LONDON METROPOLITAN UNIVERSITY

ITAHARI INTERNATIONAL COLLEGE

Module Code & Module Title

CC5067NT Smart Data Discovery

Assessment Weightage & Type

60% Individual Coursework

Year and Semester

2022-23 Spring

Student Name: MD Arsalan

Group: L2C5

London Met ID: 22016061

College ID: np05cp4s220032

Assignment Due Date: 4th May 2023

Assignment Submission Date: 4th May2023

Table of Contents

Introduction 1
  Python 1
  Pandas 2
  Matplotlib 3
Data Understanding 3
  Introduction 3
Data Preparation 7
  Introduction 7
Data Analysis 11
  Introduction 11
Data Exploration 13
  Introduction 13
Conclusion 19
Bibliography 20

Introduction

Python

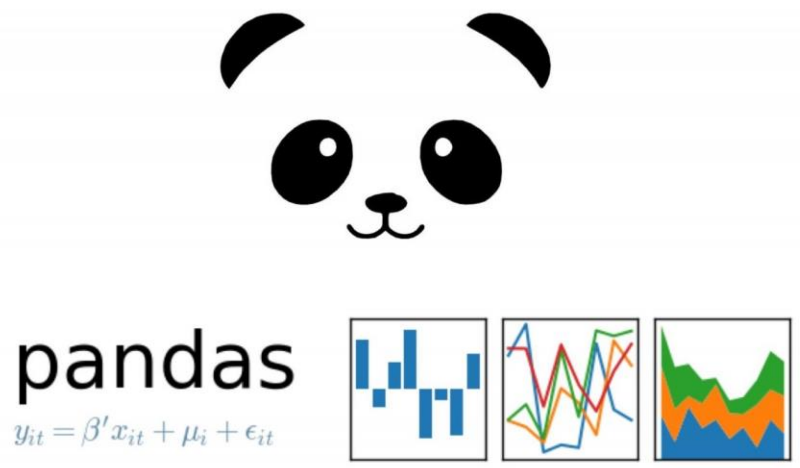
Python is a popular programming language that turned into created in 1991 via Guido van Rossum. It is understood for being a "high-level" language, which means it is designed to be without difficulty study and written by human beings. Python is also an "interpreted" language, that means that every line of code is carried out separately via a pc program. One of the greatest strengths of Python is its versatility. It can be used for a wide variety of duties, which include building websites, analyzing information, and creating system learning fashions. This is due in element to Python's simple syntax, which makes it a really perfect choice for beginners.

Python boasts numerous crucial functions, consisting of its ease of gaining knowledge of. The language turned into designed with readability in thoughts, so even novice programmers can speedy hold close the fundamentals. Additionally, Python's interpreted nature permits builders to check and refine their code unexpectedly.

Another key advantage of Python is its dynamic typing. This means that you don't want to specify statistics types to your variables, making coding greater flexible and much less time-consuming. Python additionally helps object-orientated programming, taking into consideration the creation of complicated code structures. Finally, Python consists of a huge standard library that gives pre-written functions and gear to simplify programming tasks. (python, 2023)

Python is also renowned for its lively and numerous developer network. This community contributes to the continuing improvement of Python and creates severa libraries and frameworks, inclusive of NumPy, Pandas, Django, Flask, and TensorFlow, that expand Python's competencies and make it a popular preference for many programming tasks.

Pandas



Pandas is a famous open-source statistics manipulation and analysis library for the Python programming language. It gives a wide variety of equipment and features for running with established statistics, making it a effective tool for facts scientists, analysts, and engineers.

Pandas can deal with plenty of information kinds together with tabular, time-collection, and matrix statistics. It presents gear for cleaning, filtering, and reworking statistics, in addition to strategies for running with lacking or incomplete data. Pandas also offers powerful indexing and grouping capabilities, making it easy to explore and examine large datasets. (Answers, 2023)

Some of the important thing functions of Pandas encompass its ability to load data from loads of resources together with CSV, Excel, SQL databases, and HTML. It also has sturdy abilties for merging and becoming a member of datasets, and might cope with complicated information manipulation duties easily. Additionally, Pandas can produce tremendous visualizations of information with its built-in plotting capabilities.

