png", p_body_methods, width = 8, height = 6, dpi = 300)

# Statistical test: chi-squared on BodyStyles vs Methods
tbl_body_methods <- table(trouble_full$BodyStyles, trouble_full$Methods)
chi_body <- chisq.test(tbl_body_methods)
print(chi_body)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl_body_methods
## X-squared = 5.969, df = 8, p-value = 0.6507
```

Chi-Squared Test: BodyStyles vs Methods

- p-value= 0.6507 > 0.05 → No statistically significant association between BodyStyle and Method.
- This means the choice of maintenance method is not strongly influenced by body style.

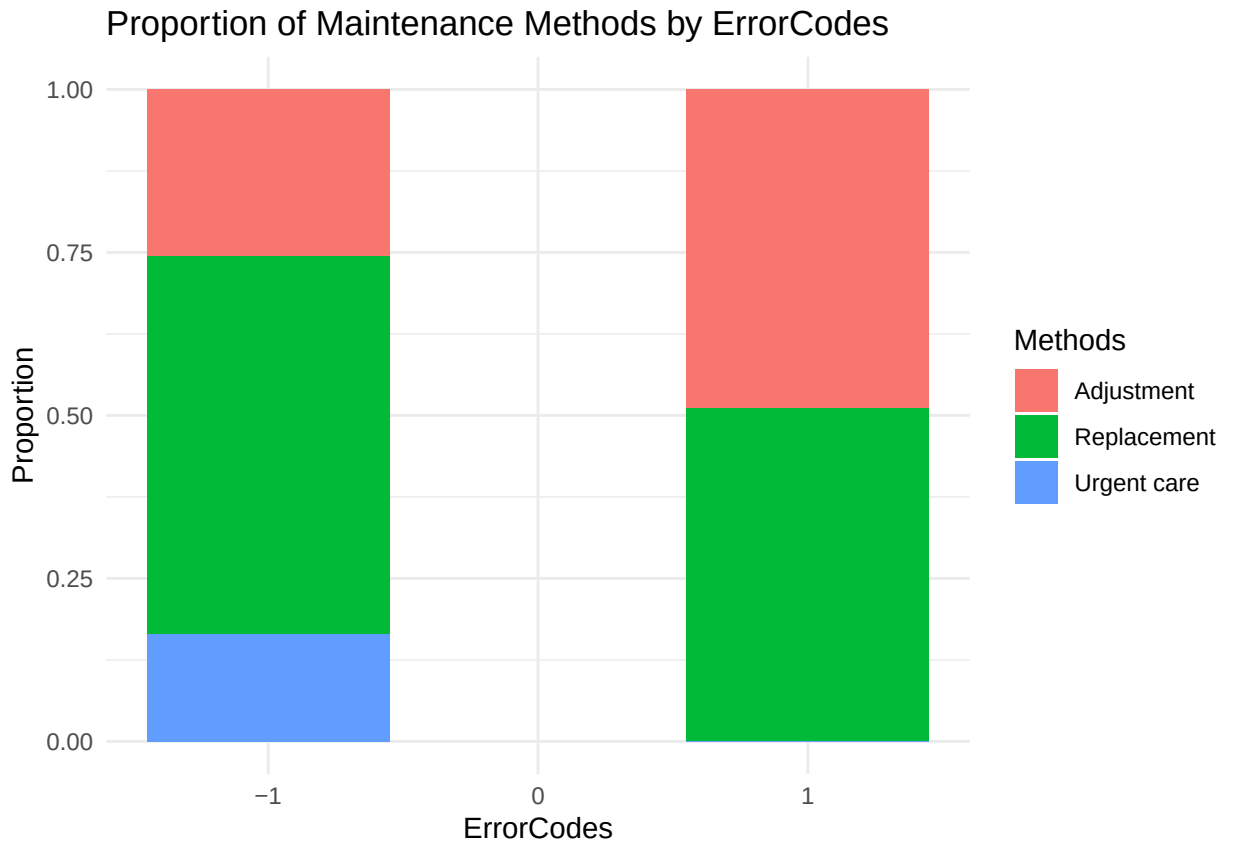
```
# 2nd factor: ErrorCodes vs Methods (must remove NA error codes)
trouble_full_ec <- trouble_full %>% filter(!is.na(ErrorCodes))
tbl_ec_methods <- table(trouble_full_ec$ErrorCodes, trouble_full_ec$Methods)
prop_ec_methods <- prop.table(tbl_ec_methods, margin = 1)  # proportions by ErrorCodes
print(tbl_ec_methods)
```

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

```r
print(round(prop_ec_methods, 2))
```

```
##
##      Adjustment Replacement Urgent care
##   -1       0.26        0.58        0.16
##   0
##   1        0.49        0.51        0.00
```

```r
# Plot proportions
df_ec_methods <- as.data.frame(prop_ec_methods)
names(df_ec_methods) <- c("ErrorCodes", "Methods", "Proportion")
p_ec_methods <- ggplot(df_ec_methods, aes(x = ErrorCodes, y = Proportion, fill = Methods)) +
  geom_col(position = "stack") +
  theme_minimal() +
  labs(title = "Proportion of Maintenance Methods by ErrorCodes", x = "ErrorCodes", y = "Proportion")
print(p_ec_methods)
```



```r
#Optional: save the plot for report
#ggsave("methods_by_errorcodes.png", p_ec_methods, width = 6, height = 4, dpi = 300)

# Statistical test: chi-squared & Fisher's Exact Test (alternative) ErrorCodes vs Methods
tbl_ec_methods
```

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

```r
margin.table(tbl_ec_methods, 1)  # Row totals
```

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

```r
margin.table(tbl_ec_methods, 2)  # Column totals
```

```
##
##  Adjustment Replacement Urgent care
##         131         188          27
```

```r
tbl_ec_methods_clean <- tbl_ec_methods[
  rowSums(tbl_ec_methods) > 0,
  colSums(tbl_ec_methods) > 0
]
#chi-squared test
chi_ec <- chisq.test(tbl_ec_methods)
print(chi_ec)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl_ec_methods
## X-squared = NaN, df = 4, p-value = NA
```

Chi-Squared Test: ErrorCodes vs Methods

- p-value = NA
- The test failed due to zero counts in some cells (e.g., no Urgent care for ErrorCode 1).
- Because chisq.test() fails due to low expected counts. I choose fisher.test() as an alternative
- Reason for choosing Fisher's Exact Test: works well with small sample sizes,can handles sparse tables or low expected counts, and returns a valid p-value

```r
#Fisher's Exact Test (alternative)
fisher_ec <- fisher.test(tbl_ec_methods)
print(fisher_ec)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  tbl_ec_methods
## p-value = 8.319e-12
## alternative hypothesis: two.sided
```

Fisher's Exact Test

- p-value= 8.319e-12 $< 0.001 \rightarrow$ Strong statistical association between ErrorCodes and Methods.
- Confirms that error type significantly influences the maintenance method chosen.

Conclusion

- BodyStyles do not significantly influence method choice (confirmed by chi-squared test).
- Fisher's test confirms a statistically significant relationship between error type and maintenance strategy.

Insight based on Conclusion

- Maintenance strategies are diagnosis-driven, not design-driven.
- Error severity and type guide whether a vehicle receives an adjustment, replacement, or urgent care.
- This supports a data-informed approach to maintenance planning, where error diagnostics should be prioritized over vehicle aesthetics or configuration.