Step No :- 1

#imprt useful library

import pandas as pd

import os

import matplotlib.pyplot as plt

Step No :- 2

Merge 12 Month sales data in single file for this gather all data(All month sales csv file) in one particular folder.

# Now we Get all file in single array

files = [ file for file in os.listdir("E:/python/video/data analysis/Project-3 (Sales analysis Pandas)/SalesAnalysis/Sales\_Data")]

# Create empty data frame

all\_month\_data = pd.DataFrame()

Now we run for loop and make single csv file which contain all month data.

for file in files:

    df = pd.read\_csv("E:/python/video/data analysis/Project-3 (Sales analysis Pandas)/SalesAnalysis/Sales\_Data/" + file)

    all\_month\_data = pd.concat([all\_month\_data,df])

convert all\_month\_data data frame in csv file by following step

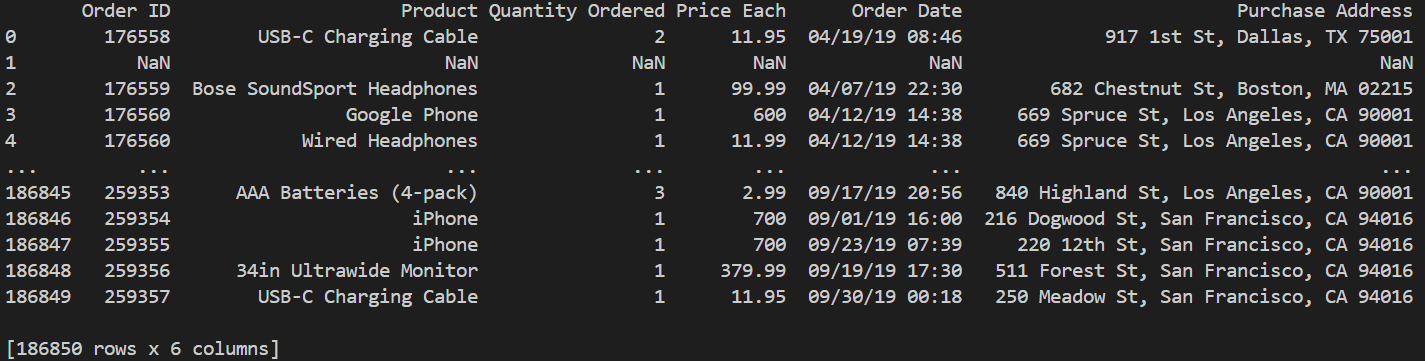
all\_month\_data.to\_csv("all\_data.csv",index=False)

Step No :- 3

Now we read csv file and print.

df = pd.read\_csv("E:/python/video/data analysis/Project-3 (Sales analysis Pandas)/all\_data.csv")

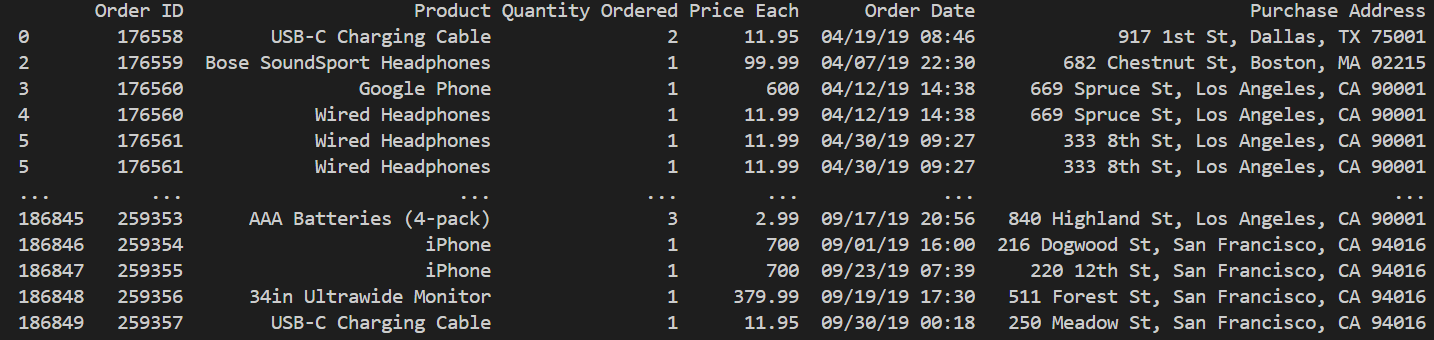
print(df)

