Data Cleaning

ANLY-511-04 Group 3

12/06/2022

```
# load libraries for this assignment
library(tidyverse)
library(ggplot2)
library(readxl)
library(knitr)
library(kableExtra)
```

Functions for Use

```
calculate_proportions <- function(input_df, threshold) {

  # create data frame for printing proportions of of NA values
  df <- data.frame(colnames(input_df), round(colMeans(is.na(input_df)), 5), row.names = NULL)
  colnames(df) <- c('Column Name', 'Proportion of Values NA')
  df <- df[order(df$`Proportion of Values NA`, decreasing = TRUE),]
  df_table <- df[df$`Proportion of Values NA` > threshold,]

  return (df_table)
}
```

```
print_table <- function(input_table) {
  knitr::kable(input_table, row.names = FALSE) %>% kable_styling(bootstrap_options = "striped", full_width = F, p
osition = "center")
}
```

```
print_shape <- function(input_df) {
  print(paste('Our dataset shape is now: (', nrow(input_df), ', ', ncol(input_df), ')', sep = ''))
}
```

Read in the Data

Let's first read in the data. We have five datasets to read in, all by the naming convention of `data/cardataXX.xlsx` , where `XX ∈ {18, 19, 20, 21, 22}` . We can also observe the *shape* of each dataset, indicated with the following syntax: `(number of rows, number of columns)` .

```
# load in the data
cardata2018 <- read_excel('../data/cardata2018.xlsx')
cardata2019 <- read_excel('../data/cardata2019.xlsx')
cardata2020 <- read_excel('../data/cardata2020.xlsx')
cardata2021 <- read_excel('../data/cardata2021.xlsx')
cardata2022 <- read_excel('../data/cardata2022.xlsx')
```

Dataset Name	Shape
data/cardata2018.xlsx	(4727, 67)
data/cardata2019.xlsx	(4719, 67)
data/cardata2020.xlsx	(4450, 67)
data/cardata2021.xlsx	(4265, 67)
data/cardata2022.xlsx	(4455, 67)

The five datasets come from the same source, but we want to ensure that they are compatible with one-another. The best way to check this is to ensure that the columns are the same across all datasets. If they are, we can easily append the five datasets together, by row, in order to create one, larger dataset.

```
# extract column names for each dataset
colnames2018 <- colnames(cardata2018)
colnames2019 <- colnames(cardata2019)
colnames2020 <- colnames(cardata2020)
colnames2021 <- colnames(cardata2021)
colnames2022 <- colnames(cardata2022)

# ensure that the column names are all the same across all datasets
if ( mean(colnames2018 == colnames2019) == 1 &
     mean(colnames2018 == colnames2020) == 1 &
     mean(colnames2018 == colnames2021) == 1 &
     mean(colnames2018 == colnames2022) == 1 ) {
  print('The column names are the same and in the same order across all five datasets.')
} else {
  print('The column names are NOT the same and in the same order across all five datasets.')
}
```

```
## [1] "The column names are the same and in the same order across all five datasets."
```

Let’s append these datasets together since we know now that they have the same structure and column names.

```
# bind datasets together by row
cardata <- rbind(cardata2018, cardata2019, cardata2020, cardata2021, cardata2022)
print_shape(cardata)
```

```
## [1] "Our dataset shape is now: (22616, 67)"
```

There are also some columns that we are not concerned with for the purpose of our analysis, so we will drop those before proceeding with the cleaning phase. These columns are:

- `Test Vehicle ID`
- `Test Veh Configuration #`
- `Actual Tested Testgroup`
- `Engine Code`
- `Transmission Overdrive Code`
- `Transmission Overdrive Desc`
- `Shift Indicator Light Use Cd`
- `Shift Indicator Light Use Desc`
- `Test Number`
- `Test Originator`
- `Analytically Derived FE?`
- `ADFE Test Number`
- `Test Procedure Cd`
- `Test Procedure Description`
- `Test Fuel Type Cd`
- `Test Category`
- `FE_UNIT`
- `FE Bag 1`
- `FE Bag 2`
- `FE Bag 3`
- `FE Bag 4`

```
# define columns to remove
remove <- c("Test Vehicle ID", "Test Veh Configuration #", "Actual Tested Testgroup",
            "Engine Code", "Transmission Overdrive Code", "Transmission Overdrive Desc",
            "Shift Indicator Light Use Cd", "Shift Indicator Light Use Desc", "Test Number",
            "Test Originator", "Analytically Derived FE?", "ADFE Test Number",
            "Test Procedure Cd", "Test Procedure Description", "Test Fuel Type Cd",
            "Test Category", "FE_UNIT", "FE Bag 1",
            "FE Bag 2", "FE Bag 3", "FE Bag 4")

# remove columns and report new shape
cardata <- cardata[, !( colnames(cardata) %in% remove)]
print_shape(cardata)
```

```
## [1] "Our dataset shape is now: (22616, 46)"
```

