Description of Data Cleaning

Megan Fantes

October 24, 2016

Introduction

Our lives today depend on automobiles. We like our cars to be the "best," and one mark of automotive superiority to fuel economy. The better a car's fuel economy, the more money you save, and who doesn't like to save money? Today we are looking at fuel economy data collected at the Environmental Protection Agency's National Vehicle and Fuel Emissions Laboratory in Ann Arbor, Michigan, and by vehicle manufacturers with oversight by EPA.

Our dataset measures 83 variables on 38,017 cars. Some variables are properties of the cars – like year, make, and model – and most are measures of the car's fuel efficiency. As a companion to this document, you will find a spreadsheet containing metadata about our dataset containing the variables names, their descriptions, and their units.

Normalization

We will split our dataset into 2 parts, or "normalize" it. "Normalization" is the process of splitting a dataset into its individual pieces so that information is not unecessarily repeated. The two pieces of our dataset are Vehicle Properties and Fuel Type Properties. Many variable describe the individual cars, but some of the cars can run on 2 different types of fuel, and some variables are measured for *both* types of fuel. So we will split out dataset into 2 discrete parts: the unique properties of each observed car, and the properties of fuel measured on each type of fuel for each car.

It is important to note that our Fuel Type Properties table still contains the vehicle ID. We keep the vehicle ID in the Fuel Types table to make it easier to combine the table later for data cleaning. Also, the values for the fuel types vary across cars, thus to get a complete picture of the distribution and effect of fuel type we need to keep the individual vehicle and the their fuel types connected.

Data Cleaning: Fuel Type Properties

We start out data cleaning by creating the Fuel Type Properties Table.

There are many variables that are measure as "xxx for fuel type 1" and "xxx for fuel Type 2". We want to gather all of these variables into a smaller amount of columns, so that there is one column indicating fuel type – either one or 2 – and multiple columns indicating the value of each variable for each fuel type for each car.

We will use the plyr package to gather the variables (plyr actually uses the word "melt") because plyr allows flexibility in specifying the value that we gather on (i.e. which column does not change when we gather other variables).

1) find the annual petroleum consumption for each fuel type for each vehicle in the dataset.

(lines _____ in R code)

We create a temporary table title "to melt" (referencing the plyr function "melt", which gathers spread-out variables into one column), and give it 3 columns: id, the petroleum consumption for fuel type 1 for each car, and the petroleum consumption for fuel type 2 for each car. Then we "melt" the data table and split it into cunsumption per fuel type for each id. Then we give the columns more descriptive names, and change the

gathered factors to "1" and "2", to reflect fuel types 1 and 2. Finally, we add these columns to a new, final table called Fuel Type Properties.

We will join every subsequent variable onto this Fuel Type Properties table, and we will join on the unique identifier combination of of vehicle id and fuel type.