Overall, Pandas is a effective library that makes it smooth to work with established statistics in Python, and is extensively used within the fields of information technological know-how, finance, economics, and greater.

Matplotlib

Matplotlib is a popular open-source information visualization library for the Python programming language. It presents a huge variety of gear for growing static, lively, and interactive visualizations in Python. Matplotlib turned into designed to duplicate the functionality of MATLAB's plotting abilities, however in a Pythonic way.

Matplotlib allows users to create loads of plots, which include line plots, scatter plots, bar plots, histograms, and more. It gives significant customization options for colour, line fashion, and markers, in addition to the potential to add annotations, legends, and titles to plots. Matplotlib additionally supports 3D plotting and geographical mapping. In addition to its big talents for growing static visualizations, Matplotlib additionally offers gear for creating interactive visualizations. These interactive plots allow users to explore records and modify visualizations in real-time, making it a valuable device for data analysis and exploration.

Matplotlib is extensively used in medical research, finance, facts evaluation, and many different fields that require visualizing complicated statistics. It is also frequently used along side different Python libraries, such as NumPy and Pandas, to create powerful information analysis and visualization workflows.

Data Understanding

Introduction

Data expertise is a important step inside the facts mining manner that involves getting to know and gaining insights into the statistics that you'll be analyzing. It is the procedure of inspecting and exploring the records to gain a deeper expertise of its structure, content, and relationships.

During the statistics expertise section, you'll generally perform tasks which includes:

Data collection: Collecting the records with a purpose to be used for analysis, whether or not it's from internal or outside sources.

- Data description: Describing the statistics in phrases of its length, layout, and structure. This can contain reviewing facts dictionaries, metadata, and other documentation.
- Data exploration: Conducting an initial exploration of the facts to perceive patterns, tendencies, and outliers. This can contain producing precis information, visualizations, and other exploratory statistics analysis techniques.
- Data first-rate assessment: Assessing the first-rate of the facts and figuring out any troubles that may effect the analysis. This can contain figuring out missing or incomplete facts, records access errors, or other problems.
- Data preparation: Preparing the information for evaluation through cleansing, transforming, and integrating the records. This can contain responsibilities along with information cleaning, function choice, and characteristic engineering.

Data information is a important step within the information mining procedure that helps to make sure that the facts used for evaluation is accurate, applicable, and significant. By gaining a deeper information of the records, you can make extra informed decisions and generate more accurate insights.

E-trade has ended up a ubiquitous trouble of modern-day buying. It gives quite some of blessings in conjunction with consolation, easy accessibility, and various product availability. As a end result, the e-trade marketplace has seen fantastic increase in modern-day years. With the growth in on-line buying, it has emerged as vital to research e-commerce transaction information to higher recognize purchaser behavior and make informed alternatives to improve enterprise outcomes.

This file specializes in e-trade transaction facts files containing facts about diverse orders, including charging cables, headphones and batteries The information covers a period of three hundred and sixty five days, from January 2019 via December 2019, and has six columns: order ID, products. The amount ordered, each charge, date of order, and deal with of buy information documents contain a few lacking values, duplicates, and scheduling troubles that have to be solved so that you can cast off analysis it makes feel.