Split Data into Two

Since one large are of focus of ours is comparing electric vehicles to non-electric vehicles, we will split our dataset into two groups: one containing only electric vehicles and one containing only non-electric vehicles. This will allow us to make easy comparisons both between datasets and within datasets.

```
# look at the unique fuel types
unique(cardata$`Test Fuel Type Description`)
```

```
##  [1] "Tier 2 Cert Gasoline"
##  [2] "Cold CO Premium (Tier 2)"
##  [3] "Federal Cert Diesel 7-15 PPM Sulfur"
##  [4] "Electricity"
##  [5] "Cold CO Regular (Tier 2)"
##  [6] "E85 (85% Ethanol 15% EPA Unleaded Gasoline)"
##  [7] "Tier 3 E10 Regular Gasoline (9 RVP @Low Alt.)"
##  [8] "Hydrogen 5"
##  [9] "Cold CO E10 Premium Gasoline (Tier 3)"
## [10] "Tier 3 E10 Premium Gasoline (9 RVP @Low Alt.)"
## [11] "CARB Phase II Gasoline"
## [12] "Cold CO Diesel 7-15 ppm Sulfur"
## [13] "CARB LEV3 E10 Regular Gasoline"
```

Observing the unique fuel types above, we will split the dataset on cars with `Test Fuel Type Description` labeled as `Electricity` or `Hydrogen 5`, leaving the remaining fuel types for the other dataset.

```
# split the dataset in two
cardata_electric <- cardata[cardata$`Test Fuel Type Description` %in% c('Electricity', 'Hydrogen 5'),]
cardata_nonelectric <- cardata[!(cardata$`Test Fuel Type Description` %in% c('Electricity', 'Hydrogen 5')),]
```

Address Missing Values by Column

Now, with the columns that we do have, we can observe how many `NA` values exist in each column. We will set a threshold of **10%**; a column with more than 10 percent of its values being `NA` expresses a large number of missing values in our eyes and we will remove it from our analysis. Below, we can see all columns expressing more than 10% of their values as `NA`.

```
# print the proportions of NA values in each column using a threshold of 10%
electric_table <- calculate_proportions(cardata_electric, 0.10)
nonelectric_table <- calculate_proportions(cardata_nonelectric, 0.10)
print_table(electric_table)
```

Column Name	Proportion of Values NA
ADFE Total Road Load HP	1.00000
ADFE Equiv. Test Weight (lbs.)	1.00000
ADFE N/V Ratio	1.00000
THC (g/mi)	1.00000
CO2 (g/mi)	1.00000
PM (g/mi)	1.00000
CH4 (g/mi)	1.00000
N2O (g/mi)	1.00000
Averaging Group ID	0.96811
CO (g/mi)	0.93850
Averaging Weighting Factor	0.92711
NOx (g/mi)	0.90774

Column Name	Proportion of Values NA
# of Cylinders and Rotors	0.86105
Aftertreatment Device Cd	0.86105
Aftertreatment Device Desc	0.86105
DT-Inertia Work Ratio Rating	0.85991
DT-Absolute Speed Change Ratg	0.85991
DT-Energy Economy Rating	0.85991
RND_ADJ_FE	0.15034

```
print_table(nonelectric_table)
```

Column Name	Proportion of Values NA
Averaging Group ID	0.98114
Averaging Weighting Factor	0.97249
ADFE Total Road Load HP	0.88683
ADFE Equiv. Test Weight (lbs.)	0.88683
ADFE N/V Ratio	0.88683
PM (g/mi)	0.81990
N2O (g/mi)	0.47796
CH4 (g/mi)	0.15457
NOx (g/mi)	0.10907

We can then remove these columns from each respective dataset. Note that the `RND_ADJ_FE` and `CO2 (g/mi)` variables are ones that we'd like to work with, however, so we will keep those regardless of their missing value counts.

```
# remove high-probability NA columns from data frame
remove_electric <- electric_table$`Column Name`
remove_electric <- remove_electric[!(remove_electric %in% c('RND_ADJ_FE', 'CO2 (g/mi)'))]
remove_nonelectric <- nonelectric_table$`Column Name`

# remove columns and report new shape
cardata_electric <- cardata_electric[, !( colnames(cardata_electric) %in% remove_electric)]
cardata_nonelectric <- cardata_nonelectric[, !( colnames(cardata_nonelectric) %in% remove_nonelectric)]
print_shape(cardata_electric)
```

```
## [1] "Our dataset shape is now: (878, 29)"
```

```
print_shape(cardata_nonelectric)
```

```
## [1] "Our dataset shape is now: (21738, 37)"
```

