

IST652 – Scripting for Data Analysis

Prof.: Dr. Landowski

Topic: Structured Data

**Name: Yehuda Perry**

**Course: IST652 – Scripting for Data Analysis**

**Professor: Dr. Landowski**

**Assignment: 1**

**Date: 04/16/2020**

Table of Contents

[1. Introduction 3](#_Toc37915542)

[2. Assignment Details 5](#_Toc37915543)

[3. Analysis and Models 7](#_Toc37915544)

[3.1 About the Data 7](#_Toc37915545)

[3.2 Uploaded Libraries (Google Colab) 7](#_Toc37915546)

[3.3 Uploading dataset & use of data frame 8](#_Toc37915547)

[3.4 View and analyze the data 8](#_Toc37915548)

[3.5 Dropping un-necessary Columns 9](#_Toc37915549)

[3.6 Renaming Columns 10](#_Toc37915550)

[3.7 Cleansed Data Frame 10](#_Toc37915551)

[3.8 Tune the data Due to Outliers 10](#_Toc37915552)

[3.9 Questions, Answers and Results 11](#_Toc37915553)

[3.9.1 Question 1: Promotions and Sum Amount of Donations 11](#_Toc37915554)

[3.9.2 Question 2: Number of Promotions vs. Avg. Amount of Donations 12](#_Toc37915555)

[3.9.3 Question 3: Number of Promotions vs. the Time Since the Last Donation 13](#_Toc37915556)

[3.9.4 Question 4: Number of Promotions vs. the Time Bet. First and Sec. Donation 14](#_Toc37915557)

[3.9.5 Question 5: Number of Promotions vs. Donation Frequency 15](#_Toc37915558)

[3.9.6 Question 6: Income vs. Donation Amount 16](#_Toc37915559)

[4. Conclusions 17](#_Toc37915560)

[5. References 18](#_Toc37915561)

# Introduction

Data Structures are a specialized means of organizing and storing data in computers in such a way that we can perform operations on the stored data more efficiently. Data structures have a wide and diverse scope of usage across the fields of Computer Science and Software Engineering. Data structures are being used in almost every program or software system that has been developed. Moreover, data structures come under the fundamentals of Computer Science and Software Engineering. It is a key topic when it comes to Software Engineering interview questions. Therefore, as developers, it is essential to have good knowledge about data structures. As data structures are used to store data in an organized form, and since data is the most crucial entity in computer science, the true worth of data structures is clear. There are eight commonly used data structured: Arrays, Stacks, Queues, Linked Lists, Trees, Graphs, heaps (they are effectively trees, but it is still good to call them out separately) and Hash tables. An array is the simplest and most widely used data structure. Other data structures like stacks and queues are derived from arrays. There are two types of arrays: One-dimensional arrays and Multi-dimensional arrays (arrays within arrays). Basic operation on arrays: Insert, inserts an element at a given index, Get, Returns the element at a given index, Delete, deletes an element at a given index and Size, Gets the total number of elements in an array. A stack is a LIFO (Last in First Out, the element placed at last can be accessed at first) structure which can be commonly found in many programming languages. This structure is named as “stack” because it resembles a real-world stack, a stack of plates. A queue is a FIFO (First in First Out, the element placed at first can be accessed at first) structure which can be commonly found in many programming languages. This structure is named as “queue” because it resembles a real-world queue, people waiting in a queue. A linked list is a sequential structure that consists of a sequence of items in linear order which are linked to each other. Hence, it is a must have to access data sequentially and random access is not possible. Linked lists provide a simple and flexible representation of dynamic sets. A tree is a hierarchical structure where data is organized hierarchically and are linked together. This structure is different than a linked list whereas, in a linked list, items are linked in a linear order. Various types of trees have been developed throughout the past decades, in order to suit certain applications and meet certain constraints. Some examples are binary search tree, B tree, treap, red-black tree, splay tree, AVL tree and n-ary tree. A binary search tree (BST), as the name suggests, is a binary tree where data is organized in a hierarchical structure. This data structure stores values in sorted order. A graph consists of a finite set of vertices or nodes and a set of edges connecting these vertices. The order of a graph is the number of vertices in the graph. The size of a graph is the number of edges in the graph. Two nodes are said to be adjacent if they are connected to each other by the same edge. A Heap is a special case of a binary tree where the parent nodes are compared to their children with their values and are arranged accordingly. A Hash Table is a data structure that stores values which have keys associated with each of them. Furthermore, it supports lookup efficiently if we know the key associated with the value. Hence it is very efficient in inserting and searching, irrespective of the size of the data.

# Assignment Details

You can choose to complete this assignment by yourself or with a group of at most two total participants. Each person must turn the assignment in for grading, and each person must contribute to the development of the program. Use the file Donors\_Data.csv.