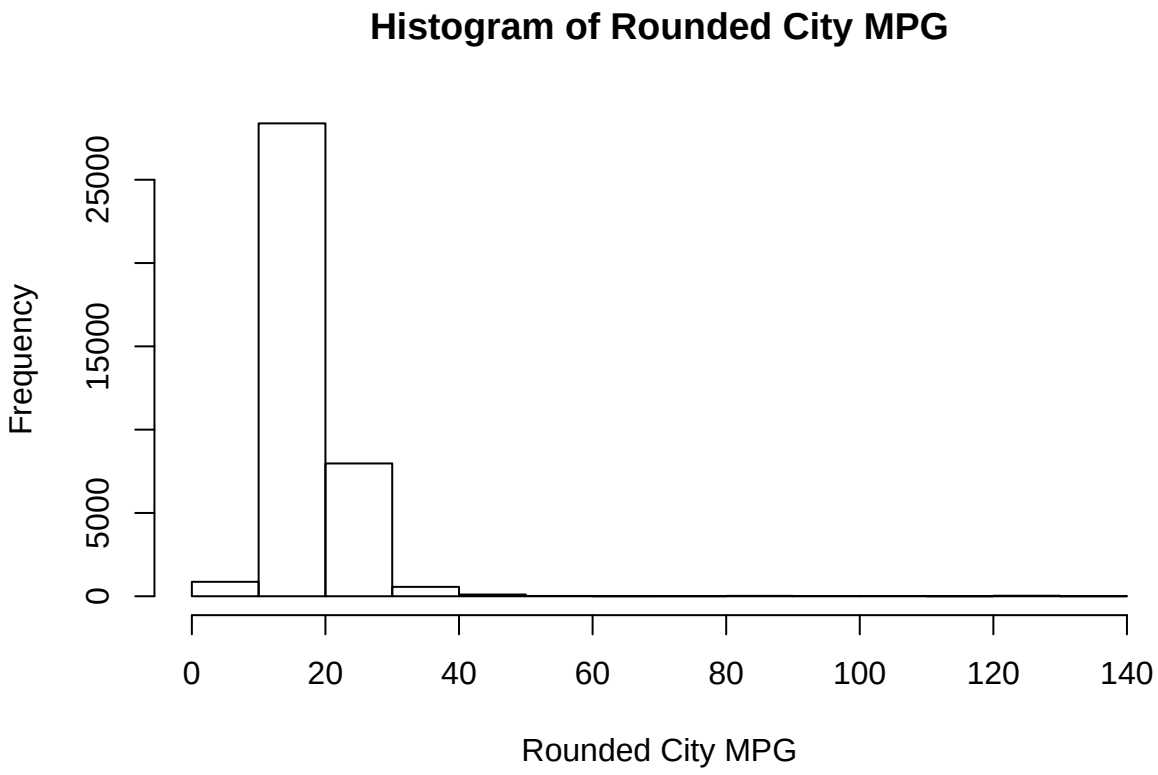
We then arrange the table in ascending order first by id and then by fuel type.

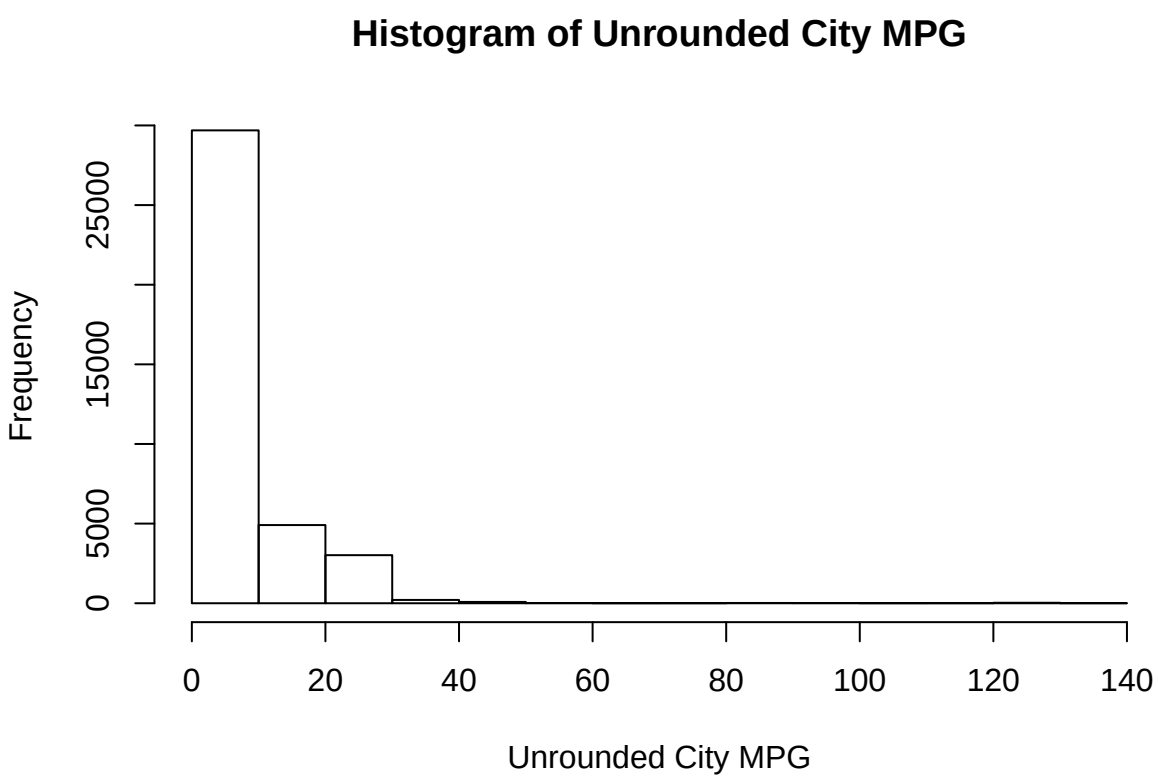
2) find the city MPG for each fuel type for each vehicle.

