In [1]: **import** pandas **as** pd import matplotlib.pyplot as plt df= pd.read_csv(r"D:\Console Flare\Pandas-Data-Science-Tasks-master\SalesAnalysis\Output\all_data.csv") In [2]: df.head(5)In [3]: Order ID Product Quantity Ordered Price Each **Order Date Purchase Address** Out[3]: 11.95 04/19/19 08:46 176558 917 1st St, Dallas, TX 75001 **USB-C Charging Cable** 2 NaN NaN NaN NaN 99.99 04/07/19 22:30 682 Chestnut St, Boston, MA 02215 2 176559 Bose SoundSport Headphones 1 176560 600 04/12/19 14:38 669 Spruce St, Los Angeles, CA 90001 Google Phone 1 176560 Wired Headphones 1 11.99 04/12/19 14:38 669 Spruce St, Los Angeles, CA 90001 Clean up the data! Drop rows of NaN values nan_df= df[df.isna().any(axis=1)] In [4]: nan_df.head() df=df.dropna(how='all') df.head() **Purchase Address** Order ID Product Quantity Ordered Price Each **Order Date** Out[4]: **USB-C Charging Cable** 917 1st St, Dallas, TX 75001 176558 2 11.95 04/19/19 08:46 176559 Bose SoundSport Headphones 99.99 04/07/19 22:30 682 Chestnut St, Boston, MA 02215 1 600 04/12/19 14:38 669 Spruce St, Los Angeles, CA 90001 Google Phone 1 176560 Wired Headphones 1 11.99 04/12/19 14:38 669 Spruce St, Los Angeles, CA 90001 176561 Wired Headphones 11.99 04/30/19 09:27 333 8th St, Los Angeles, CA 90001 Find 'Or' and delete it df= df[df['Order Date'].str[0:2]!='Or'] Convert columns to the correct type df['Quantity Ordered'] = pd.to_numeric(df['Quantity Ordered'])# Make int df['Price Each']=pd.to_numeric(df['Price Each'])# Make float Augment data with additional columns Add Month column df['Month']= df['Order Date'].str[0:2] df['Month'] = df['Month'].astype('int32') df.head() Product Quantity Ordered Price Each Out[7]: Order ID **Order Date** Purchase Address Month 11.95 04/19/19 08:46 176558 **USB-C Charging Cable** 917 1st St, Dallas, TX 75001 2 99.99 04/07/19 22:30 1 176559 Bose SoundSport Headphones 682 Chestnut St, Boston, MA 02215 176560 Google Phone 1 600.00 04/12/19 14:38 669 Spruce St, Los Angeles, CA 90001 176560 Wired Headphones 11.99 04/12/19 14:38 669 Spruce St, Los Angeles, CA 90001 Wired Headphones 333 8th St, Los Angeles, CA 90001 176561 11.99 04/30/19 09:27 1 Add a city column # use a .apply() df['City']=df['Purchase Address'].apply(lambda x: x.split(',')[1]) df.head() Order ID Product Quantity Ordered Price Each **Order Date** Purchase Address Month Sales City Out[21]: **USB-C Charging Cable** 23.90 176558 11.95 04/19/19 08:46 917 1st St, Dallas, TX 75001 Dallas 176559 Bose SoundSport Headphones 99.99 04/07/19 22:30 682 Chestnut St, Boston, MA 02215 4 99.99 Boston 176560 Google Phone 1 600.00 04/12/19 14:38 669 Spruce St, Los Angeles, CA 90001 4 600.00 Los Angeles 176560 Wired Headphones 11.99 04/12/19 14:38 669 Spruce St, Los Angeles, CA 90001 4 11.99 Los Angeles 176561 Wired Headphones 1 11.99 04/30/19 09:27 333 8th St, Los Angeles, CA 90001 4 11.99 Los Angeles Add a Sales column df['Sales']=df['Quantity Ordered'] * df['Price Each'] df.head() Order ID Product Quantity Ordered Price Each **Order Date** Purchase Address Month Sales Out[8]: **USB-C Charging Cable** 176558 2 11.95 04/19/19 08:46 917 1st St, Dallas, TX 75001 23.90 176559 Bose SoundSport Headphones 1 99.99 04/07/19 22:30 682 Chestnut St, Boston, MA 02215 4 99.99 176560 Google Phone 1 600.00 04/12/19 14:38 669 Spruce St, Los Angeles, CA 90001 4 600.00 176560 11.99 04/12/19 14:38 669 Spruce St, Los Angeles, CA 90001 4 11.99 Wired Headphones 176561 Wired Headphones 1 11.99 04/30/19 09:27 333 8th St, Los Angeles, CA 90001 4 11.99 Best month for Sales results =df.groupby('Month').sum() months=range(1,13) In [14]: plt.bar(months, results['Sales']) plt.xticks(months) plt.ylabel('Sales in USD (\$)') plt.xlabel('Month Number') plt.show() 1e6 4 (\$) Sales in 2 1 8 Month Number City had the highest number of sales? result=df.groupby('City').sum() In [31]: result Quantity Ordered Price Each Month Out[31]: Sales City Atlanta 16602 2779908.20 104794 2795498.58 **Austin** 11153 1809873.61 69829 1819581.75 22528 3637409.77 141112 3661642.01 **Boston Dallas** 16730 2752627.82 104620 2767975.40 