Structured Data Processing

For purposes of this writeup, we will use examples from the Donors data file.

The main outline of your assignment is to write a program that will read in the data from a file, such as the .csv file saved from Excel. This will be in a format that is structured with lines of data representing one type of unit (i.e., one donor in the donors file). Your program will represent the data as Python data structures. You may choose for the overall structure to be one or both of the following:

A list of dictionaries, or some combination of lists, dictionaries, and NumPy arrays

A pandas data frame

You will do data exploration and cleaning on this data.

The program will do some processing to convert the data to a form that will answer at least two questions as described below, and write files with the data suitable for answering each question. Graphing is optional.

**Data**

You may choose a data set to work with. As a guideline, data sets should have somewhere between 500 and 4,000 lines of data with some number of columns between 4 and 50. These guidelines are not exact limits, just guidance for selecting data.

If the data comes in an Excel spreadsheet with a lot of columns, it is okay to first edit the file to remove columns that you do not need for your processing. For example, in the Donors data, you might wish to create a separate spreadsheet with only a few columns of data.

**Questions**

For this assignment, at least one question that you choose to answer should look at the data in a different unit of analysis than is present in the data file. For example, instead of looking at individual donors, you could look at the donors of each of the nine income or wealth types.

Simplest example question (you should do one more complex than this):

For each wealth type, what is the average home value of all the donors of that type?

Unit of analysis: wealth types

Comparison: for each wealth type, compute the average home values of the neighborhoods of all the donors of that type

**Output**

should be in a file with nine rows of data (you may also produce header and label rows), where each row has an income type (1–9) and the average home values

One way to increase the complexity of this particular question would be to add more items to be compared to the income types, e.g., add columns to the output with average total gifts or values of the last gifts. Another way is to introduce a more detailed unit of analysis; for example, suppose that for each income level you reported by gender, giving the average home value for both men and women in each category.

**Other ideas**

Compare donors in the various zip codes with various types or amounts of giving.

Compare donors by the number of promotions with the total amount of donations and the frequency of donations.

Compare the number of months since the last donation to the donation amounts.

**What to Submit**

In addition to the program that you write, you should write a small report. In it you should provide:

Data and its source

Description of your data exploration and data cleaning steps

At least two clearly stated comparison questions with the unit of analysis, the comparison values, and how they are computed

Brief description of the program

Description of the output files

For your program, you may use any of the code developed in class as a template, but it is absolutely essential that you use appropriate variable names and that you write original comments for what your program does. Recall that good comments demonstrate your understanding of the code that you write and the problem that you are trying to solve.

# Analysis and Models

## About the Data

**Dataset name:** ‘donoros\_data’

**Dataset format:** CSV format

**Number of columns and rows:** The dataset contain 24 columns and 3120 rows.

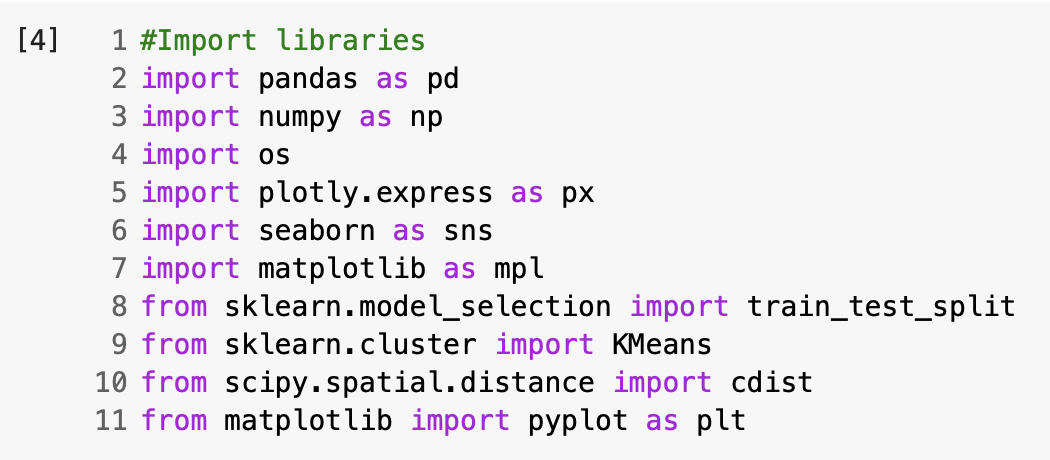
**Attributes Names**: ‘Row Id’, ‘Row id.’, ‘zipconvert\_2’, ‘zipconvert\_3’, ‘zipconvert\_4’, ‘zipconvert\_5’, ‘homeowner dummy’, ‘NUMCHLD’, ‘INCOME’, ‘gender dummy’, ‘WEALTH’, ‘HV’, ‘Icmed’, ‘Icavg’, ‘IC15’, ‘NUMPROM’, ‘RAMNTALL’, ‘MAXRAMNT’, ‘LASTGIFT’, ‘totalmonths’, ‘TIMELAG’, ‘AVGGIFT’, ‘TARGET\_B’, ‘TARGET\_D’

