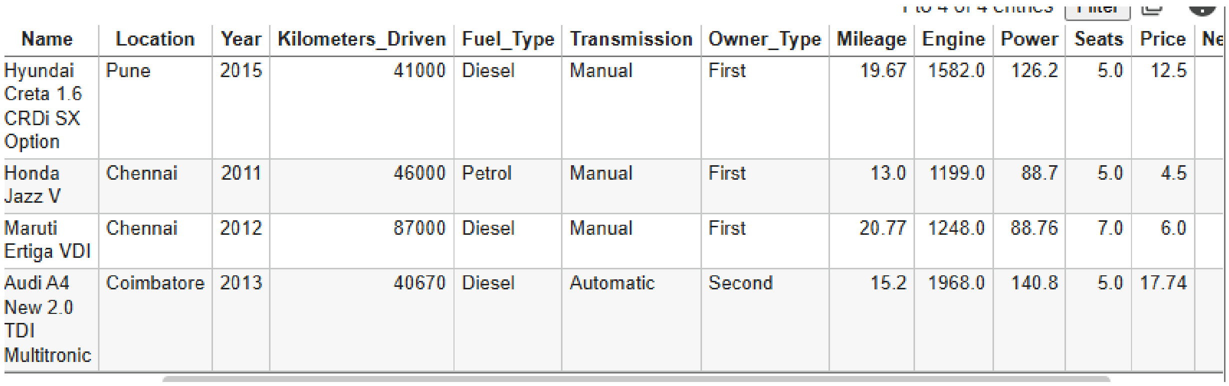
Exercise 1:

a) Look for the missing values in all the columns and either impute them (replace with mean, median, or mode) or drop them. Justify your action for this task:

1. Drop the New_price column because it contains 86% of the missing value.
2. Replace The missing value in column Engine, Power and Mileage by the median because if the distribution is symmetric so median=mean if skewed the median is the most accurate.
3. Replace the missing value in column seat by mode because it is a discrete variable that means we replace the missing value with the most repeated value.

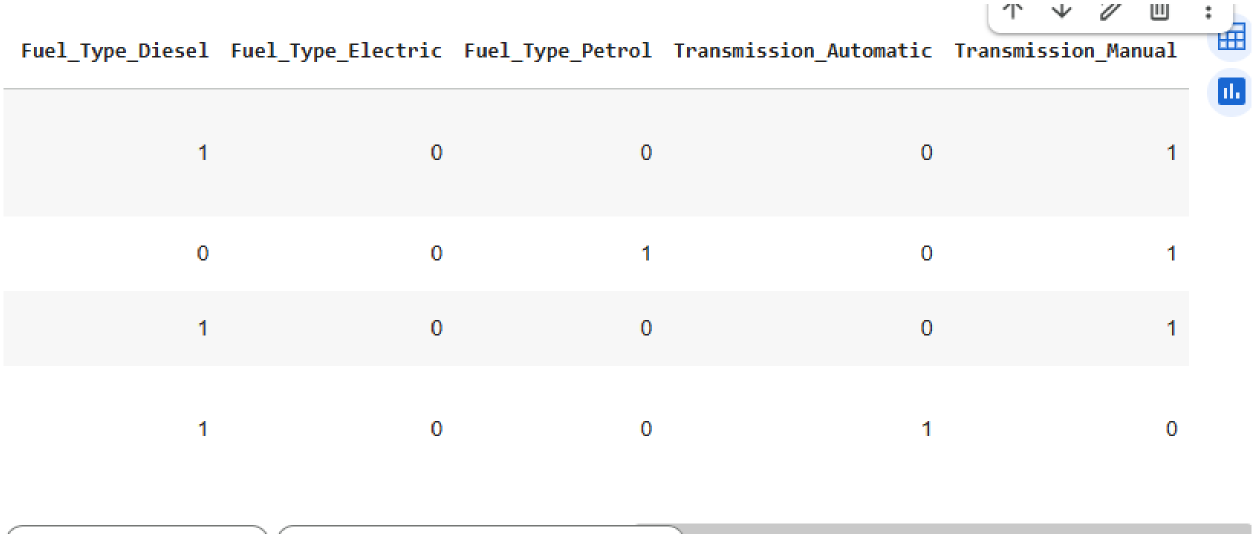
b) Remove the units from some of the attributes and only keep the numerical values (for example remove kmpl from "Mileage", CC from "Engine", bhp from "Power", and lakh from "New_price").



