**DATA PROCESSING USING PYTHON**

**Sales across country for 7 years**

**DATASET LINK:**

**<https://eforexcel.com/wp/downloads-18-sample-csv-files-data-sets-fo-testing-sales/>**

**ABOUT:**

This is a sales dataset contain details about sales record over the period of 7 years (2010-2017) this displays over all profit and revenue of that particular product.(this file is in CSV format)

Empty values (or null values) are used within the data set to indicate an absence of data, such as missing of ordered date or unknown data. They are not used to indicate a zero value. For example, an unknown ordered date value will be marked as Null.

**TO-DO List:**

1. Relationship between unit sold and Total profit
2. Which sales channel is used more in that particular year
3. A chart for order priority
4. Count of item in the data set
5. Which year has more sales

**DATA CLEANING**

Data cleaning is one of the essential part of data pre-processing and data visualisation. For cleaning of our dataset, we used the python package called pandas. In my data of “Sales Across Country” there are some mismatch data is there and for the removal of that data, we used numpy and pandas’ package of python both.

**Preprocessing done in my dataset:**

1. First we are viewing the info of the data set
2. Formatting the date in the data set
3. Checking the presence of duplication
4. Removing the unwanted column
5. Extracting years from the date column and saving in separate column
6. Saving the cleaned data in a new CSV file

**Package need to be imported:**

Packages like numpy for numerical operations

pandas for preprocessing

Matplotlib for plotting graphs

from turtle import color

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

from collections import Counter

**Reading the data and viewing info:**

Reading the values into a variable db and printing the info using info()

df=pd.read\_csv('sales.csv')

#printing info

print(df.info())

**Formatting the data:**

Formatting the order date and ship date

#fix the formate like date

df['Order Date']=pd.to\_datetime(df['Order Date'])

#fix the formate like date

df['Ship Date']=pd.to\_datetime(df['Ship Date'])

**Checking the presence of duplicate:**

Checking for duplicates and no duplicates found

#checking presence of duplicate

print(df.duplicated())

**Remove unwanted column:**

#removing country

del df['Country']

del df['Unit Price']

**Extracting needed value to new column:**

df['Order Date'] = pd.to\_datetime(df['Order Date'])

df['year'] = df['Order Date'].dt.year

**Saving cleaned data into new csv file:**

#saving in another file

df.to\_csv('cleaned1.csv')

**Visualization of data**

1. This is a plot mapped between unit sold and total profit ,using the visualization we can see the relation between unit sold and total profit

x = df1['Units Sold']

y = df1['Total Profit']

plt.subplot(1,2,1)

plt.plot(x,color='r')

plt.title("Unit Sold")

plt.subplot(1,2,2)

plt.plot(y,color='b')

#title for plot

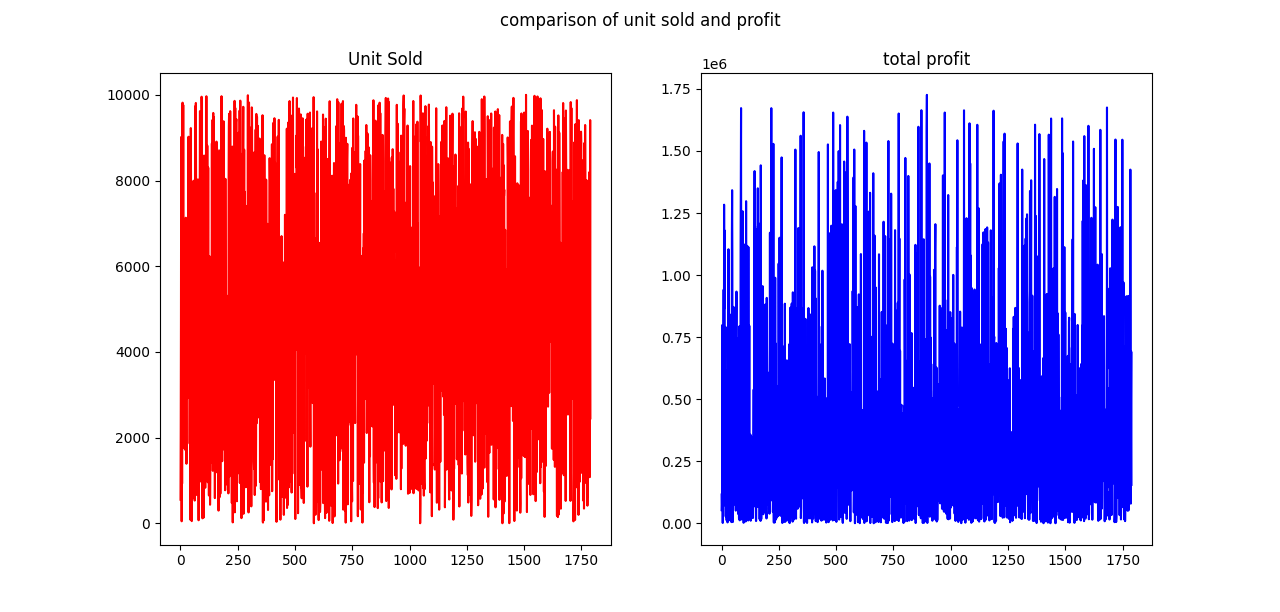
plt.title("total profit")

