**Python for Business Analytics – INFO 4120**

**Mid Term Project**

**Professor: Neba Nfonsang**

**Requirement:** *For this assignment, you are required to use Jupyter Notebook. Follow the directions for each question keenly and insert screenshots and codes where required.*

*Submit both python and word document files as Mid\_Term\_firstName\_LastName*

**Data Cleaning, Data Analysis and Data Visualization**

1. Import the BusinessData (csv file) into your python environment and store the data in a Pandas DataFrame. Then view the first 10 rows of your data

**Insert screenshot of your output and codes here**

1. Run a single line of code that outputs the number of columns and rows in your DataFrame.

**Insert screenshot of your output and codes here**

1. Change the OrderDate to Pandas date format. View the first 5 rows of your data

**Insert screenshot of your output and codes here**

1. A) Use the .isnull () function to verify the missing values on your DataFrame. Then print the first 10 rows of your data **without** using the .head () function. You may **use the .loc () or iloc()** functions. **Do not** store the selected data to any DataFrame, you just want to view. View the first 10 rows of your data.

**Insert screenshot of your output and codes here**

B) Which column has missing values in your DataFrame?

**Put your answer here**

1. Fill the missing values in your DataFrame with the mean value of the data in that column. View the first 10 rows of your data.

**Insert screenshot of your output and codes here**

1. Delete the ProductID and CustomerID columns since we already have the ProductName and Customer columns which could serve the same purpose.

**Insert screenshot of your output and codes here**

1. Add another column to your DataFrame called “TotalSales”. The values of your TotalSales column will be calculated from the values of the UnitPrice and Quantity columns. (Hint: Note that if we have two columns A and B and want to add column C; for a any row, if (value of A)\*(value of B)= (value of C), then the column C can be included to the DataFrame using the code df[“C”]=df.A\*df.B . This code performs element-wise multiplication of columns A and B. Note that element-wise addition, and subtraction can be done in a similar way. View the first 5 rows of your DataFrame

**Insert screenshot of your output and code here**

1. Add another column to your DataFrame called “TotalCost”. The values of your TotalCost column will be calculated from the values of the UnitCost and Quantity columns. (Hint: Use a code that performs an elementwise multiplication of UnitCost and Quantity columns). View the first 5 rows of your DataFrame

**Insert screenshot of your output and code here**

1. Add the final column to your DataFrame called “Profit”. The values of your Profit column will be calculated from the values of the TotalSales and TotalCost columns. (Hint: Use a code that performs an elementwise subtraction of TotalSales and TotalCost columns). View the first 5 rows of your DataFrame

**Insert screenshot of your output and code here**

1. A) Write the names of the columns with categorical data here

B) Write the name of the columns with quantitative data here

C) Using a single line of code, compute the summary statistics of the categorical variables. (Hint: select all the categorical variables and use the .describe () function).

**Insert screenshot of your output and codes here**

D) Using a single line of code, compute the summary statistics of the quantitative variables. (Hint: select all the quantitative variables and use the .describe () function).

**Insert screenshot of your output and code here**

1. Use the .agg function to compute the mean, median, minimum, maximum, variance, and standard deviation values of the quantitative data. (Use a single line of code and do this only for the quantitative data, not for the entire DataFrame).

**Insert screenshot of your output and code here**

1. Use the .sum function to calculate the total profit from all sales transactions.

**Insert screenshot of your output and code here**

1. Plot a time series graph for Total sales. That is, using features of the matplotblib package, plot a graph of OrderDate versus TotalSales.

-Your plot should be formatted to include the title; axes labels and the dates should be vertically orientated (rotation =90).

**Insert screenshot of your output and code here**

1. A) Use the .unique () to see the different Sales Representatives in your DataFrame

**Insert screenshot of your output and code here**

1. Use the .value\_counts() function to count how any business transactions were made by each sales representative. Store your output in an object (or Series) called counts, then store the data in a DataFrame called “counts” using a code such as counts=Pd.DataFrame(counts)

**Insert screenshot of your output and code here**

1. Plot a bar chart for the SalesRep data. The chart should be formatted to have a title, axes labels (**Use matplotblib features**). From the chart, which Sales representative made the highest number of business transactions?

**Insert screenshot of your output and code here**

1. Use the DataFrame in question 14B to create a **Pie chart** showing each representative and the count of transactions the carried out with each product. Use the regular pandas plot function. (That is, do not use matplotblib.pyplot or plt)

**Insert screenshot of your output and code here**

1. A) Use the pd.crosstab() function to find how many sales transactions each sales representative performed with each product type. (Hint: Let the ProductName column and SalesRep column be parameters of your pd.crosstab() function). Store the results of your pd.crosstab() code in a DataFrame call SalesRep\_ProdCount. Print the SalesRep\_ProdCount

**Insert screenshot of your output and code here**

B). Use the DataFrame in question 15A to create a **bar chart** showing each representative and the counts of transactions they carried out with each product. Use the regular pandas plot function. (That is, do not use matplotblib.pyplot or plt)

**Insert screenshot of your output and code here**

C). From visual observation of the bar chart, which sales representative had the highest number of sales transactions with gummi bears?

**Write your answer here**

1. Use the grouby() function to group Quantity by ProductName. You want to see the quantity of each product sold.

**Insert screenshot of your output and code here**

1. Use Boolean selection to select the TotalSales values for the “Pop Rocks” product and store in these TotalSales values in a DataFrame. Then find the mean of this data in the DataFrame.

**Insert screenshot of your output and code here**