(lines _____ in R code)

There were 2 types of City MPG collected in this dataset: rounded MPG and unrounded MPG. Since they express the same information, just in slightly different scales, we will keep only one in our dataset. In order to choose which one we should keep, we can look at histograms of the rounded and unrounded MPG data.

We will only look at histograms for fuel type 1, because all cars have a fuel type 1 while only some have fueld type 2, meaning variables concerning fuel type 1 are more indicative of data trends. (And we will continue looking only at fuel type 1 for future variable choice decisions.)





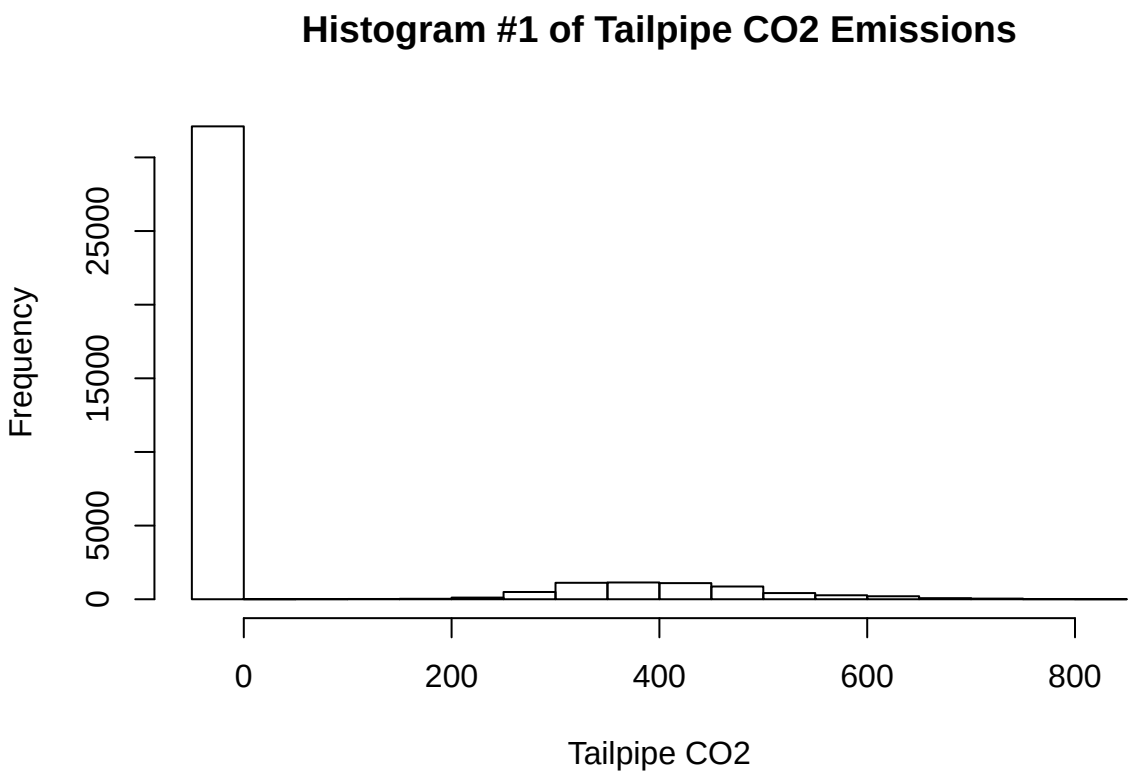
Unavailable or missing data is coded as 0 in this data set, and we can see that the majority of the data are coded 0 in the Unrounded data, indicating that the majority of the data is missing from the Unrounded data. Therefore we will use the Rounded City MPG.