1 to 4 of 4 entries Filter

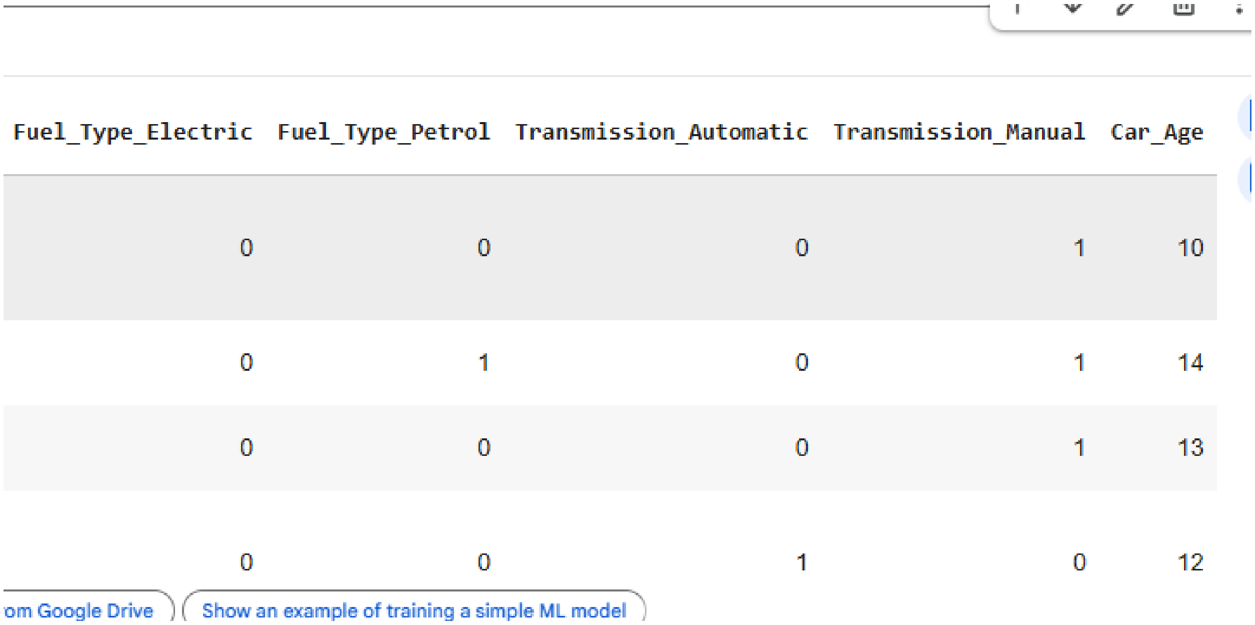
Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price	Ne
Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67	1582.0	126.2	5.0	12.5	
Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	13.0	1199.0	88.7	5.0	4.5	
Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77	1248.0	88.76	7.0	6.0	
Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2	1968.0	140.8	5.0	17.74	

c) Change the categorical variables ("Fuel_Type" and "Transmission") into a numerical one hot encoded value.