**Data Dictionary:** Data Dictionary was provided as a part of ‘donors\_data’ dataset.

## 

## Uploaded Libraries (Google Colab)

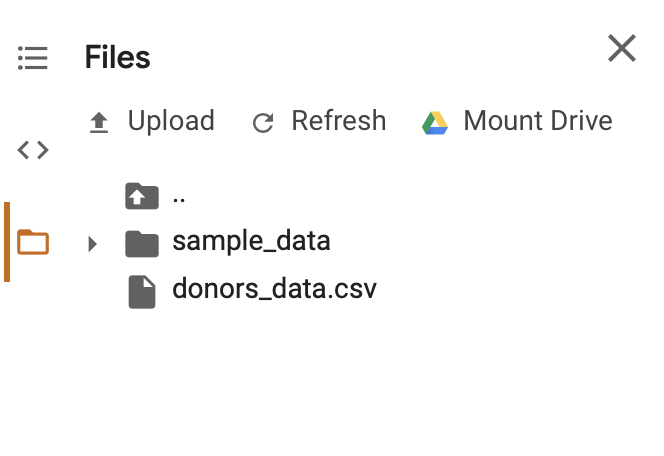
The following libraries were uploaded to support the python code and dataset analysis:



**Figure 1: Uploaded Libraries**

## Uploading dataset & use of data frame

The dataset manually loaded to ~/content Google Colab folder and data frame was assigned.



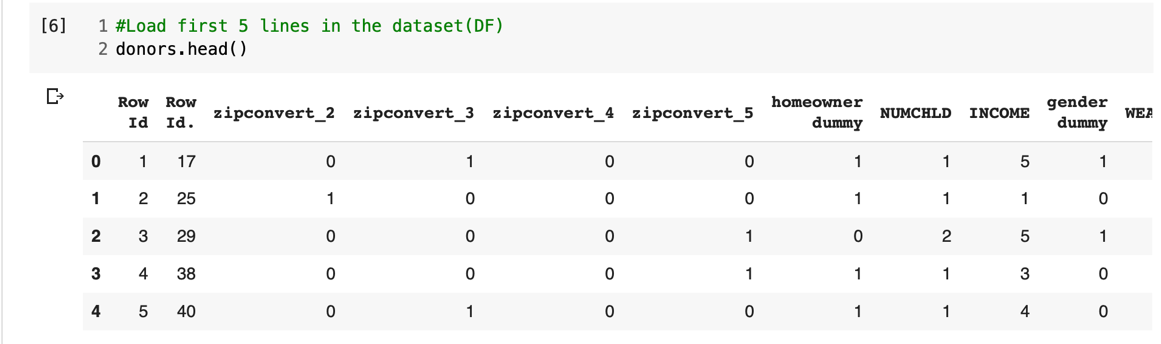
**Figure 2: Dataset loaded to ~ /Content folder**