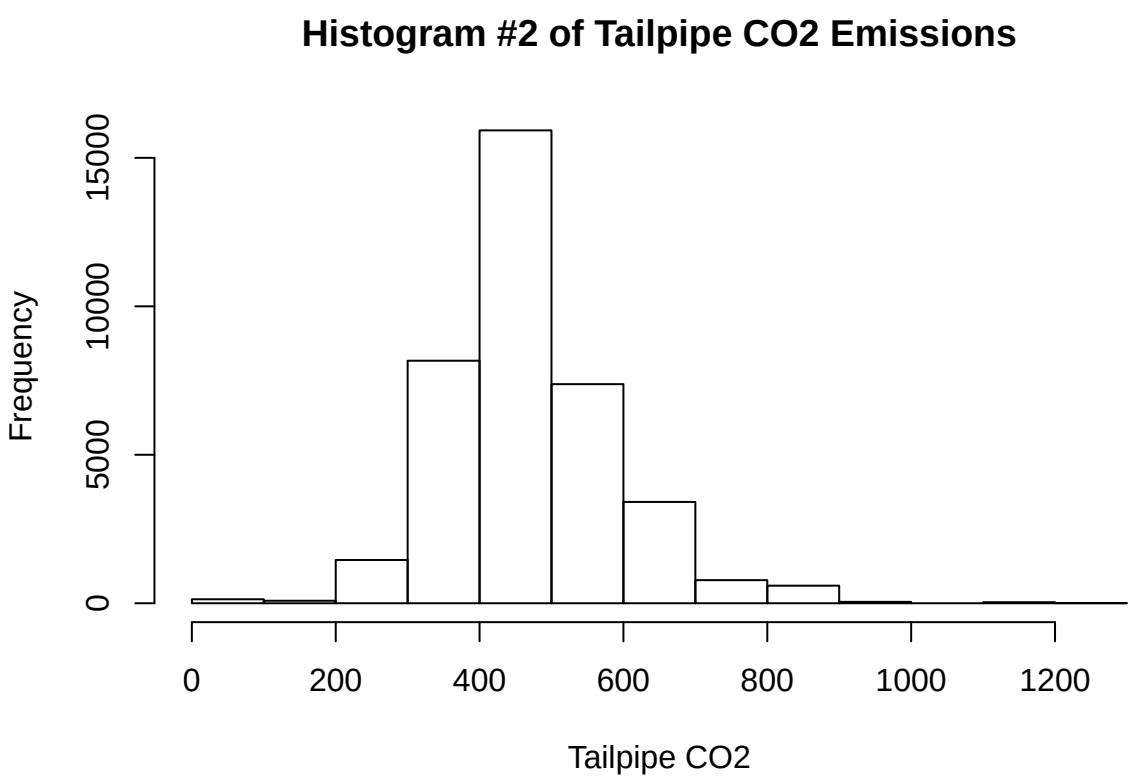
We then follow the exact same process for melting the data as with the Annual Petroleum Consumption variable.

3) Find Tailpipe CO2 emissions for each fuel type for each vehicle.

(lines _____ in R code)

There were 2 sets of variables collected to measure Tailpipe CO2 emissions from the 2 types of fuel (co2/co2A and co2TailpipeGpm/co2TailpipeAGpm), and their desccriptions appeared to be the same. Since they measure the same quantity, we can again look at histograms to determine which variable holds more meaningful data.


Histogram #1 of Tailpipe CO2 Emissions
Frequency
0
5000
15000
25000
0
200
400
600
800
Tailpipe CO2