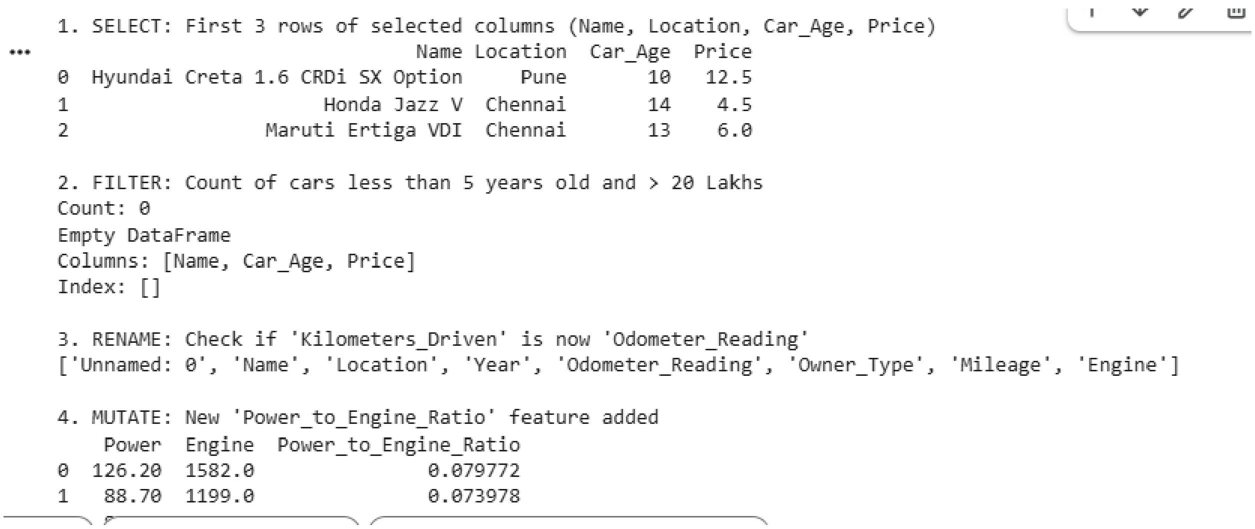
Fuel_Type_Diesel	Fuel_Type_Electric	Fuel_Type_Petrol	Transmission_Automatic	Transmission_Manual
1	0	0	0	1
0	0	1	0	1
1	0	0	0	1
1	0	0	1	0

d) Create one more feature and add this column to the dataset (you can use mutate function in R for this). For example, you can calculate the current age of the car by subtracting the "Year" value from the current year.

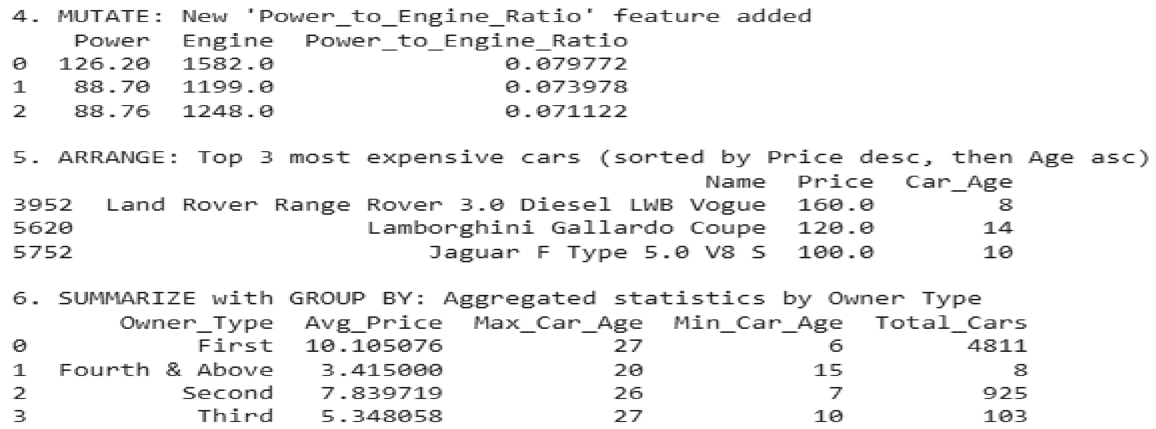


Fuel_Type_Electric	Fuel_Type_Petrol	Transmission_Automatic	Transmission_Manual	Car_Age
0	0	0	1	10
0	1	0	1	14
0	0	0	1	13
0	0	1	0	12

e) Perform select, filter, rename, mutate, arrange and summarize with group by operations (or their equivalent operations in python) on this dataset.