#over all title

plt.suptitle("comparison of unit sold and profit")

plt.show()

**The Graph of unit sold vs total profit:**



1. This is a bar chart over 7 years(2010-2017) displaying which sales channel(online/offline) is used more in that particular year. If it is online the bar will be in red color and if offline then in Blue color.

a=df1['year']

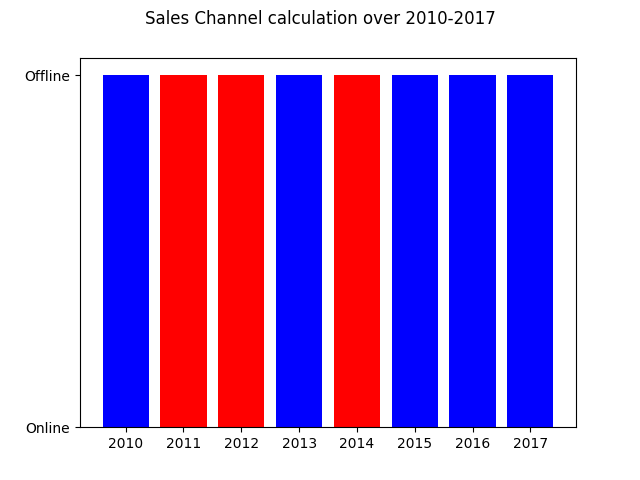
b=df1['Sales Channel']

plt.bar(a,b,color=['r','b'])

plt.suptitle("Sales Channel calculation over 2010-2017")

plt.show()

**Bar chart of year vs sales channel:**



1. A histogram for the order priority so that we can understand which priority is choosed more.

b=df1['Order Priority']

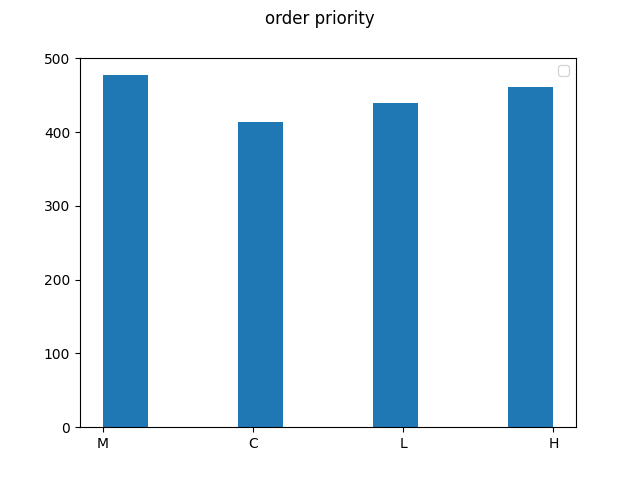
plt.suptitle("order priority")

plt.hist(b)

plt.legend()

plt.show()

**Histogram for order priority:**

****

1. This is a bar chart which display the count of different items,here we are converting the data to dictionary .the key is the item name and values is the count .

a= df1['Item Type']

plt.suptitle("Overall item count")

item\_count = Counter(a)