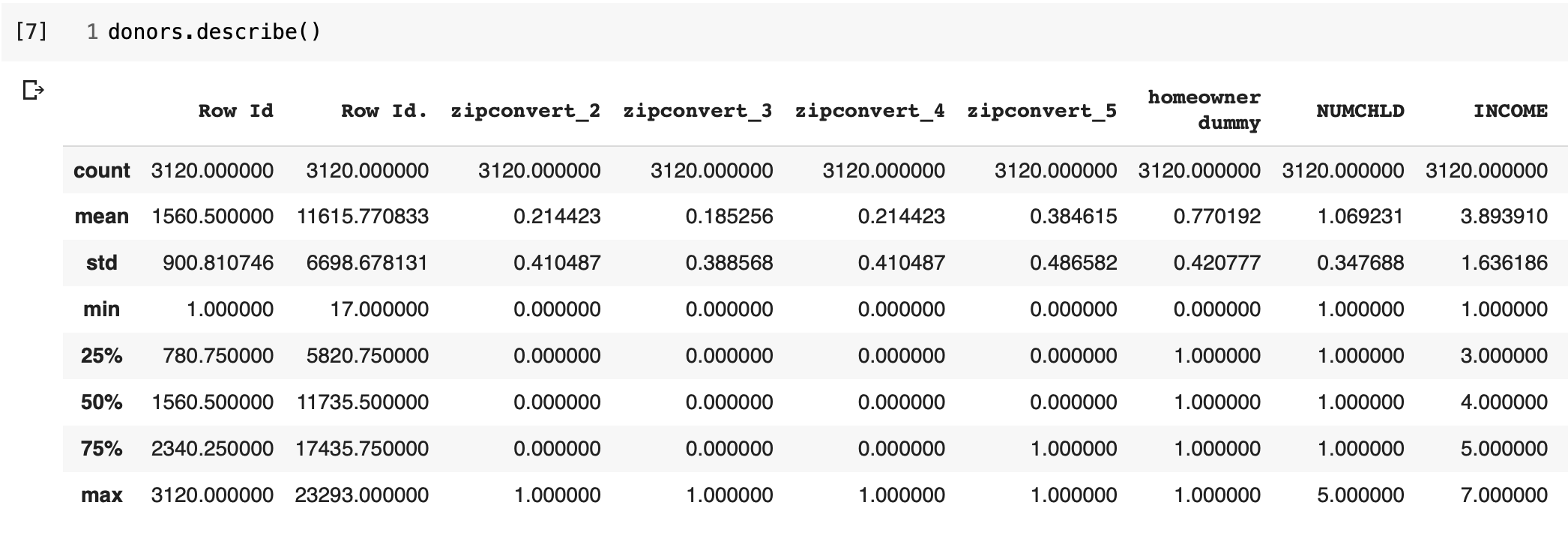
**Figure 3: Data frame ‘donors’**

## View and analyze the data

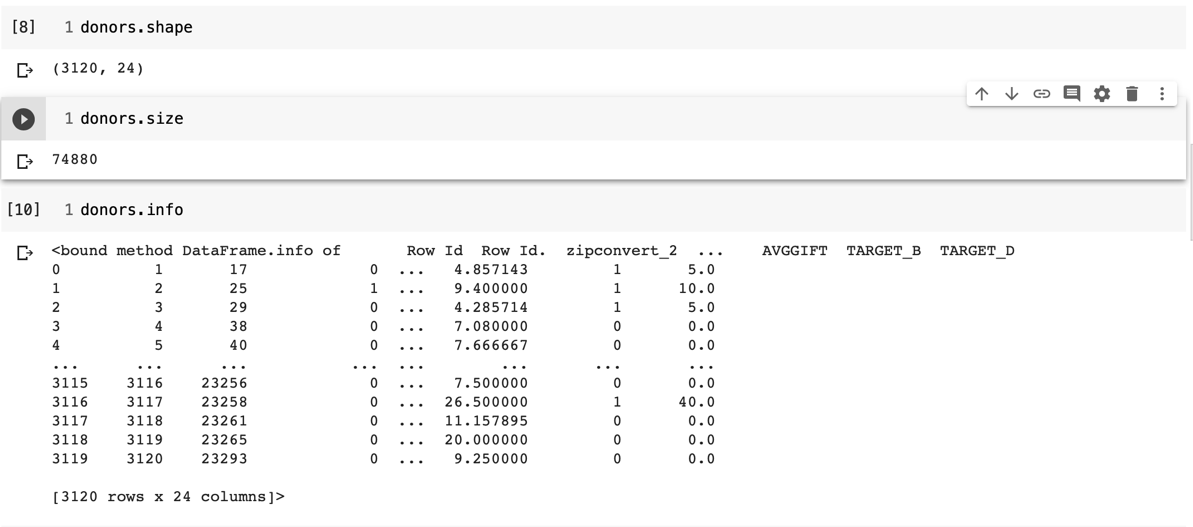
Using the following functions (head, describe, shape, size and info) to make sure the data was loaded successfully and perform simple data analysis:



**Figure 4: head() function**



**Figure 5: describe function**



**Figure 6: shape, Size and Info**

## Dropping un-necessary Columns

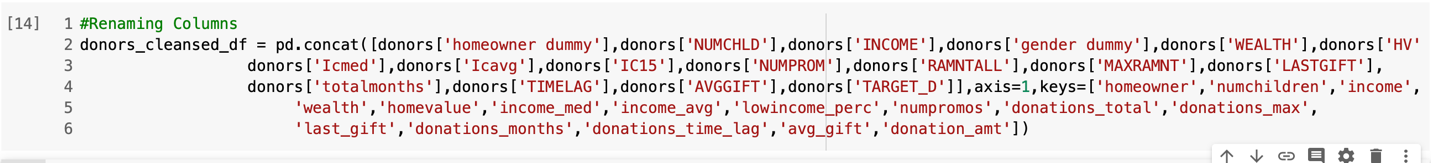
In order to arrange correctly the attributes, I decided to drop some of the columns that I will not use.