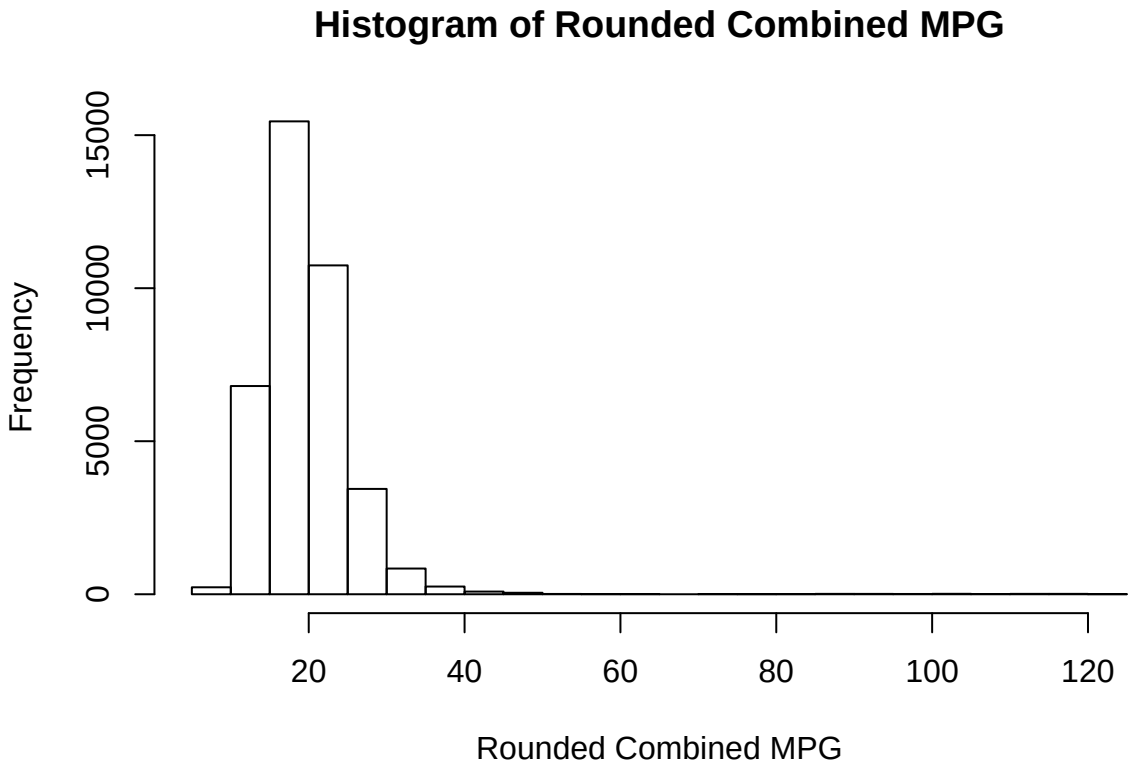
Again, missing data is coded as 0 for these variables, and we can see that the first set of variables, in Histogram #1, have more missing data, therefore we will use the variables in Histogram #2.

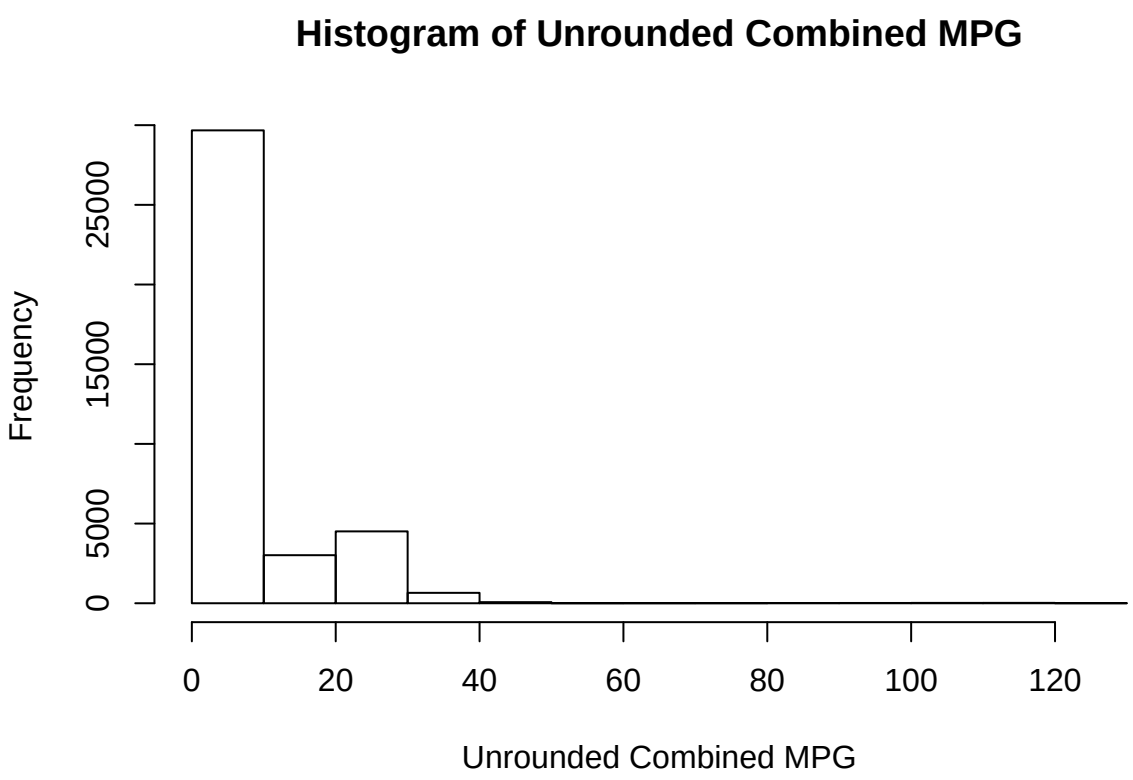
We then follow the exact same process for melting the data as with the Annual Petroleum Consumption variable.