Electric Vehicle Data

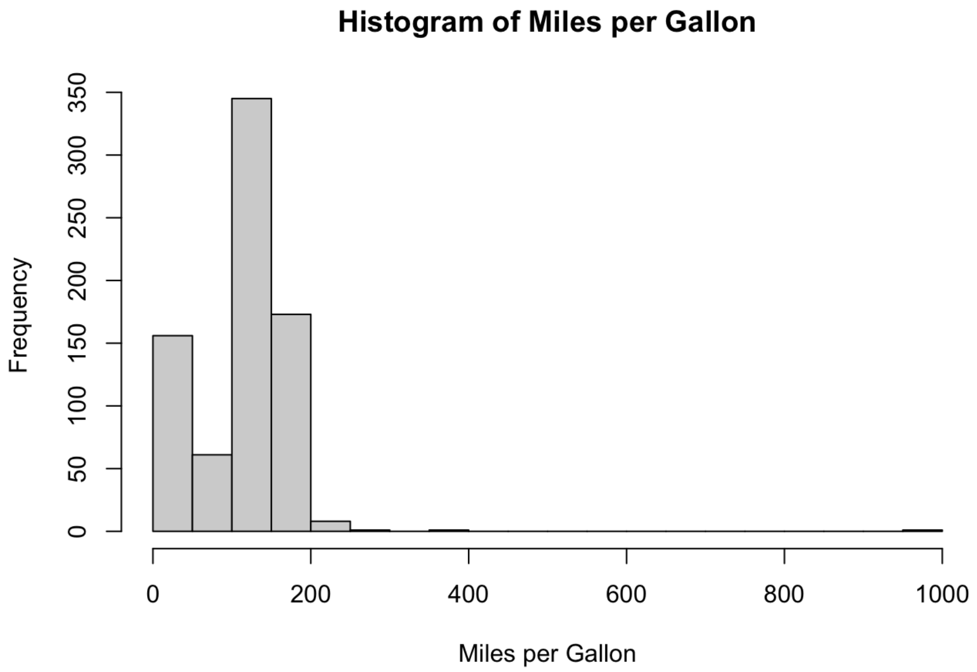
We still have columns expressing `NA` values, however. These columns, of course, have less than 10% of their values being `NA` , after the removal performed above. These columns can be seen below.

```
# print the proportions of NA values in each column using a threshold of 0%
electric_table <- calculate_proportions(cardata_electric, 0)
print_table(electric_table)
```

Column Name	Proportion of Values NA
CO2 (g/mi)	1.00000
RND_ADJ_FE	0.15034

We can see that the electric vehicle data only has two columns with missing values: `RND_ADJ_FE` and `CO2 (g/mi)`. We've decided to keep the `CO2 (g/mi)` in order to provide comparisons between the electric and non-electric vehicle data, but we want to address the specific missing values of the `RND_ADJ_FE` variable. This variable is an expression of miles per gallon. Let's take a look at a histogram of this variable.

```
# plot histogram of the electric car MPG values
hist(cardata_electric$RND_ADJ_FE, xlab = 'Miles per Gallon', main = 'Histogram of Miles per Gallon', breaks = 20)
```



We can see that there is at least one outlier on the high end of the distribution, expressing a value of 1000. We will replace these value(s) with missing values in order to obtain results that have more validity. We will then replace the missing values with the *median* value of the distribution, as this is a safer way to do so without making any bold assumptions.

```
# replace high outliers with missing value
cardata_electric$RND_ADJ_FE[cardata_electric$RND_ADJ_FE > 500] <- NA
# replace missing values with the median value
cardata_electric$RND_ADJ_FE[is.na(cardata_electric$RND_ADJ_FE)] <- median(cardata_electric$RND_ADJ_FE, na.rm = TR
UE)
```

We can now observe the columns with missing values in the electric vehicle dataset and notice that we have cleaned it to our liking. Note that the `CO2 (g/mi)` variable still remains and will likely be interpreted as **0** emission from these electric vehicles.

```
# print the proportions of NA values in each column using a threshold of 0%
electric_table <- calculate_proportions(cardata_electric, 0)
print_table(electric_table)
```

Column Name	Proportion of Values NA
CO2 (g/mi)	1

Non-Electric Vehicle Data

We still have columns expressing `NA` values for the non-electric vehicle dataset. These columns, of course, have less than 10% of their values being `NA` , after the removal performed above. These columns can be seen below.

```
# print the proportions of NA values in each column using a threshold of 0%
nonelectric_table <- calculate_proportions(cardata_nonelectric, 0)
print_table(nonelectric_table)
```