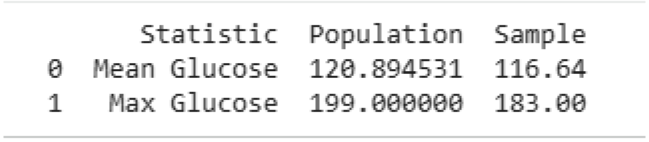
```
1. SELECT: First 3 rows of selected columns (Name, Location, Car_Age, Price)
                               Name Location  Car_Age  Price
0  Hyundai Creta 1.6 CRDi SX Option     Pune       10   12.5
1                      Honda Jazz V  Chennai       14    4.5
2                 Maruti Ertiga VDI  Chennai       13    6.0

2. FILTER: Count of cars less than 5 years old and > 20 Lakhs
Count: 0
Empty DataFrame
Columns: [Name, Car_Age, Price]
Index: []

3. RENAME: Check if 'Kilometers_Driven' is now 'Odometer_Reading'
['Unnamed: 0', 'Name', 'Location', 'Year', 'Odometer_Reading', 'Owner_Type', 'Mileage', 'Engine']

4. MUTATE: New 'Power_to_Engine_Ratio' feature added
    Power  Engine  Power_to_Engine_Ratio
0  126.20  1582.0               0.079772
1   88.70  1199.0               0.073978
```



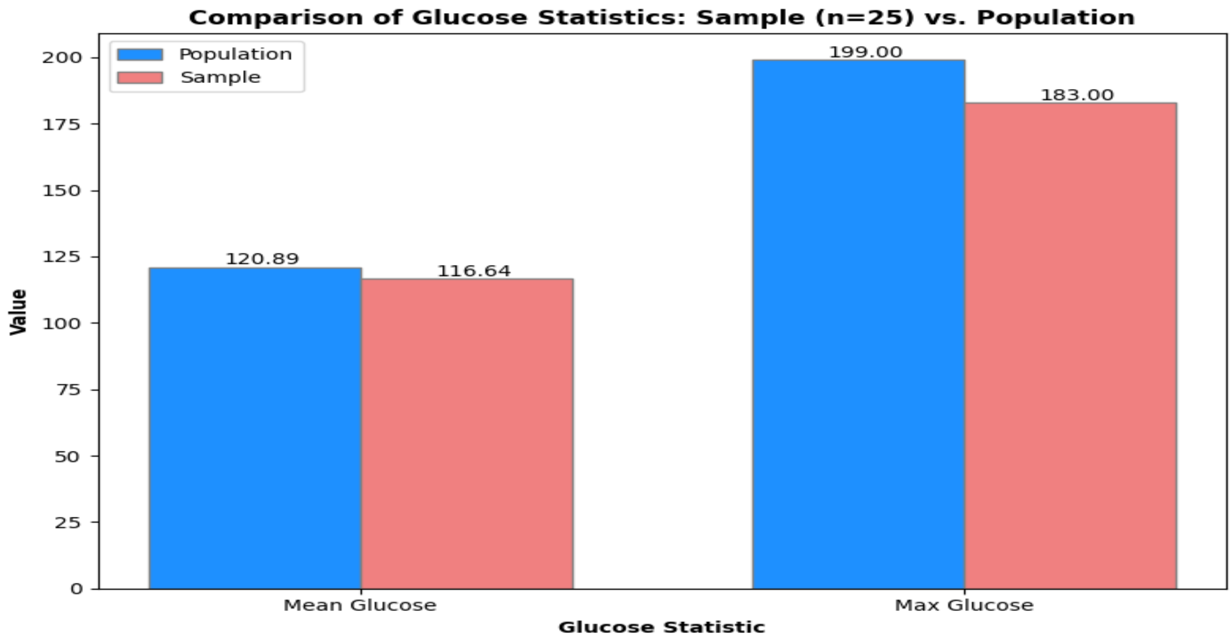
```
4. MUTATE: New 'Power_to_Engine_Ratio' feature added
    Power  Engine  Power_to_Engine_Ratio
0  126.20  1582.0               0.079772
1   88.70  1199.0               0.073978
2   88.76  1248.0               0.071122

5. ARRANGE: Top 3 most expensive cars (sorted by Price desc, then Age asc)
                                         Name  Price  Car_Age
3952  Land Rover Range Rover 3.0 Diesel LWB Vogue  160.0        8
5620                    Lamborghini Gallardo Coupe  120.0       14
5752                        Jaguar F Type 5.0 V8 S  100.0       10

6. SUMMARIZE with GROUP BY: Aggregated statistics by Owner Type
       Owner_Type  Avg_Price  Max_Car_Age  Min_Car_Age  Total_Cars
0           First  10.105076           27            6        4811
1  Fourth & Above   3.415000           20           15           8
2          Second   7.839719           26            7         925
3           Third   5.348058           27           10         103
```

Exercise 2:

a) Set a seed (to ensure work reproducibility) and take a random sample of 25 observations and find the mean Glucose and highest Glucose values of this sample and compare these statistics with the population statistics of the same variable. You should use charts for this comparison.



