Israel Nolazco

Concepts of Statistics 2

SU20-Data-51200-002

July 19, 2020

Examining Data

In this exercise, we are going to convey the importance of multivariate analysis given the data set provided by the HBAT.xls file. We are going to run through different scenarios in which Multivariate Analysis will unravel several key details by our analysis. Additionally, in this exercise, we are going to utilize Python as our main tool.

As seen below, we are going to start by importing different sets of libararies that will help us read the data file, modifiy said data file and ultimately plot different dataset that will reveal the deeper meaning behind our dataset.

In [1]:
```
import pandas as pd
from pandas import Series, DataFrame
import numpy as np
import matplotlib as plt
import seaborn as sns
%matplotlib inline

df_data = pd.read_excel(r'HBAT.xls')

df = DataFrame(df_data)

df.head()
```

Out[1]:

	id	x1	x2	x3	x4	x5	x6	x7	x8	x9	...	x14	x15	x16	x17	x18	x19	x20	x21	x22	x23
0	1	2	0	1	1	1	8.5	3.9	2.5	5.9	...	4.7	4.3	5.0	5.1	3.7	8.2	8.0	8.4	65.1	1
1	2	3	1	0	0	0	8.2	2.7	5.1	7.2	...	5.5	4.0	3.9	4.3	4.9	5.7	6.5	7.5	67.1	0
2	3	3	0	1	1	1	9.2	3.4	5.6	5.6	...	6.2	4.6	5.4	4.0	4.5	8.9	8.4	9.0	72.1	1
3	4	1	1	1	1	0	6.4	3.3	7.0	3.7	...	7.0	3.6	4.3	4.1	3.0	4.8	6.0	7.2	40.1	0
4	5	2	0	1	0	1	9.0	3.4	5.2	4.6	...	6.1	4.5	4.5	3.5	3.5	7.1	6.6	9.0	57.1	0

5 rows × 24 columns

After importing our libraries the next step is to read out HBAT.xls file by using the command pd.read_excel. This command will grab the file and save it in a reference name of the df_Dataframe. Then, we are going to view only the top five rows of this dataset in order to visually graphs what our data consist of. As such, we can immediately discover columns "id" and "x22" are the outlier, meaning their information will serve no meaning to our analysis or their values are off the scale that will skew our analysis. Thefore, in the following cell we are going to remove the aforementioned columns. Additionally, we are going to save this new dataframe with the variable "df2"

In [2]:
```
df2 = df.drop(columns=['id','x22'])
```

In [3]:
```
df2.head()
```

Out[3]:

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	...	x13	x14	x15	x16	x17	x18	x19	x20	x21	x23
0	2	0	1	1	1	8.5	3.9	2.5	5.9	4.8	...	6.8	4.7	4.3	5.0	5.1	3.7	8.2	8.0	8.4	1
1	3	1	0	0	0	8.2	2.7	5.1	7.2	3.4	...	5.3	5.5	4.0	3.9	4.3	4.9	5.7	6.5	7.5	0
2	3	0	1	1	1	9.2	3.4	5.6	5.6	5.4	...	4.5	6.2	4.6	5.4	4.0	4.5	8.9	8.4	9.0	1
3	1	1	1	1	0	6.4	3.3	7.0	3.7	4.7	...	8.8	7.0	3.6	4.3	4.1	3.0	4.8	6.0	7.2	0
4	2	0	1	0	1	9.0	3.4	5.2	4.6	2.2	...	6.8	6.1	4.5	4.5	3.5	3.5	7.1	6.6	9.0	0

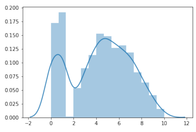
5 rows × 22 columns

Distribution Analysis

In order to get an adequate perspective of the variables in this dataset we are going to use a Histogram; which is a graphical representation of a single variable that represents the frequency of occurrence within the data category. Therefore, we are using our command sns.distplot(df2) to get this graph in Python.

In [4]:
```
sns.distplot(df2)
```

Out[4]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2e3883250>
```



We quickly realize that there are additional pieces of data that are still affecting the analysis of our distribution. These data points affecting our distribution are coming from a range of 0 to 1. Those values are typically coming from columns x1, x2, x3, x4, x5 and x23. Therefore, we are going to remove those columns and run our Histogram once more. Lastly, we are saving this new datset as variable "df3."

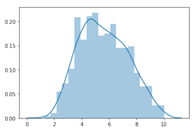
In [5]:
```
df3 = df2.drop(columns=['x1','x2','x3','x4','x5','x23'])

df3.head()
```

Out[5]:

	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21
0	8.5	3.9	2.5	5.9	4.8	4.9	6.0	6.8	4.7	4.3	5.0	5.1	3.7	8.2	8.0	8.4
1	8.2	2.7	5.1	7.2	3.4	7.9	3.1	5.3	5.5	4.0	3.9	4.3	4.9	5.7	6.5	7.5
2	9.2	3.4	5.6	5.6	5.4	7.4	5.8	4.5	6.2	4.6	5.4	4.0	4.5	8.9	8.4	9.0
3	6.4	3.3	7.0	3.7	4.7	4.7	4.5	8.8	7.0	3.6	4.3	4.1	3.0	4.8	6.0	7.2
4	9.0	3.4	5.2	4.6	2.2	6.0	4.5	6.8	6.1	4.5	4.5	3.5	3.5	7.1	6.6	9.0

In [6]:
```
sns.distplot(df3)
```

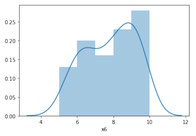
Out[6]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2e2fd9810>
```