Column Name	Proportion of Values NA
THC (g/mi)	0.09803
CO (g/mi)	0.09789
DT-Inertia Work Ratio Rating	0.04835
DT-Absolute Speed Change Ratg	0.04835
DT-Energy Economy Rating	0.04835
CO2 (g/mi)	0.01702
Aftertreatment Device Cd	0.00888
Aftertreatment Device Desc	0.00888
RND_ADJ_FE	0.00823
# of Cylinders and Rotors	0.00745

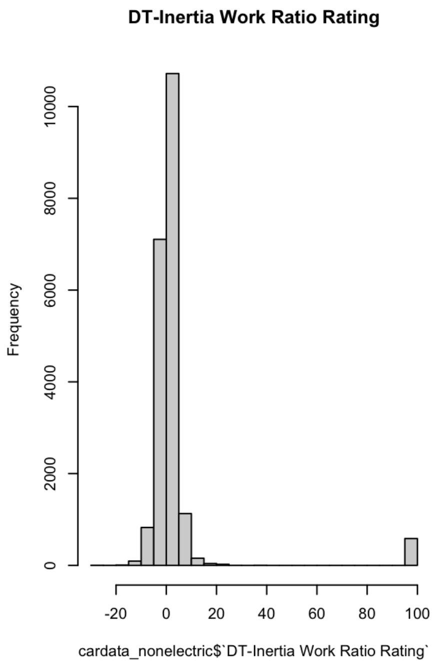
`DT-Inertia Work Ratio Rating`, `DT-Absolute Speed Change Ratg`, and `DT-Energy Economy Rating`

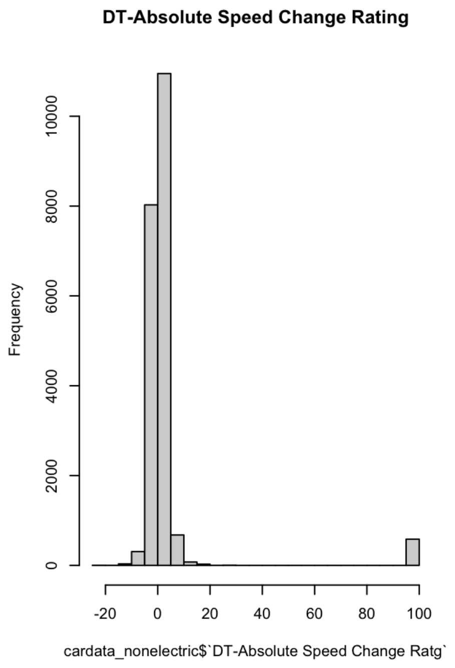
Let's start by observing these three variables:

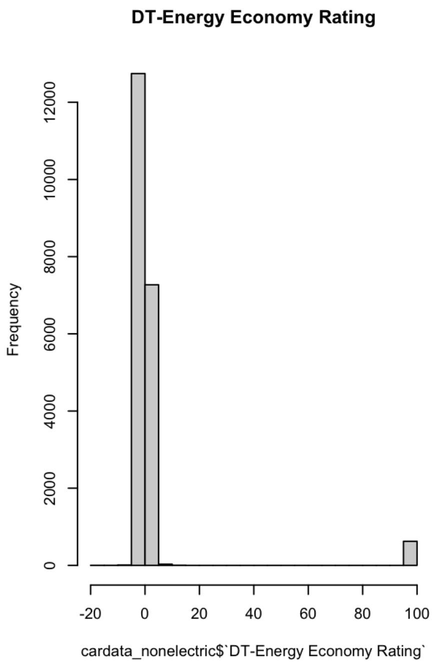
- `DT-Inertia Work Ratio Rating`
- `DT-Absolute Speed Change Ratg`
- `DT-Energy Economy Rating`

We can make histograms to see what their distributions are like.

```
par(mfrow = c(1, 3))
hist(cardata_nonelectric$`DT-Inertia Work Ratio Rating`, main = 'DT-Inertia Work Ratio Rating', breaks = 20)
hist(cardata_nonelectric$`DT-Absolute Speed Change Ratg`, main = 'DT-Absolute Speed Change Rating', breaks = 20)
hist(cardata_nonelectric$`DT-Energy Economy Rating`, main = 'DT-Energy Economy Rating', breaks = 20)
```