print result

here we see that there is empty raw in index number 1 so we delete this type of all empty raw from our data frame for better analysis.

df = df.dropna(how='all')

print(df)

now we get our new data frame which have no any blank raw as follow.

Step No :- 4

we merge 12-month csv file in single. So we get heading in merge file multiple time so we remove this from our data frame And over write our data frame.

df = df[df["Order Date"].str[0:2] != "Or"]

Add month colum and covert there data type string to integer.

#add month colum

df["month"] = df["Order Date"].str[0:2]

## #convert month data type str to int

df["month"] = df["month"].astype('int32')

Question No:1

what was the best month for sales? how much we earned on that month?

First we convert quantity order colum and price colum in numeric than we multiply this both colum and make new colum of sales.

#we first multiply quantity & price so we get total sale for order

df["Quantity Ordered"] = pd.to\_numeric(df["Quantity Ordered"])

df["Price Each"] =  pd.to\_numeric(df["Price Each"])

df["sales"] = df["Quantity Ordered"] \* df["Price Each"]

now we group data month wise and sum of monthly sales.

monthly\_sale = df.groupby(["month"]).sum(["sales"])

now we plot graph of monthly sales.

x = range(1,13)

y = monthly\_sale["sales"]

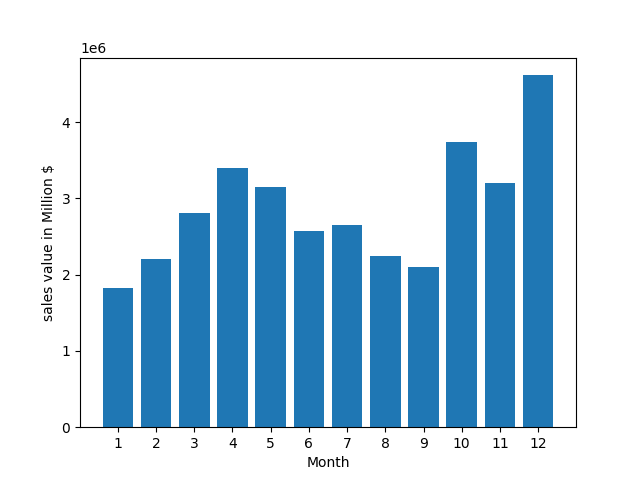
plt.bar(x,y)

plt.xticks(x)

plt.xlabel("Month")

plt.ylabel("sales value in Million $")

plt.show()



Question No:2

Which city get highest sales?

First we define function to extract city and state name from address.

def state(address):

    a =  address.split(",")[2]

    return a.split(" ")[1]

def city(address):

    return address.split(",")[1]

now we create new colun in our data frame which contain city name and there state.

df["city"] = df["Purchase Address"].apply(lambda x : f"{city(x)}[{state(x)}]")

now we group city wise and sum of city wise sales.