4) Find Combined MPG for each fuel type for each vehicle.

(lines _____ in R code)

Like with City MPG, there were 2 types of Combined MPG measured in this dataset: rounded and unrounded. Like with City MPg, we can look at histograms to determine which set of variables we should use.


Histogram of Rounded Combined MPG
Frequency
0
5000
10000
15000
20
40
60
80
100
120
Rounded Combined MPG



Again, missing data is coded as 0 for these variable, and we can see that the Unrounded MPG has more missing data, therefore we will use the Rounded Combined MPG.

We then follow the exact same process for melting the data as with the Annual Petroleum Consumption variable.

5) Find Fuel Cost for each fuel type for each vehicle.

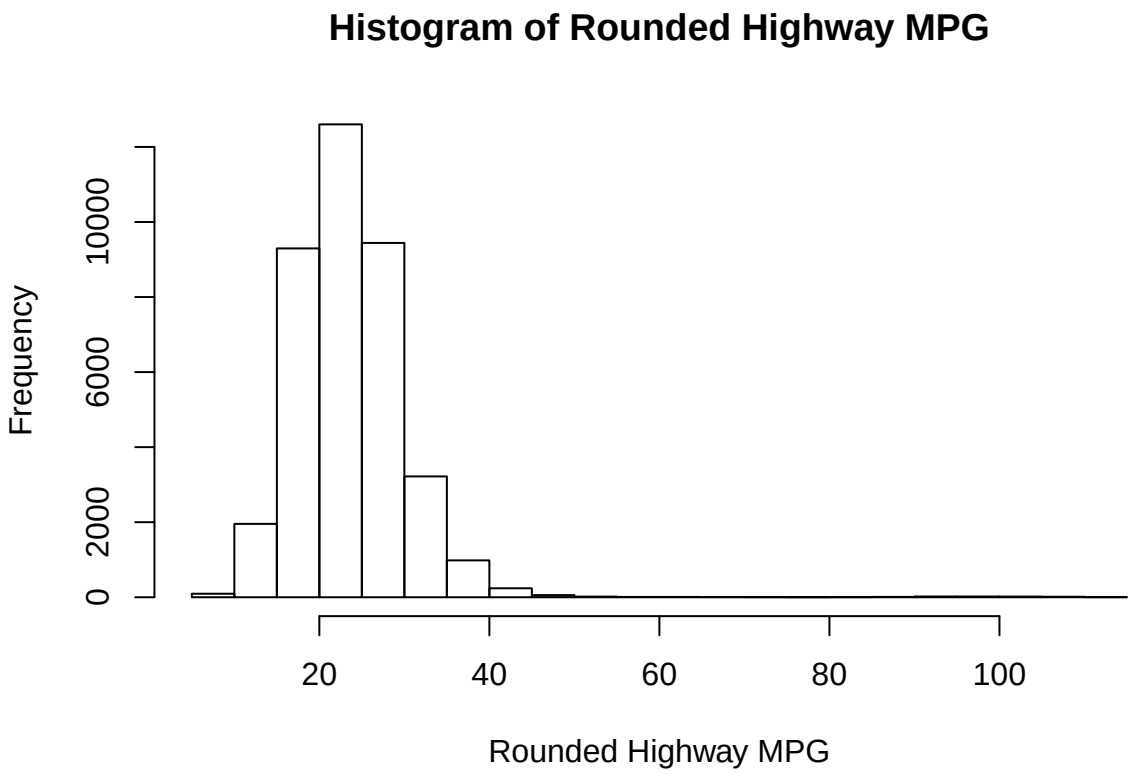
(lines _____ in R code)