It is evident that each of these columns has a set of observations marked with a rating around 100. This is likely comparable to an `NA` value, as the distribution of values for these attributes is almost entirely hovering around a value of 0. Let's replace these values with `NA` values so that we don't confuse ourselves with them being valid. Then, since the remaining distribution appears to be very symmetric, we can replace the missing values with the mean of the column for each variable.

```
# replace high outliers with NA
cardata_nonelectric$`DT-Inertia Work Ratio Rating`[cardata_nonelectric$`DT-Inertia Work Ratio Rating` > 50] <- NA
cardata_nonelectric$`DT-Absolute Speed Change Ratg`[cardata_nonelectric$`DT-Absolute Speed Change Ratg` > 50] <-
NA
cardata_nonelectric$`DT-Energy Economy Rating`[cardata_nonelectric$`DT-Energy Economy Rating` > 50] <- NA

# replace missing values with the mean value
cardata_nonelectric$`DT-Inertia Work Ratio Rating`[is.na(cardata_nonelectric$`DT-Inertia Work Ratio Rating`)] <-
 mean(cardata_nonelectric$`DT-Inertia Work Ratio Rating`, na.rm = TRUE)
cardata_nonelectric$`DT-Absolute Speed Change Ratg`[is.na(cardata_nonelectric$`DT-Absolute Speed Change Ratg`)] <
- mean(cardata_nonelectric$`DT-Absolute Speed Change Ratg`, na.rm = TRUE)
cardata_nonelectric$`DT-Energy Economy Rating`[is.na(cardata_nonelectric$`DT-Energy Economy Rating`)] <- mean(car
data_nonelectric$`DT-Energy Economy Rating`, na.rm = TRUE)
```

We can now observe our new, updated measurements of proportions of values in columns that are `NA`.

```
# create data frame for printing proportions of NA values
nonelectric_table <- calculate_proportions(cardata_nonelectric, 0)
print_table(nonelectric_table)
```

Column Name	Proportion of Values NA
THC (g/mi)	0.09803
CO (g/mi)	0.09789
CO2 (g/mi)	0.01702
Aftertreatment Device Cd	0.00888
Aftertreatment Device Desc	0.00888
RND_ADJ_FE	0.00823
# of Cylinders and Rotors	0.00745

`Aftertreatment Device Cd` and `Aftertreatment Device Desc`

The following two variables have the same number of missing values in the dataset. Let’s see if they are from the same rows.

- `Aftertreatment Device Cd`
- `Aftertreatment Device Desc`

```
# extract row names where these three variables are missing
after_cd_null <- rownames(cardata_nonelectric[is.na(cardata_nonelectric$`Aftertreatment Device Cd`), ])
after_desc_null <- rownames(cardata_nonelectric[is.na(cardata_nonelectric$`Aftertreatment Device Desc`), ])

# ensure that the column names are all the same across all datasets
if ( mean(after_cd_null == after_desc_null) == 1 ) {
  print('These variables are missing in the same rows in the dataset.')
} else {
  print('These variables are missing in different rows in the dataset.')
}
```

```
## [1] "These variables are missing in the same rows in the dataset."
```

The missing values in the dataset for these two variables *do* come from the same rows, which makes sense given their association with each other; the `Aftertreatment Device Desc` variable is just a description for the label of the `Aftertreatment Device Cd` variable. Since these are categorical variables, and there are only a small number of `NA` values (less than 1%), we will leave the `NA` values as they are and likely remove them if we engage in an analysis on both columns. We opt not to remove the *rows* corresponding to these missing values because there is valuable data in the other 30 columns present.

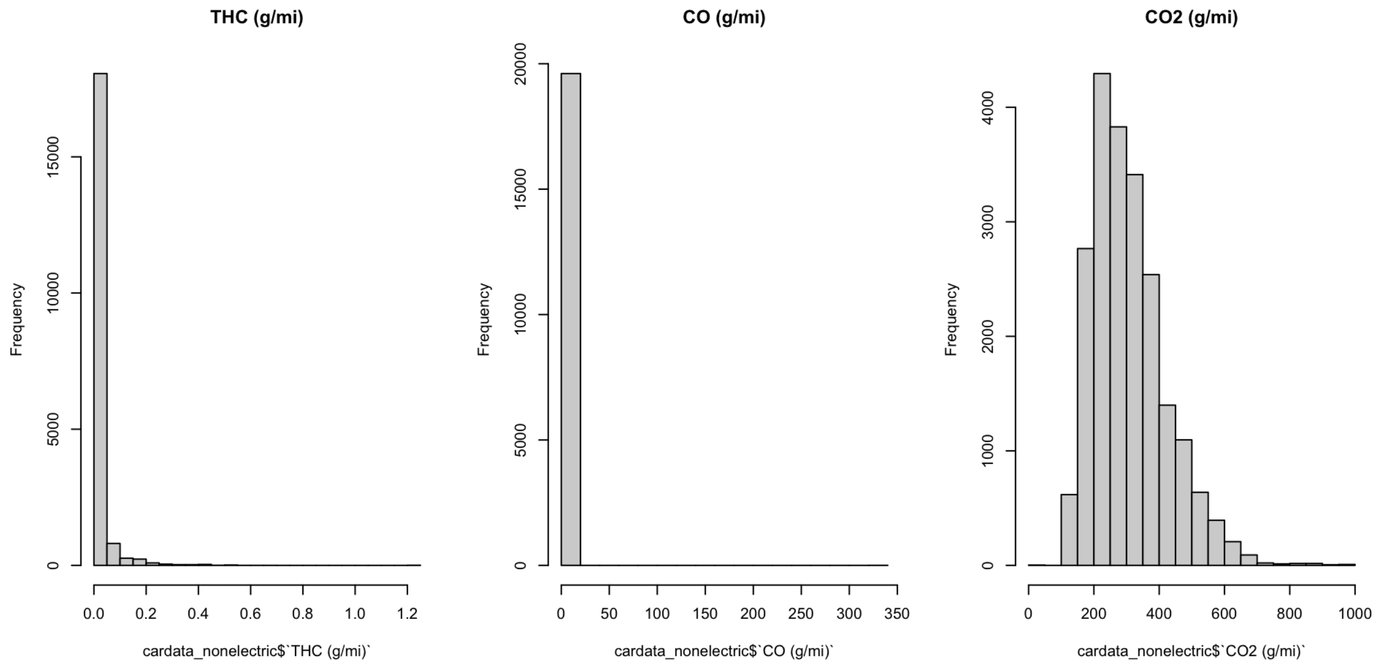
The last few variables containing missing values left to address are:

- `THC (g/mi)`
- `CO (g/mi)`
- `CO2 (g/mi)`
- `# of Cylinders and Rotors`
- `RND_ADJ_FE`

`THC (g/mi)`, `CO (g/mi)`, and `CO2 (g/mi)`

Let’s start with these variables and observe their missing values.

```
par(mfrow = c(1, 3))
hist(cardata_nonelectric$`THC (g/mi)`, main = 'THC (g/mi)', breaks = 20)
hist(cardata_nonelectric$`CO (g/mi)`, main = 'CO (g/mi)', breaks = 20)
hist(cardata_nonelectric$`CO2 (g/mi)`, main = 'CO2 (g/mi)', breaks = 20)
```



Given the shapes of these distributions, we believe it is a safe choice to replace the missing values (which make up less than 10% of the data for each distribution) with the median value in that distribution. This is a conservative approach and allows us to refrain from making any bold assumptions about the true values that are missing.

```
# replace missing values with the mean value
cardata_nonelectric$`THC (g/mi)`[is.na(cardata_nonelectric$`THC (g/mi)`)] <- mean(cardata_nonelectric$`THC (g/mi)
`, na.rm = TRUE)
cardata_nonelectric$`CO (g/mi)`[is.na(cardata_nonelectric$`CO (g/mi)`)] <- mean(cardata_nonelectric$`CO (g/mi)`,
 na.rm = TRUE)
cardata_nonelectric$`CO2 (g/mi)`[is.na(cardata_nonelectric$`CO2 (g/mi)`)] <- mean(cardata_nonelectric$`CO2 (g/mi)
`, na.rm = TRUE)
```

of Cylinders and Rotors

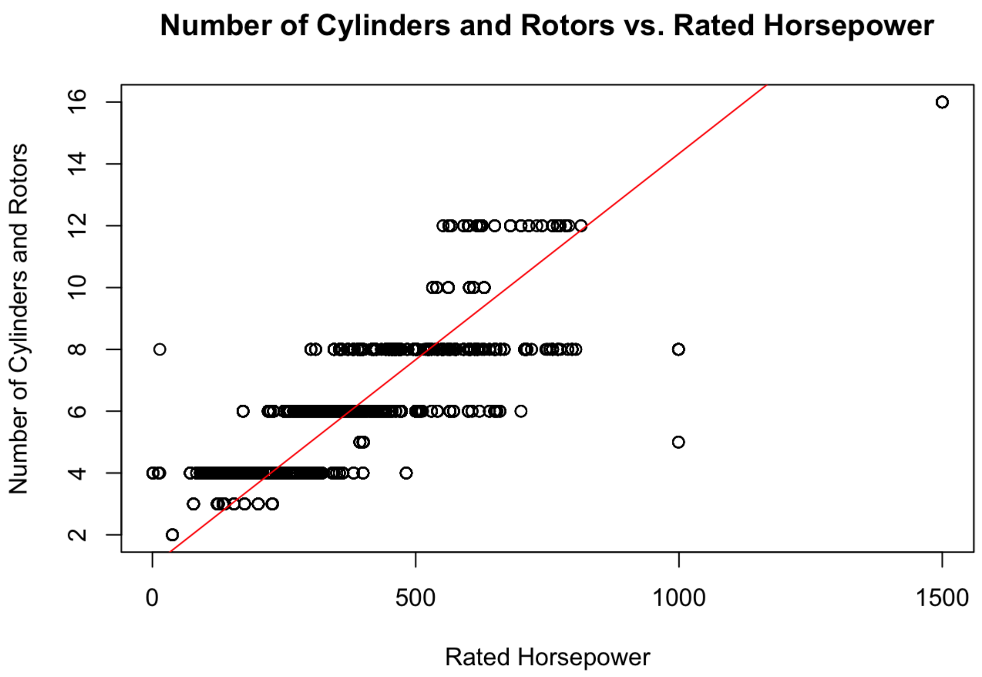
Let's move on to `# of Cylinders and Rotors`.

```
# table of number of cylinders
table(cardata_nonelectric$`# of Cylinders and Rotors`)
```

```
## 
##     2     3     4     5     6     8    10    12    16 
##    21   390 10709    45  6531  3333   166   350    31
```