As we can see from the above image, our new data frame df3 is visually close to that our Figure 2-1 as shown in book Multivariate Data Analysis Chapter 2, section 1, Page 38. However, keep in mind that the graph itself is relative to column "x6" rather than taking into consideration off the whole datset. The image below is going to represent solely column "x6."

In [7]:
```
sns.distplot(df3['x6'])
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2e3a6b050>
```



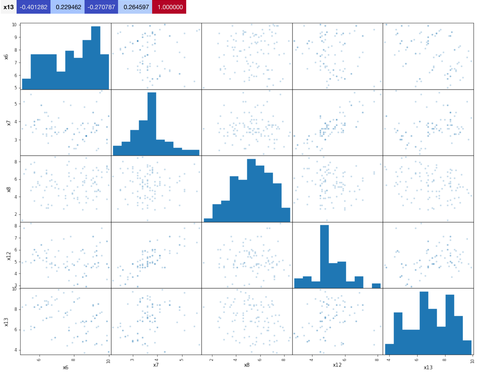
Scatter Plot Matrix

Examaning the relationship between variables could lead analysts to understand behavior and predict change within a group. Typically, the best way to do so is by using a scatterplot graph. However, the best-suited format for multivariate analysis is the scatterplot matrix in which the diagonal contains histograms of the variables. In this exercise, we are primarily focusing on columns x6, x7, x8, x12 and x13.

In [8]:
```
from pandas.plotting import scatter_matrix
df4 = pd.DataFrame(df, columns = ['x6', 'x7', 'x8', 'x12','x13'])
scatter_matrix(df4, alpha = 0.3, figsize = (16, 12), diagonal = 'hist')

corr = df4.corr()
corr.style.background_gradient(cmap='coolwarm')
```

Out[8]:

	x6	x7	x8	x12	x13
x6	1.000000	-0.137163	0.095600	-0.151813	-0.401282
x7	-0.137163	1.000000	0.000867	0.791544	0.229462
x8	0.095600	0.000867	1.000000	0.016991	-0.270787
x12	-0.151813	0.791544	0.016991	1.000000	0.264597
x13	-0.401282	0.229462	-0.270787	0.264597	1.000000



In order to completely show the correlation between each column, we have utilized the command df4.corr() along with the scatter_matrix(). In the lower-left of the graphs, we have a scatter plot representing all combinations of the variables. The diagonal contains histograms of the variables. Lastly, the graph without any graphs, but just numbers show the correlation matrix. In which we can see between x7 and x12 they have the highest correlations.

Blox Pot

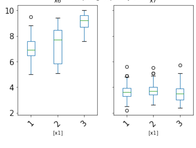
Thus far in our analysis, we have gathered an understanding of what variables our data is using, what variables are similar within categories, but what about differences? The best way to figure out the differences is by assessing group differences is through univariate analyses such as t-test and analysis of variance. The graphical method to do so is using the boxplot. Following the information given by the book Multivariate Data Analysis, Chapter 2, Section1 on Page 40. Our column x1 represents the customer type, x6 represents Product Quality and x7 E-commerce Activities. Therefore, we are assessing the group difference between our customer types to the Product Quality and E-commerce Activities with respect to customer type. Below, we have isolated the aforementioned columns with the variable df5. Additionally, in this excersice we are using the command boxplot.

In [9]:
```
df5 = pd.DataFrame(df, columns = ['x1', 'x6', 'x7'])

df5.head()
```

Out[9]:

	x1	x6	x7
0	2	8.5	3.9
1	3	8.2	2.7
2	3	9.2	3.4
3	1	6.4	3.3
4	2	9.0	3.4

In [10]:
```
boxplot = df5.boxplot(column=['x6', 'x7'], by=['x1'],fontsize = 15, rot=45, grid=False)
```



Let us first look at the relationship between Customer Type and Product Quality (left image, x1 vs x6). The image shows a difference across all groups. This could indicate that there are significant differences between the customer type and the product quality. Now for the second figure between Ecommerce activities and customer type we see no real difference. However, we do see multiple outliers. This could potentially describe the outliers in our data and with further research figure out what type of effect does this has in the overall analysis. Let us first look at the relationship between Customer Type and Product Quality (left image, x1 vs x6). The image shows a difference across all groups. This could indicate that there are significant differences between the customer type and the product quality. Now for the second figure between Ecommerce activities and customer type we see no real difference. However, we do see multiple outliers. This could potentially describe the outliers in our data and with further research figure out what type of effect does this has in the overall analysis.