# #groupby city and sum sales citywise

sale\_city = df.groupby(["city"]).sum("sales")

now we plot bar chart for same.

x = df["city"].unique()

y = sale\_city["sales"]

plt.bar(x,y)

plt.xticks(x,rotation = "vertical" , size = 8)

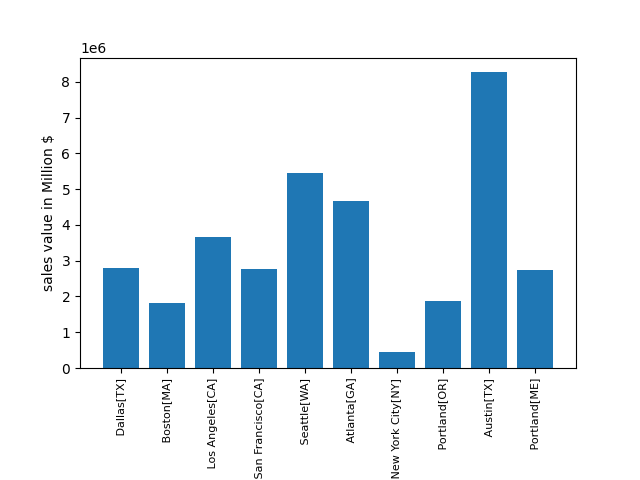
plt.xlabel("city")

plt.ylabel("sales value in Million $")

plt.show()

Question No:3

Which time is best for put advertisement?

First, we convert date into date and time object

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

df["hour"] = df["Order Date"].dt.hour

now we get hour and order count/ hour

hours = [hour for hour, d in df.groupby(["hour"])]

hours = [t[0] for t in hours]

count = df.groupby(["hour"]).count()

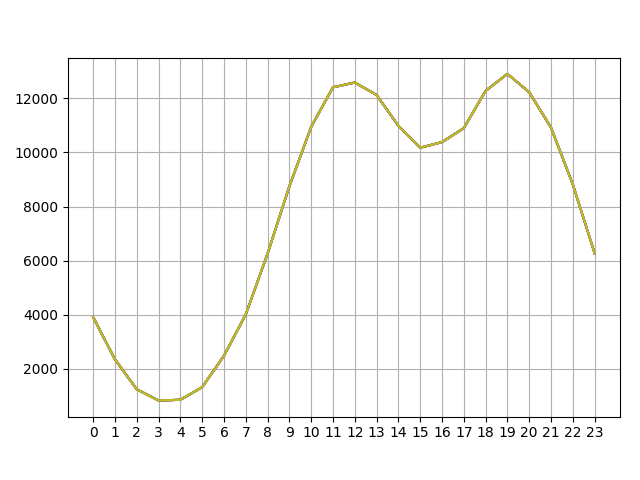
now we plot line chart for hourly order count

plt.plot(hours,count)

plt.xticks(hours)

plt.grid()

plt.show()



Question No:4

Which product are most sold together?

First, we get product which order together. We get by common order id.

combine\_order = df[df["Order ID"].duplicated(keep=False)]

combine\_order = combine\_order.copy()

we make group according product order together. Also delete duplicate raw.

combine\_order["product\_group"] = combine\_order.groupby(["Order ID"])["Product"].transform(lambda x: ",".join(x))

combine\_order = combine\_order.drop\_duplicates(["Order ID","product\_group"])

# Step 1: Sort the product combinations in each row

combine\_order["product\_group"] = combine\_order["product\_group"].apply(lambda x: ','.join(sorted(x.split(','))))

# Step 2: Count the occurrences of each combination

combination\_counts = combine\_order["product\_group"].value\_counts()