We can see above that the number of cylinders and rotors is almost exclusively an even number, with values of 3 and 5 appearing some number of times.

```
# plot number of cylinders versus horsepower
plot(cardata_nonelectric$`Rated Horsepower`, cardata_nonelectric$`# of Cylinders and Rotors`, xlab = 'Rated Horse
power', ylab = 'Number of Cylinders and Rotors', main = 'Number of Cylinders and Rotors vs. Rated Horsepower')
abline(a = 1, b = 1/75, col = 'red')
```



Above, we can see an approximation of the number of cylinders (`# of Cylinders and Rotors`) and rotors predicted by the horsepower (`Rated Horsepower`). Using this approximation, we will fill in the missing values of `# of Cylinders and Rotors` using the nearest even value. Note that the even values make up the vast majority in this dataset, as values of 1, 3, and 5, for instance, are very rare.

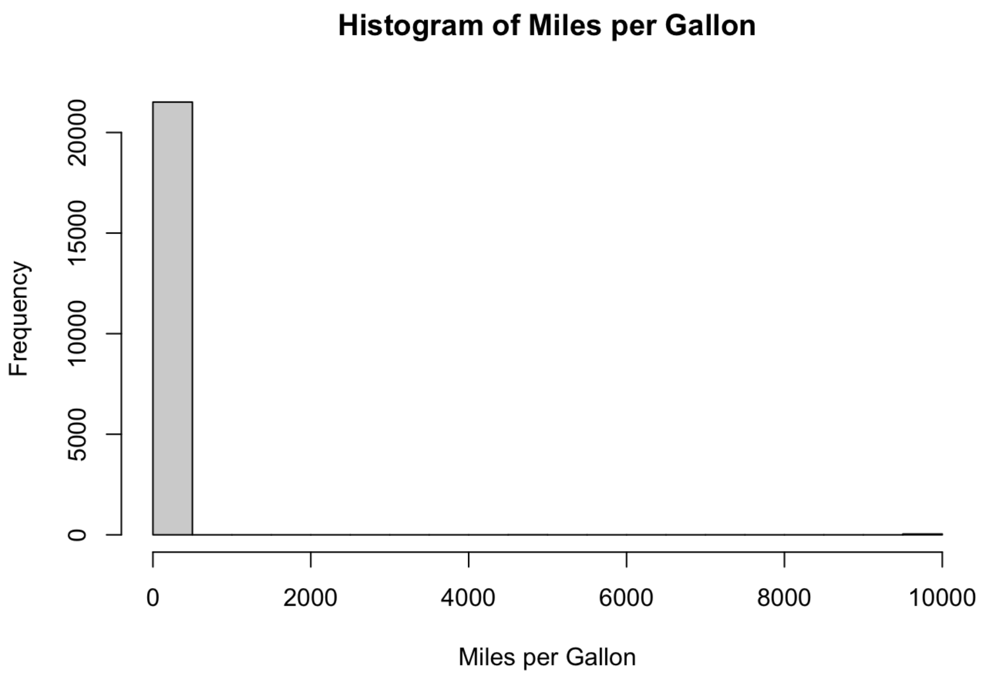
```
# replace missing cylinder values with the approximation above
for (i in 1:nrow(cardata_nonelectric)) {
  if (is.na(cardata_nonelectric$`# of Cylinders and Rotors`[i])) {
    val <- 1 + cardata_nonelectric$`Rated Horsepower`[i] / 75
    cardata_nonelectric$`# of Cylinders and Rotors`[i] <- round(val / 2) * 2
  }
}
```

`RND_ADJ_FE`

Finally, let's move on to `RND_ADJ_FE` .

As we did above with the electric vehicles, let's observe a histogram of the `RND_ADJ_FE` variable for the non-electric vehicles. Remember, this variable expresses the number of miles per gallon for the vehicle.