**Figure 7: Dropping Columns**

## Renaming Columns

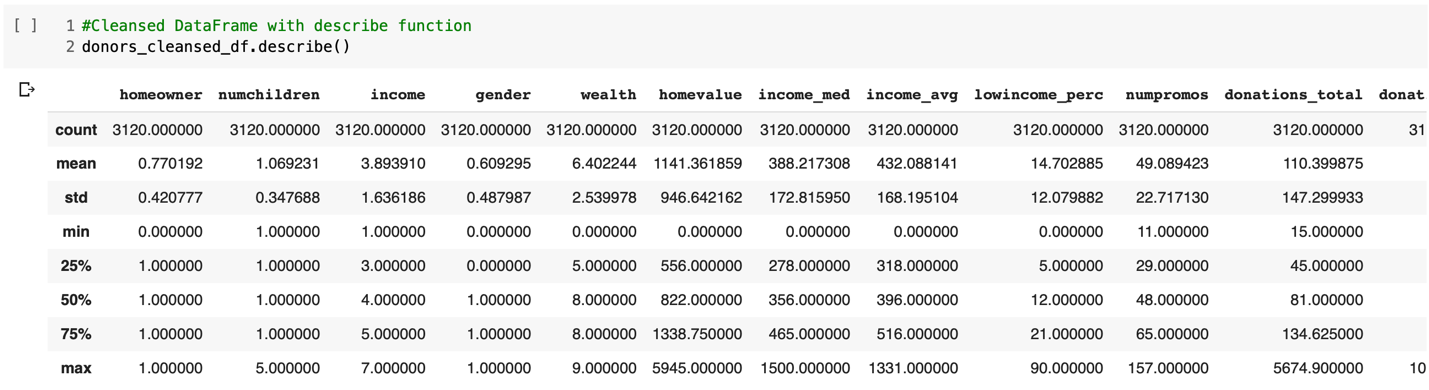
I renamed the columns to be able to read them better.



**Figure 8: Renaming Columns**

## Cleansed Data Frame

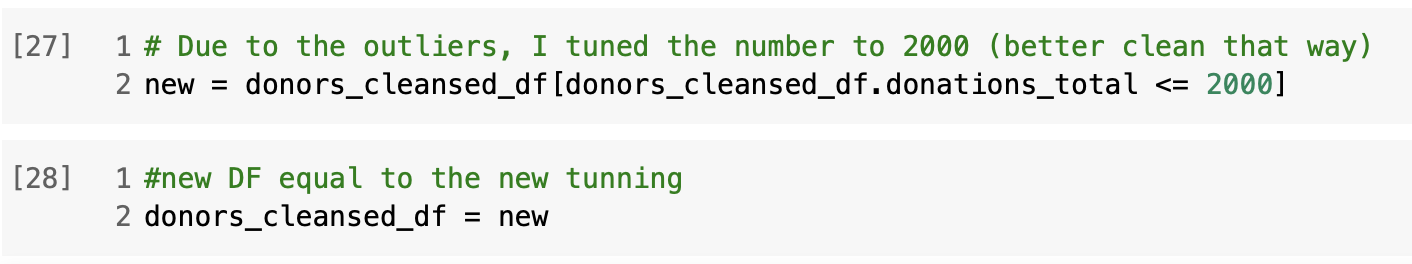
After renaming the data frame, I created a new data frame, clean one to work with named: ‘donors\_cleansed\_df.I run describe function to make sure the new data frame is working and to make sure the data is working properly according to the changes I made.



**Figure 9: New data frame with describe function**

## Tune the data Due to Outliers

Tune the data due to outliers to 2000. Later, I will use the new DF to be equal to the new tuning.



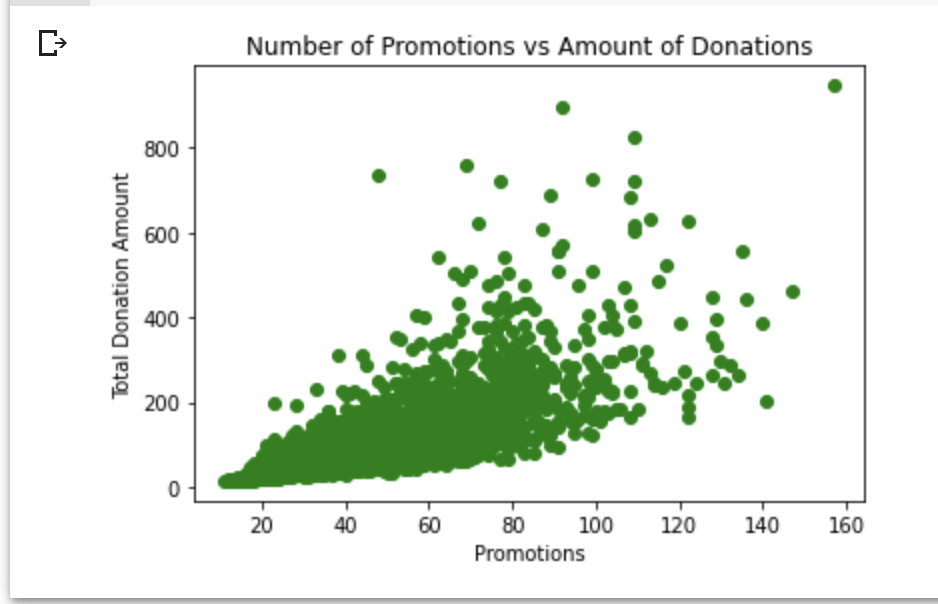
**Figure 10: Tune the data due to outliers**