top\_combinations = combination\_counts.head(5)  # Top 5 combinations

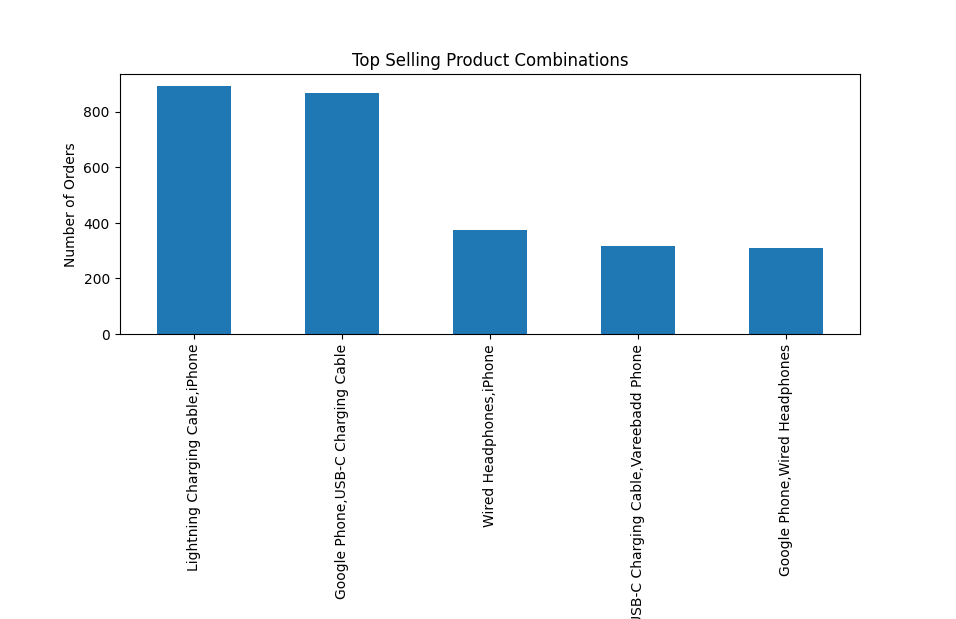
top\_combinations.plot(kind='bar')

plt.xlabel('Product Combinations')

plt.ylabel('Number of Orders')

plt.title('Top Selling Product Combinations')

plt.show()

 Question No:5

Which product sold the most? why do you think sold most?

For this we plot on x-axis product name and on y-axis sum of product & there avg price.

First, we make group by product

most\_sold = df.groupby("Product")

than we get total product sold according group

quntity = most\_sold.sum()["Quantity Ordered"]

now we get all product name and there avg price

products = [product for product,df in most\_sold ]

avg\_price =  most\_sold.sum()["Price Each"]/most\_sold.sum()["Quantity Ordered"]

now we plot bar chart of product & total qty sold. Also on same graph we plot line chart of product and there avg selling price.

fig,ax1 = plt.subplots()

ax2 = ax1.twinx()

ax1.bar(products,quntity)

ax2.plot(products,avg\_price,"red")

ax1.set\_xlabel("product name")

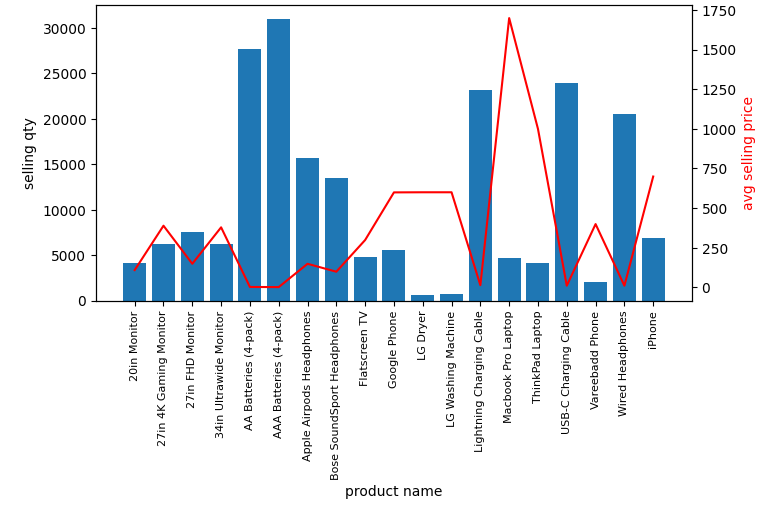
ax1.set\_ylabel("selling qty")

ax2.set\_ylabel("avg selling price",color="red")

ax1.set\_xticks(range(len(products)))

ax1.set\_xticklabels(products,rotation = "vertical" , size = 8)

plt.show()



on chart we observed that product which average selling price is law they sold most.