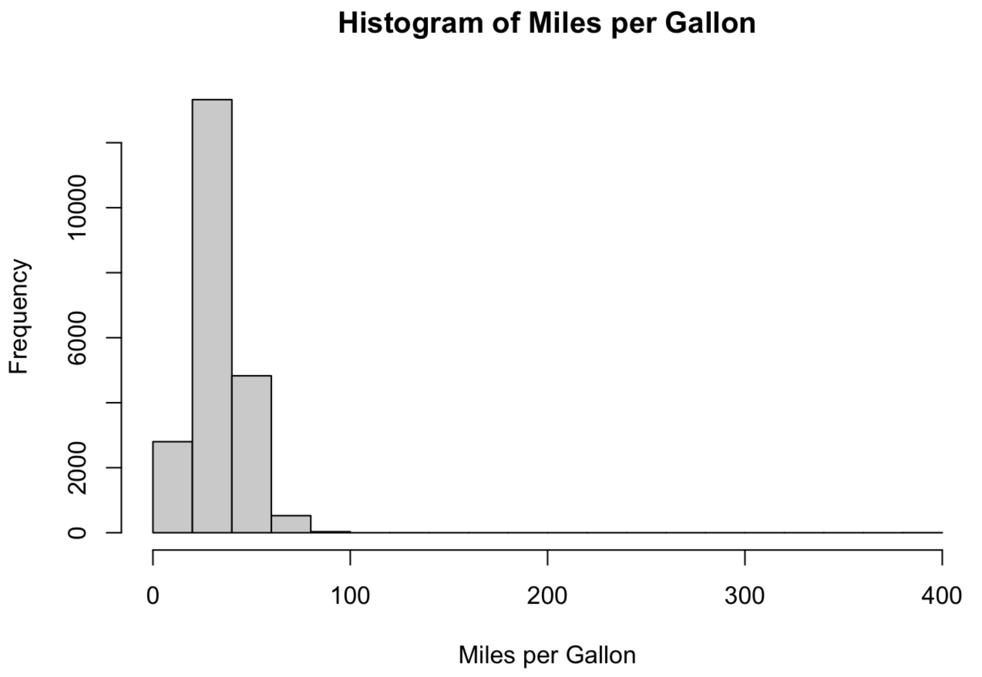
```
# plot histogram of the electric car MPG values
hist(cardata_nonelectric$RND_ADJ_FE, xlab = 'Miles per Gallon', main = 'Histogram of Miles per Gallon', breaks =
20)
```



We can see that there is at least one outlier on the high end of the distribution, expressing a value of 1000. We will replace these value(s) with missing values in order to obtain results that have more validity. We can then try to view the true distribution of values by constructing another histogram.

```
# replace high outliers with missing value
cardata_nonelectric$RND_ADJ_FE[cardata_nonelectric$RND_ADJ_FE > 500] <- NA

# plot histogram of the electric car MPG values
hist(cardata_nonelectric$RND_ADJ_FE, xlab = 'Miles per Gallon', main = 'Histogram of Miles per Gallon', breaks =
20)
```



We will perform the same task as before with the electric vehicles by replacing the missing values in this column with the *median* value of the distribution, as this is a safer way to do so without making any bold assumptions.

```
# replace missing values with the median value
cardata_nonelectric$RND_ADJ_FE[is.na(cardata_nonelectric$RND_ADJ_FE)] <- median(cardata_nonelectric$RND_ADJ_FE, n
a.rm = TRUE)
```

We can now observe the columns with missing values in the non-electric vehicle dataset and notice that we have cleaned it to our liking. Remember that we opted to keep the missing values for the `Aftertreatment Device Cd` and `Aftertreatment Device Desc` columns, as mentioned above.

```
# print the proportions of NA values in each column using a threshold of 0%
nonelectric_table <- calculate_proportions(cardata_nonelectric, 0)
print_table(nonelectric_table)
```

Column Name	Proportion of Values NA
Aftertreatment Device Cd	0.00888
Aftertreatment Device Desc	0.00888

Final Look

Let's take a look at the proportions of missing values now, as we have addressed them all (and kept some).

```
# print the proportions of NA values in each column
electric_table <- calculate_proportions(cardata_electric, 0)
nonelectric_table <- calculate_proportions(cardata_nonelectric, 0)
print_table(electric_table)
```

Column Name	Proportion of Values NA
CO2 (g/mi)	1

```
print_table(nonelectric_table)
```

Column Name	Proportion of Values NA
Aftertreatment Device Cd	0.00888
Aftertreatment Device Desc	0.00888

After addressing the missing values in the data, we find that we only have approximately **0.15%** of our data missing. This is a great improvement from the **15.64%** that we had initially and something that we can take with us as we look to analyze the data further.

Export Clean Data

Finally, we need to export this cleaned data so that we can use it in our analyses.

```
write.csv(cardata_electric, '../data/cardata_electric_clean.csv')
write.csv(cardata_nonelectric, '../data/cardata_nonelectric_clean.csv')
```