## Questions, Answers and Results

## Question 1: Promotions and Sum Amount of Donations

**Question 1: Does the number of promotions really impact the sum amount of donations?**

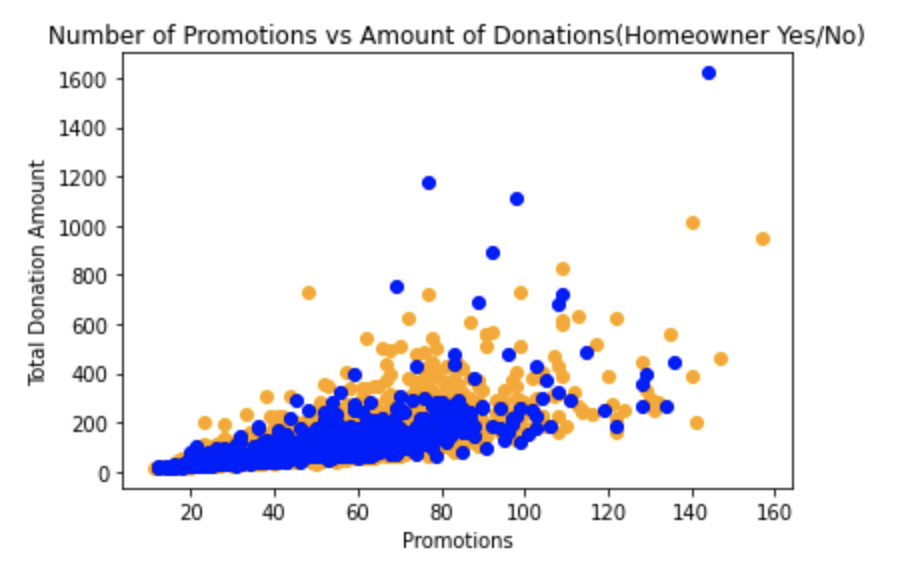
In order to answer this question, I had to plot the number of promotion attribute with the total sum of the donations.



**Figure 11: Number of Promotions vs. Amount of Donations**

In figure 11, we can observe that more promotions are being produced, more donations are being received and in a linear way and there is a direct connection between promotions and total donations amounts.

**Total donation amount to the number of promotions:**

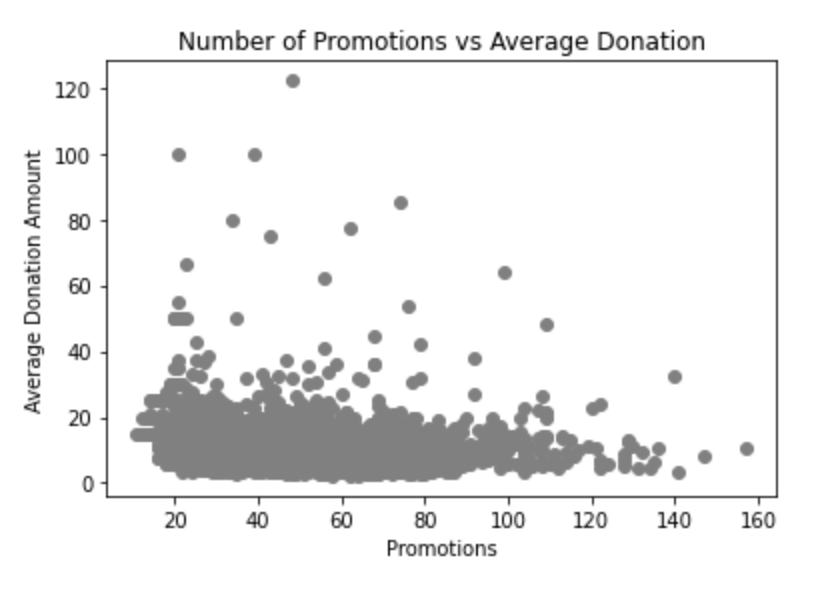


**Figure 12: Total donation amount to the number of promotions**

This graph describing the relationship between homeowners and people that doesn’t own home by promotions and amount of donations. We can defiantly identify that homeowner donate more and also high amounts of donations.