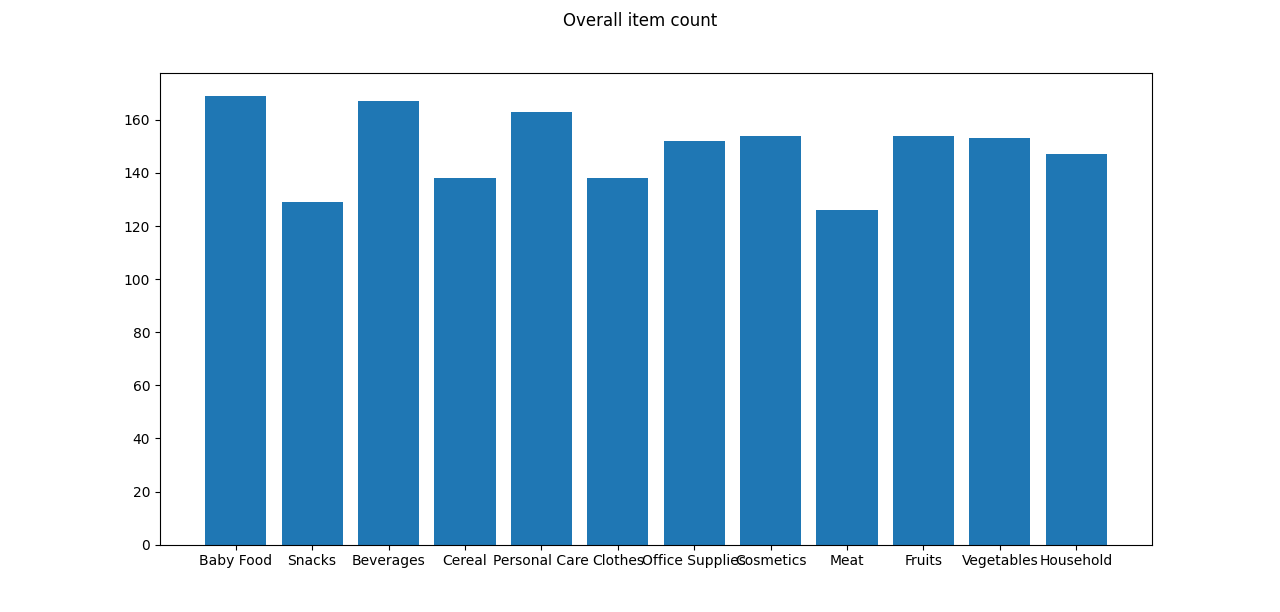
b=item\_count.values()

c=item\_count.keys()

plt.bar(c,b)

plt.show()

**Bar chart for item count:**

****

1. This pie chart display the sales over years ,which helps to understand which year have more sales.

a=df1['year']

count=Counter(a)

b=count.values()

plt.suptitle("pie chart for years")

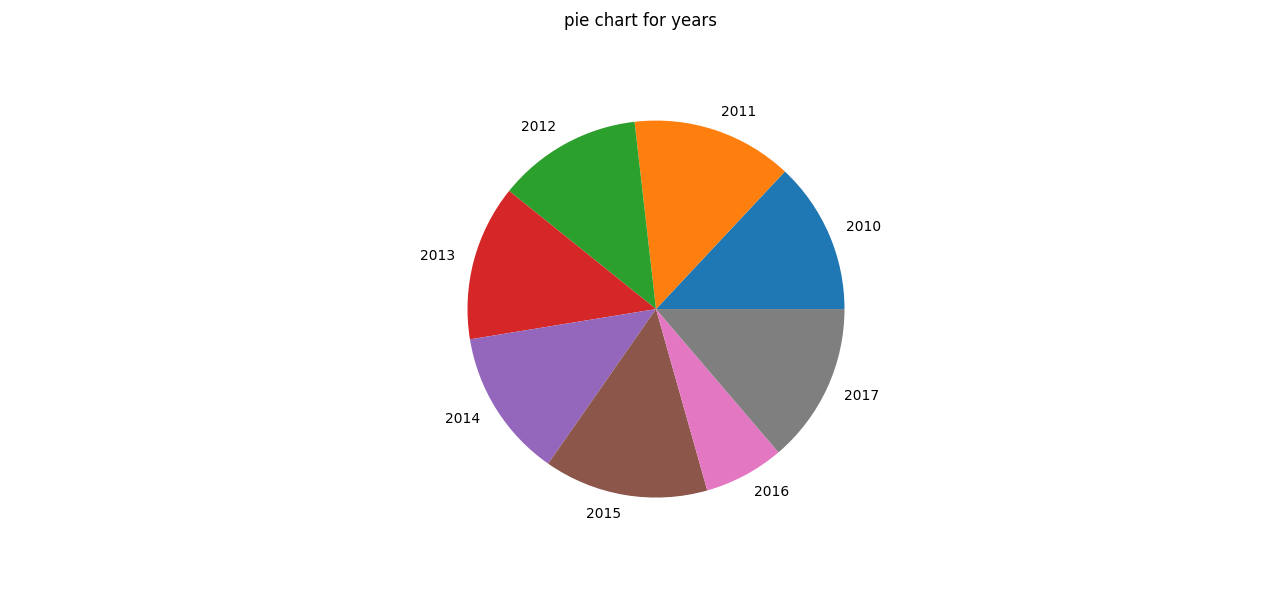
tes\_lables=[2010,2011,2012,2013,2014,2015,2016,2017]

plt.pie(b,labels=tes\_lables)

plt.show()

plt.legend(title="year statistics")

**Pie chart for sales over yeras:**

****