Los Angeles 33289 5421435.23 208325 5452570.80 **New York City** 27932 4635370.83 175741 4664317.43 **Portland** 14053 2307747.47 87765 2320490.61 50239 8211461.74 315520 8262203.91 San Francisco Seattle 16553 2733296.01 104941 2747755.48 cities=[city for city,df in df.groupby('City')] plt.bar(cities, result['Sales']) plt.xticks(cities, rotation='vertical', size=8) plt.ylabel('Sales in USD (\$)') plt.xlabel('City name') Text(0.5, 0, 'City name') Out[37]: 1e6 8 7 6 Sales in USD (\$) 3 2 1 . San Francisco Los Angeles City name what time should we display advertisements to maximize likelihood of customer's buying product? df['Order Date']=pd.to_datetime(df['Order Date']) In [39]: df.head() Order ID Product Quantity Ordered Price Each **Order Date** Purchase Address Month Sales City Out[39]: **USB-C Charging Cable** 11.95 2019-04-19 08:46:00 Dallas 176558 2 917 1st St, Dallas, TX 75001 23.90 2019-04-07 22:30:00 682 Chestnut St, Boston, MA 02215 176559 Bose SoundSport Headphones 99.99 99.99 Boston Google Phone 1 176560 600.00 2019-04-12 14:38:00 669 Spruce St, Los Angeles, CA 90001 4 600.00 Los Angeles 2019-04-12 14:38:00 176560 Wired Headphones 1 669 Spruce St, Los Angeles, CA 90001 4 11.99 Los Angeles Wired Headphones 176561 1 11.99 2019-04-30 09:27:00 333 8th St, Los Angeles, CA 90001 11.99 Los Angeles In [43]: df['Hour']=df['Order Date'].dt.hour df['Minute']=df['Order Date'].dt.minute df.head() City Hour Minute Out[43]: Order ID Product Quantity Ordered Price Each Order Date Purchase Address Month Sales 176558 **USB-C Charging Cable** 2 11.95 2019-04-19 08:46:00 917 1st St, Dallas, TX 75001 23.90 Dallas 46 4 8 176559 Bose SoundSport Headphones 99.99 2019-04-07 22:30:00 682 Chestnut St, Boston, MA 02215 99.99 Boston 22 30 176560 Google Phone 600.00 2019-04-12 14:38:00 669 Spruce St, Los Angeles, CA 90001 4 600.00 Los Angeles 1 38 2019-04-12 14:38:00 11.99 Los Angeles 176560 Wired Headphones 669 Spruce St, Los Angeles, CA 90001 38 176561 Wired Headphones 1 11.99 2019-04-30 09:27:00 9 27 333 8th St, Los Angeles, CA 90001 4 11.99 Los Angeles hours=[hour for hour, df in df.groupby('Hour')] plt.plot(hours, df.groupby(['Hour']).count()) plt.xticks(hours) plt.xlabel('Hour') plt.ylabel('Number of orders') plt.grid() plt.show() 12000 10000 Number of orders 8000 6000 4000 2000 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 Hour Most sold product product_group=df.groupby('Product') In [71]: quantity_order =product_group.sum()['Quantity Ordered'] products =[product for product, df in product_group] plt.bar(products, quantity_order) plt.ylabel('Quantity Ordered') plt.xlabel('products') plt.xticks(products, rotation='vertical', size=8) plt.show() 30000 25000 Quantity Ordered 20000 15000 10000 5000 Google Phone LG Dryer AA Batteries (4-pack) Apple Airpods Headphones Flatscreen TV Lightning Charging Cable Macbook Pro Laptop ThinkPad Laptop USB-C Charging Cable Vareebadd Phone iPhone 27in FHD Monitor 34in Ultrawide Monitor AAA Batteries (4-pack) Bose SoundSport Headphones LG Washing Machine products In [83]: prices = df.groupby('Product').mean()['Price Each'] fig, ax1 =plt.subplots() ax2 = ax1.twinx()ax1.bar(products, quantity_order, color='g') ax2.plot(products, prices, 'b-') ax1.set_xlabel('Product Name') ax1.set_ylabel('Quantity Ordered',color='g') ax2.set_ylabel('Price(\$)',color='b') ax1.set_xticklabels(products, rotation='vertical', size=8) plt.show() C:\Users\HP\AppData\Local\Temp\ipykernel_4336\293809788.py:10: UserWarning: FixedFormatter should only be used together with FixedLocator ax1.set_xticklabels(products, rotation='vertical', size=8) 1750 30000 1500 25000 1250 Quantity Ordered 20000 1000 15000 750 500 10000 250 5000 AA Batteries (4-pack) AAA Batteries (4-pack) Apple Airpods Headphones Flatscreen TV Google Phone LG Washing Machine Lightning Charging Cable Macbook Pro Laptop ThinkPad Laptop USB-C Charging Cable Vareebadd Phone iPhone 27in FHD Monitor 34in Ultrawide Monitor Bose SoundSport Headphones LG Dryer Wired Headphones 27in 4K Gaming Monitor **Product Name**

Sales Analysis

Import Libraries