## Question 2: Number of Promotions vs. Avg. Amount of Donations

**Question 2: Does the number of promotions influence the average amount of donations?**

****

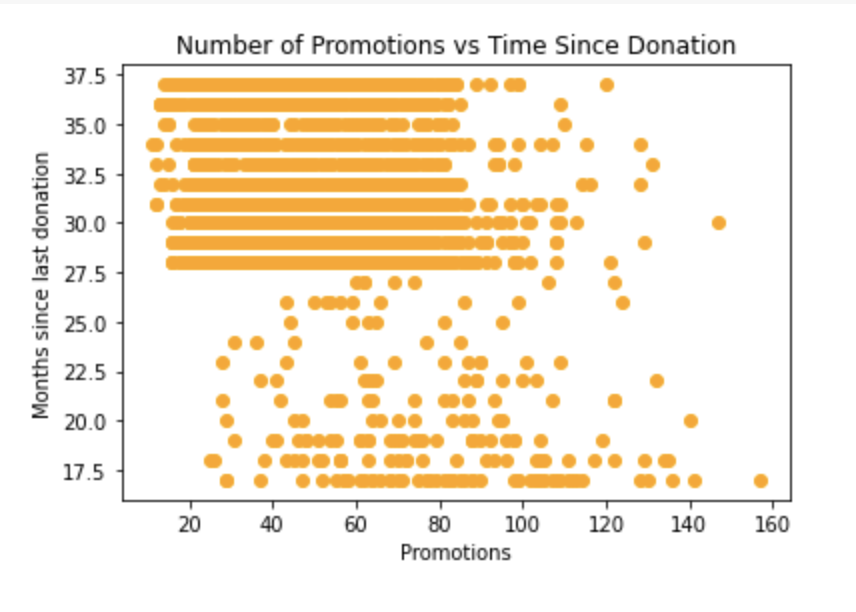
**Figure 13: Number of promotions vs. Average donations**

In the above graph we can see the number of promotions conducted and

according to that the average the average donation amount. The graph is indicating that the average amount of donation is steady regardless to the number of promotions.

## Question 3: Number of Promotions vs. the Time Since the Last Donation

**Question 3: Does the number of promotions influence the time since the last donation?**

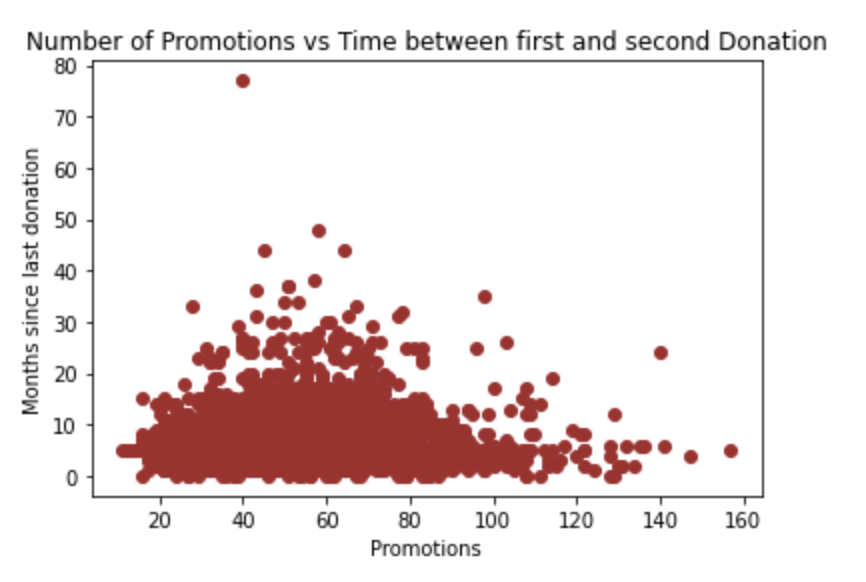


**Figure 14: Number of promotions vs. Time Since Donations**

In the graph we can see the number of promotions vs. the months since last donation. We can say that in some way the number of promotions does influence the months since the last donation. If more promotions will be conducted, than the number of moths since last donation will probably increase.