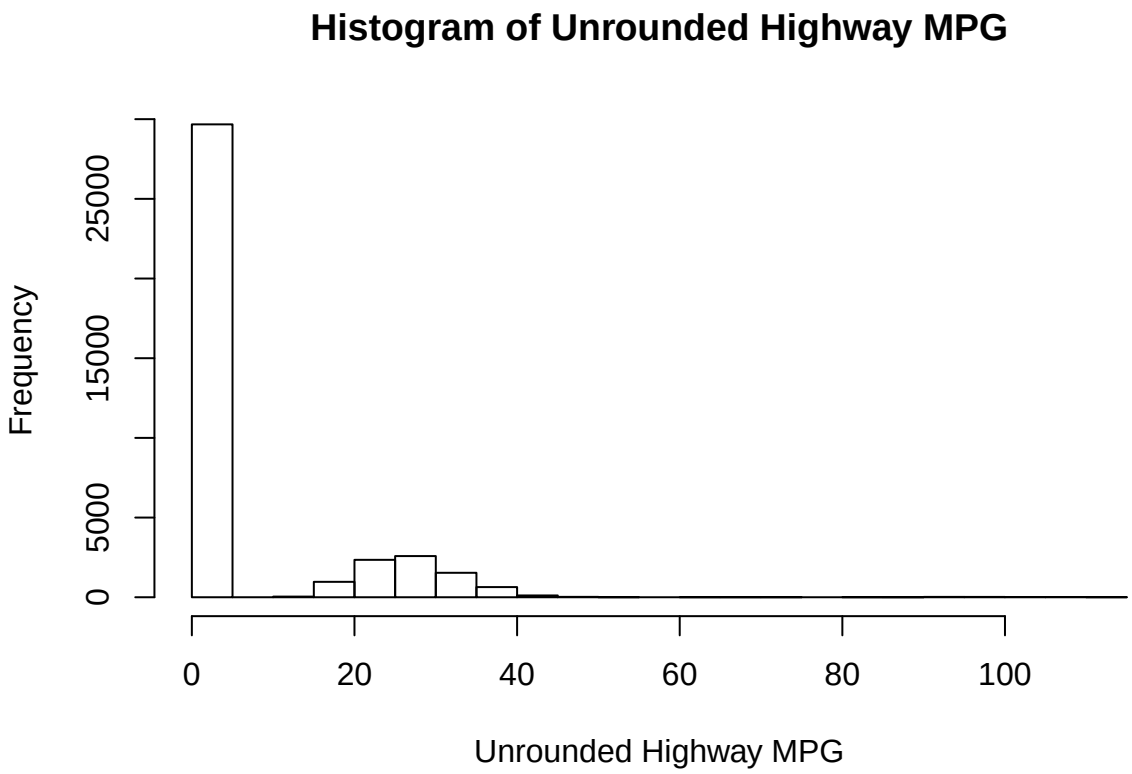
We follow the exact same process for melting the data as with the Annual Petroleum Consumption variable.

6) Find Highway MPG for each fuel type for each vehicle.

(lines _____in R code)

Like with City MPG, there were 2 types of Highway MPG measured in this dataset: rounded and unrounded. Like with City MPg, we can look at histograms to determine which set of variables we should use.


Histogram of Rounded Highway MPG
Frequency
0
2000
6000
10000
20
40
60
80
100
Rounded Highway MPG



Again, missing data is coded as 0 for these variable, and we can see that the Unrounded MPG has more missing data, therefore we will use the Rounded Highway MPG.

We then follow the exact same process for melting the data as with the Annual Petroleum Consumption variable.

7) Find Fuel Cost for each fuel type for each vehicle.

(lines _____ in R code)

We follow the exact same process for melting the data as with the Annual Petroleum Consumption variable.

8) Rearrange Fuel Type Properties into a more logical order

9) Convert columns with factor levels from string to factors

Then convert all missing data to "NA" type factors.

Data Cleaning: Vehicle Properties

To complete our normalization of the data, we make a second table with the rest of the variables of the data, all of which describe the vehicles observed during data collection. A complete table of variable names, descriptions, and units is available on my gitHub.

We do not include some of the original variables in the Vehicle Properties data, such at createdOn and modifiedOn. We did not include such variables because they do not give information about the vehicles, thus we do not need them for our analysis.