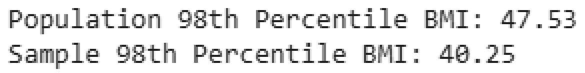
```
       Statistic  Population  Sample
0   Mean Glucose  120.894531  116.64
1    Max Glucose  199.000000  183.00
```



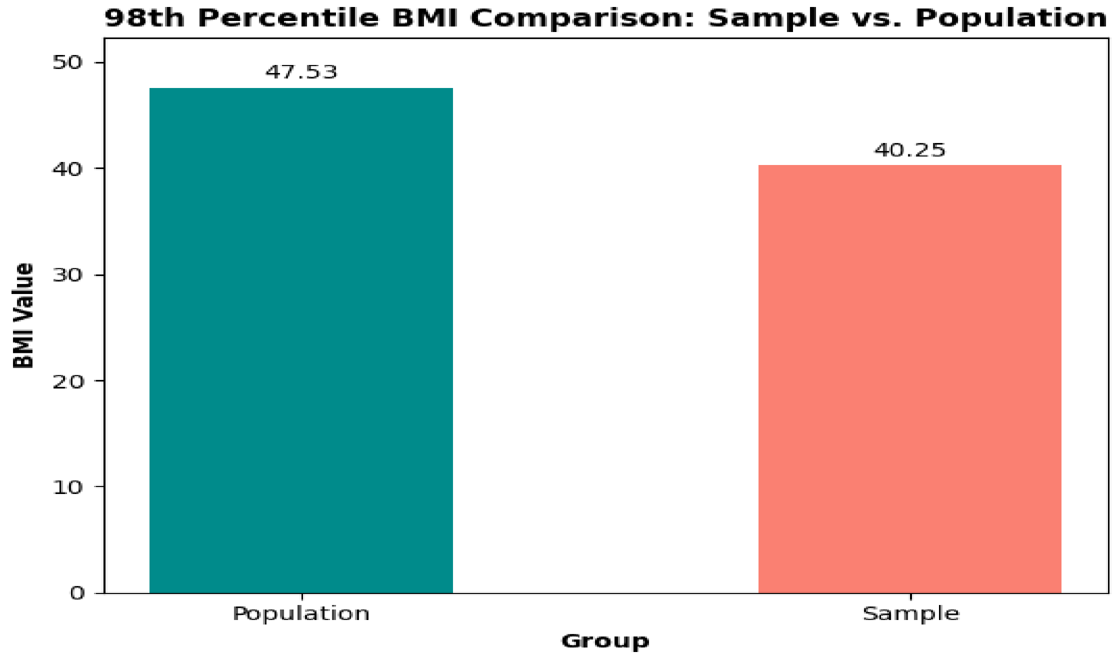
The sample mean for Glucose was calculated to be 116.64, while the true population mean is 120.89. The sample mean provided an estimate that is **reasonably close to the true population value,** with a small difference of 3.5%.

The difference between the population Highest Glucose Value (Max) and the sample Highest Glucose Value is 8%, shows that a small sample is **not sufficient** to capture extreme outliers or rare values present in the total population.

b) Find the 98th percentile of BMI of your sample and the population and compare the results using charts.

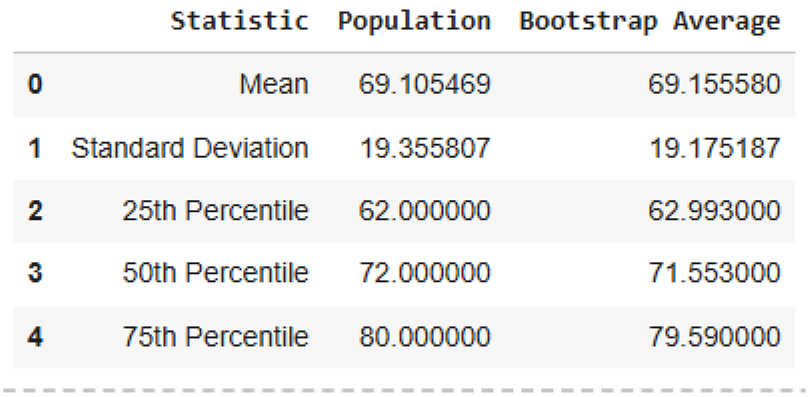


```
Population 98th Percentile BMI: 47.53
Sample 98th Percentile BMI: 40.25
```