## Question 4: Number of Promotions vs. the Time Bet. First and Sec. Donation

**Question 4: Does the number of promotions influence the time between first and second donation?**

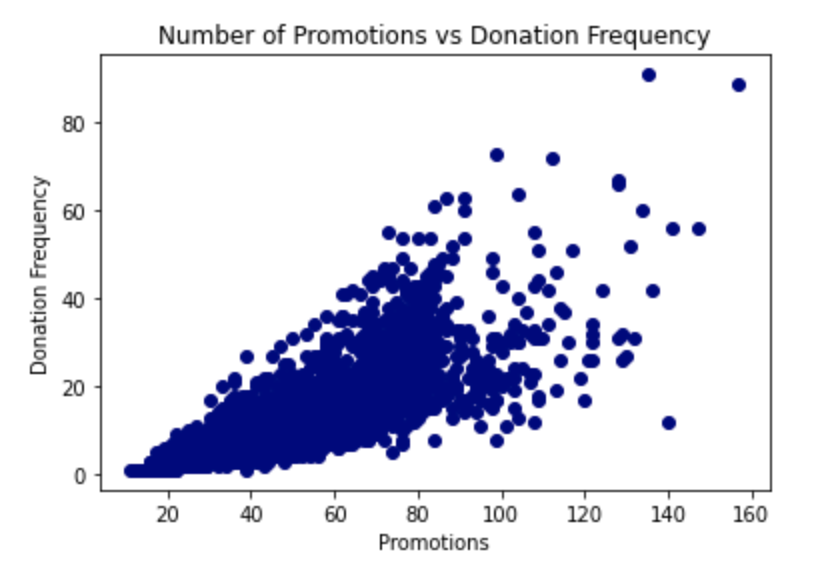
****

**Figure 15: number of promotions vs. the time between first and second donation**

In the above graph we see the number of promotions vs. the time between first and second donation. Yes. There is a correlation between the number of promotions to the time between the first and second donation.

## Question 5: Number of Promotions vs. Donation Frequency

**Question 5: Does the number of promotions influence on donation frequency?**

****

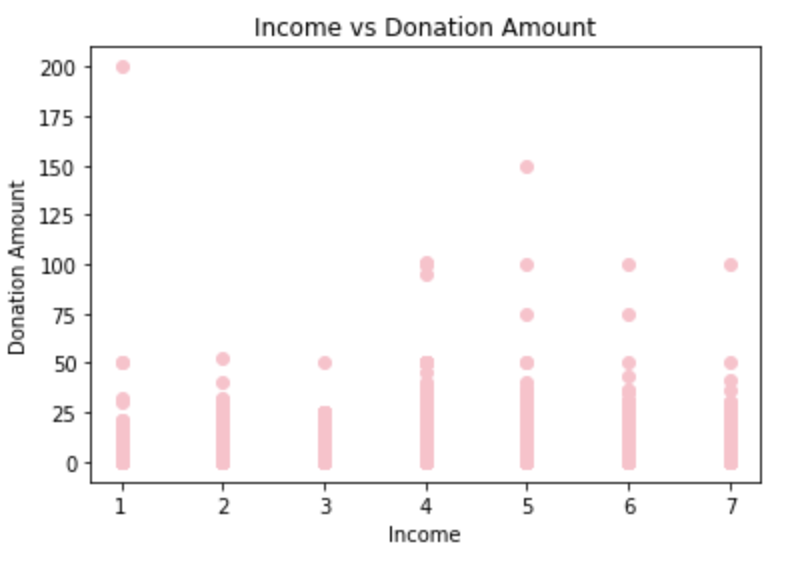
**Figure 16: number of promotions vs. donation frequency**

In this graph we see the number of promotions vs. donation frequency.

Yes. there is a correlation between the number of promotions and donation frequency. The number of promotions increase the number of donation frequency increase as well.

## Question 6: Income vs. Donation Amount

**Question 6: Is there any relationship/correlation between Income and Donation Amount?**

****

**Figure 17: relationship between income and donation amount**

In this graph we can see the relationship between the income and donation amount. What was interesting to find out that the highest donation made was done by the lowest income. Nonetheless, the averageof income 5 are the highest which reflect a little bit more than an average to donate the second highest amount.

# Conclusions

Every programming language has provision for data structures. In the Python programming language, there are a total of 4 inbuilt data structures. These are namely **list, tuple, dictionary, and set**. Each of them is unique in its own right. Learning Python is incomplete without building a good understanding of data structures.

From the exercise conducted and the questions asked above, we clearly can identify all answers by reviewing and analyzing the different graphs.

In addition, using of graphs to conduct comparison seems to be a good method to evaluate two or more attributes. In this work I have choose to ask five questions, however in a given dataset similar to donors, the researchers are able to perform any kind of analysis and tens of questions could be asked for different analysis.

# References

Data Structures for Statistical Computing in Python:

<http://conference.scipy.org/proceedings/scipy2010/pdfs/mckinney.pdf>

Data Structures and Algorithms in Python:

<https://passtest.eu/sample/Complete-Solution-Manual-for-Data-Structures-and-Algorithms-in-Python-by-Michael-T-Goodrich-Roberto-Tamassia-Michael-H-Goldwasser-EHEP002510.pdf>