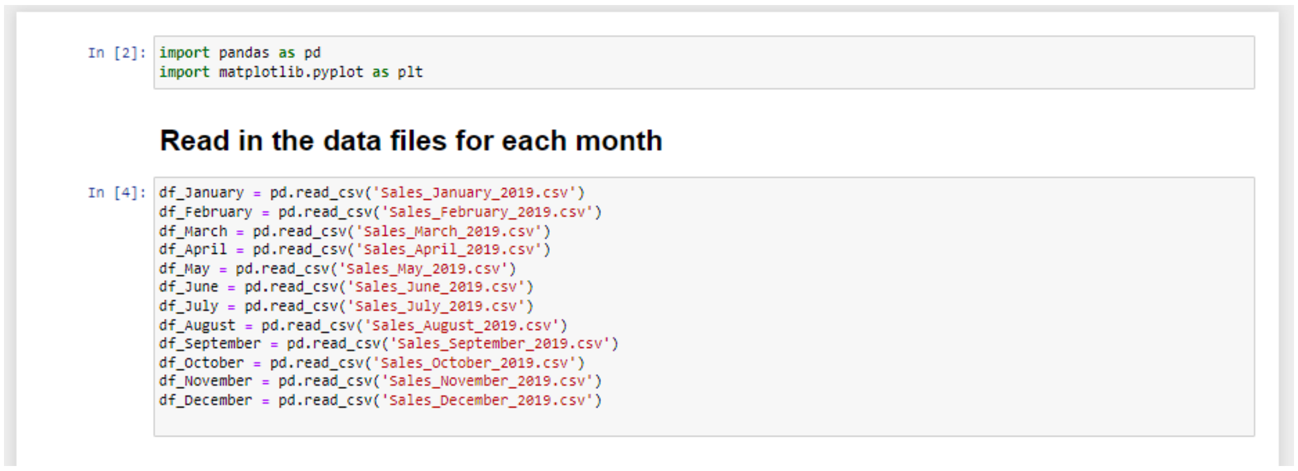
One of the most vital portions of facts in records documents is the Order ID, that is a unique identifier for each order. It lets in us to track individual orders, examine trends and pick out consumer conduct. The product category offers information of the diverse products offered on the e-commerce platform, and the Quantity Ordered column gives an concept of the amount of products offered Price Each column offers records about the rate of each product, that is essential for accounting income and profit.

The Order Date column is an vital part of records files because it offers us insight into while customers will purchase gadgets. It allows us to investigate developments, which includes seasonal making plans or promotional marketing campaign outcomes, and make suitable pricing and inventory control choices The Purchase Address column affords records about, and permits visibility into, patron region local enhancements in patron behavior and we optimize delivery and delivery strategies.

However, there are lacking values, duplicates, and formatting issues in the facts files that want to be addressed prior to evaluation. Missing values may be problematic because they can skew the analysis, and duplicates can produce faulty outcomes. Formatting problems along with dates inside the incorrect layout or more than one characters in the acquisition deal with can make searches difficult to appropriately carry out.

Therefore, the information need to be cleaned and pre-processed to ensure the accuracy and validity of the records acquired.

In conclusion, the eCommerce transaction statistics documents analyzed in this document provide precious insights into patron behavior and market dynamics. But to restore lacking requirements, issues of replica layout a, essential to make certain insights from information are correct and dependable and improve profitability Able to make knowledgeable choices concerning pricing, inventory and advertising strategies.