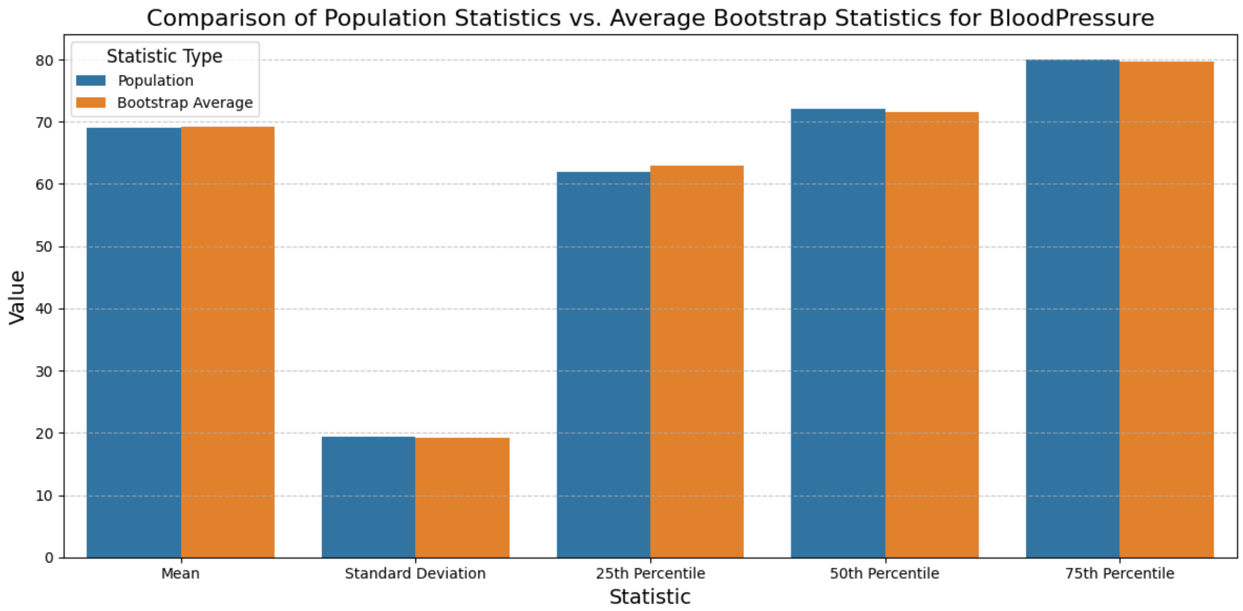


The small sample **severely underestimated** the 98th percentile of the BMI distribution (15.3% error). This confirms that small samples are highly unreliable for estimating **extreme quantiles** or the tails of a distribution.

c) Using bootstrap (replace= True), create 500 samples (of 150 observation each) from the population and find the average mean, standard deviation and percentile for Blood Pressure and compare this with these statistics from the population for the same variable. Again, you should create charts for this comparison. Report on your findings



	Statistic	Population	Bootstrap Average
0	Mean	69.105469	69.155580
1	Standard Deviation	19.355807	19.175187
2	25th Percentile	62.000000	62.993000
3	50th Percentile	72.000000	71.553000
4	75th Percentile	80.000000	79.590000



The average bootstrap means 69.16 is extremely close to the population, which means 69.11. Approximately 0.07% error

The average bootstrap median of 71.55 is very close to the population median of 72.00

The average bootstrap 25th percentile and 75 percentiles are also closer to the population of 25th percentile and 75 percentiles.

By repeatedly resampling, the Bootstrap effectively simulates drawing many samples. This technique removes the high sampling error bias inherent in a single small sample, allowing us to confidently estimate the true mean, median, and spread of the population.