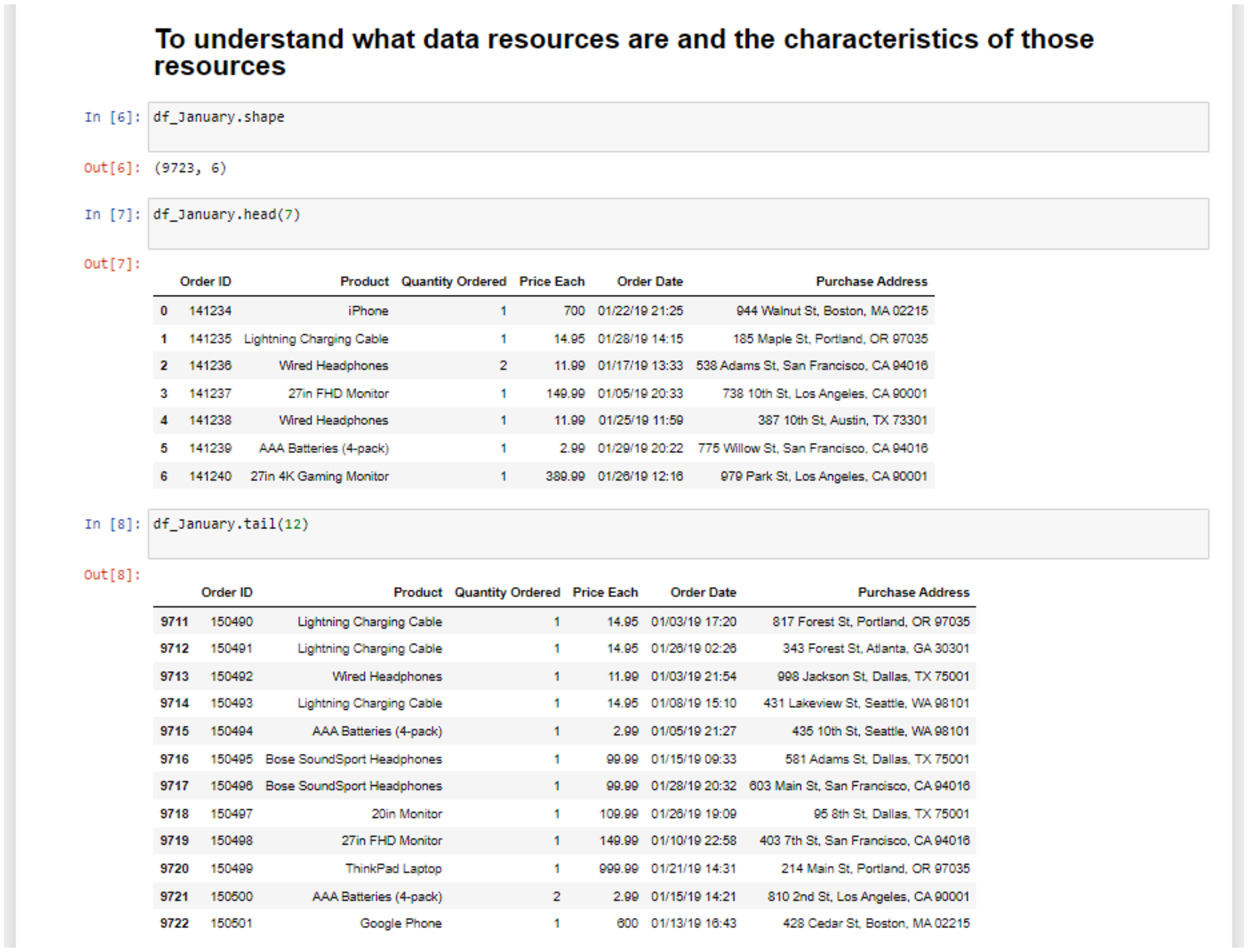


```
In [2]: import pandas as pd
        import matplotlib.pyplot as plt
```

Read in the data files for each month

```
In [4]: df_January = pd.read_csv('Sales_January_2019.csv')
        df_February = pd.read_csv('Sales_February_2019.csv')
        df_March = pd.read_csv('Sales_March_2019.csv')
        df_April = pd.read_csv('Sales_April_2019.csv')
        df_May = pd.read_csv('Sales_May_2019.csv')
        df_June = pd.read_csv('Sales_June_2019.csv')
        df_July = pd.read_csv('Sales_July_2019.csv')
        df_August = pd.read_csv('Sales_August_2019.csv')
        df_September = pd.read_csv('Sales_September_2019.csv')
        df_October = pd.read_csv('Sales_October_2019.csv')
        df_November = pd.read_csv('Sales_November_2019.csv')
        df_December = pd.read_csv('Sales_December_2019.csv')
```

This code imports two Python libraries - pandas and ‘matplotlib.pyplot’ - using the "pd" and ‘plt’ aliases, respectively. It then reads in 12 CSV files (one for each month of 2019) using pandas ‘read_csv’’ function and assigns each to a separate data frame, with each data frame being named after the corresponding month.



To understand what data resources are and the characteristics of those resources

```
In [6]: df_January.shape
```

```
Out[6]: (9723, 6)
```

```
In [7]: df_January.head(7)
```

Out[7]:

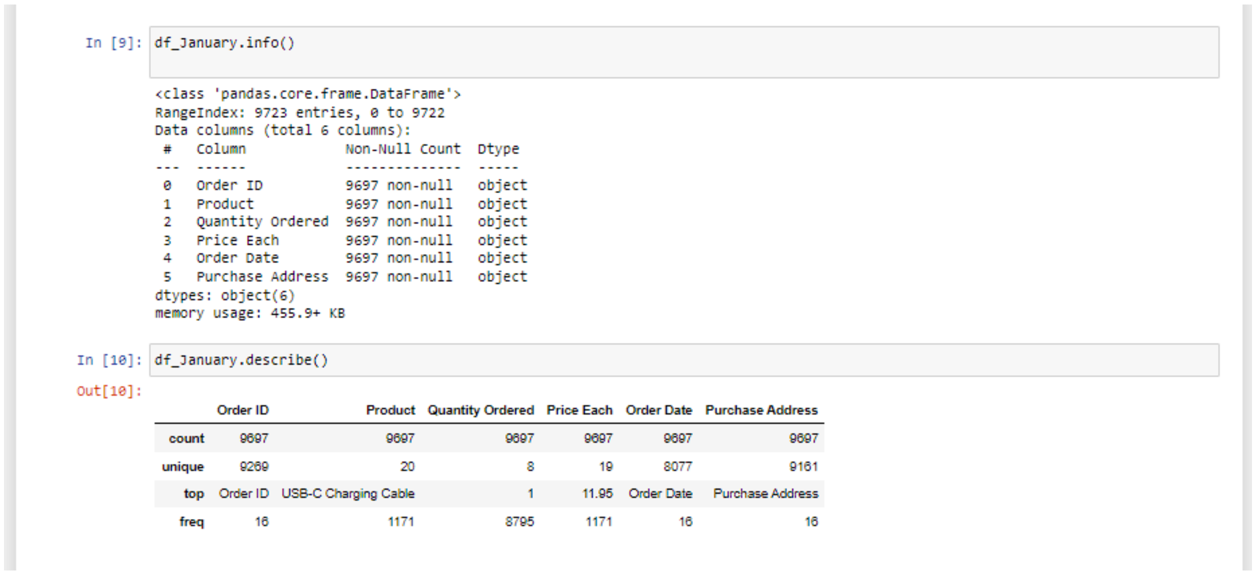
	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	141234	iPhone	1	700	01/22/19 21:25	944 Walnut St, Boston, MA 02215
1	141235	Lightning Charging Cable	1	14.95	01/28/19 14:15	185 Maple St, Portland, OR 97035
2	141236	Wired Headphones	2	11.99	01/17/19 13:33	538 Adams St, San Francisco, CA 94016
3	141237	27in FHD Monitor	1	149.99	01/05/19 20:33	738 10th St, Los Angeles, CA 90001
4	141238	Wired Headphones	1	11.99	01/25/19 11:59	387 10th St, Austin, TX 73301
5	141239	AAA Batteries (4-pack)	1	2.99	01/29/19 20:22	775 Willow St, San Francisco, CA 94016
6	141240	27in 4K Gaming Monitor	1	389.99	01/26/19 12:16	979 Park St, Los Angeles, CA 90001

```
In [8]: df_January.tail(12)
```

Out[8]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
9711	150490	Lightning Charging Cable	1	14.95	01/03/19 17:20	817 Forest St, Portland, OR 97035
9712	150491	Lightning Charging Cable	1	14.95	01/26/19 02:26	343 Forest St, Atlanta, GA 30301
9713	150492	Wired Headphones	1	11.99	01/03/19 21:54	998 Jackson St, Dallas, TX 75001
9714	150493	Lightning Charging Cable	1	14.95	01/08/19 15:10	431 Lakeview St, Seattle, WA 98101
9715	150494	AAA Batteries (4-pack)	1	2.99	01/05/19 21:27	435 10th St, Seattle, WA 98101
9716	150495	Bose SoundSport Headphones	1	99.99	01/15/19 09:33	581 Adams St, Dallas, TX 75001
9717	150496	Bose SoundSport Headphones	1	99.99	01/28/19 20:32	603 Main St, San Francisco, CA 94016
9718	150497	20in Monitor	1	109.99	01/26/19 19:09	95 8th St, Dallas, TX 75001
9719	150498	27in FHD Monitor	1	149.99	01/10/19 22:58	403 7th St, San Francisco, CA 94016
9720	150499	ThinkPad Laptop	1	999.99	01/21/19 14:31	214 Main St, Portland, OR 97035
9721	150500	AAA Batteries (4-pack)	2	2.99	01/15/19 14:21	810 2nd St, Los Angeles, CA 90001
9722	150501	Google Phone	1	600	01/13/19 16:43	428 Cedar St, Boston, MA 02215

- df_January.Form returns the scale of the statistics frame, which in this example would be the variety of rows and columns.
- df_January.Head(7) returns the first seven rows of the information frame, allowing you to speedy check out the top of the data and get a experience of its shape.
- df_January.Tail(12) returns the last 12 rows of the records frame, permitting you to check out the cease of the records and check for any patterns or outliers.



```
In [9]: df_January.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9723 entries, 0 to 9722
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Order ID          9697 non-null   object
 1   Product           9697 non-null   object
 2   Quantity Ordered  9697 non-null   object
 3   Price Each        9697 non-null   object
 4   Order Date        9697 non-null   object
 5   Purchase Address  9697 non-null   object
dtypes: object(6)
memory usage: 455.9+ KB
```

```
In [10]: df_January.describe()
```

Out[10]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
count	9697	9697	9697	9697	9697	9697
unique	9269	20	8	19	8077	9161
top	Order ID	USB-C Charging Cable	1	11.95	Order Date	Purchase Address
freq	16	1171	8795	1171	16	16

- df_January.Data() gives a precis of the facts frame, together with the wide variety of non-null values in every column the records form of every column and the overall reminiscence usage of the statistics frame.
- df_January.Describe() offers descriptive facts for each numerical column inside the data frame, inclusive of the matter, imply, general deviation, minimum and maximum values as well as the quartiles. This records can help you get a better knowledge of the distribution and variety of values within the statistics.

Data Preparation

Introduction

Data training is the technique of cleaning, remodeling, and structuring uncooked facts right into a layout this is suitable for evaluation. It is an vital step inside the records mining method and generally involves a sequence of obligations that help to ensure the accuracy, completeness, and consistency of the information.

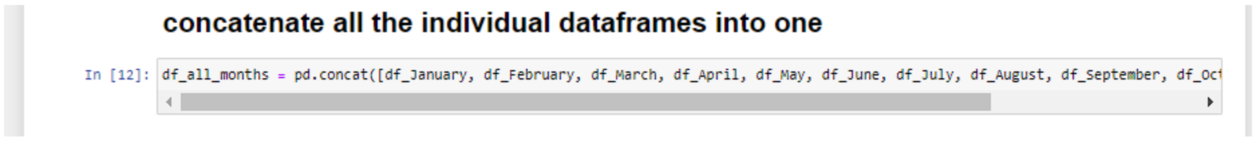
The facts education method can consist of the following steps:

- Data cleaning: Removing or solving any mistakes, inconsistencies, or lacking records within the dataset. This can involve strategies along with imputation, filtering, or disposing of beside the point information.

- Data integration: Combining information from specific resources into a unmarried dataset. This can involve matching facts based totally on common fields or the usage of greater advanced strategies together with records linkage.
- Data transformation: Changing the layout or shape of the information to make it less complicated to research. This can involve duties which include scaling, normalization, or characteristic extraction.
- Data discount: Reducing the size of the dataset via casting off redundant or inappropriate facts. This can involve strategies consisting of characteristic choice or dimensionality discount.
- Data formatting: Ensuring that the statistics is in a regular format this is suitable for evaluation. This can involve duties such as converting facts sorts or reformatting date and time fields.

Overall, statistics instruction is a important step in the data mining procedure that enables to make certain that the statistics used for analysis is correct, whole, and relevant. By well preparing the facts, you could lessen the hazard of errors and bias, enhance the quality of insights, and in the long run make better selections based on the data.

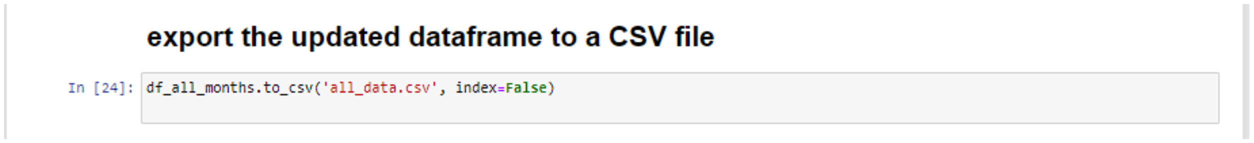
- Write a python program to merge data from each month into one CSV and read in updated dataframe.



concatenate all the individual dataframes into one

```
In [12]: df_all_months = pd.concat([df_January, df_February, df_March, df_April, df_May, df_June, df_July, df_August, df_September, df_Oct
```

concatenates or combines all the data frames for each month into one large data frame named df_all_months. This allows for easier analysis and manipulation of the data, as all the monthly data is now in one place.

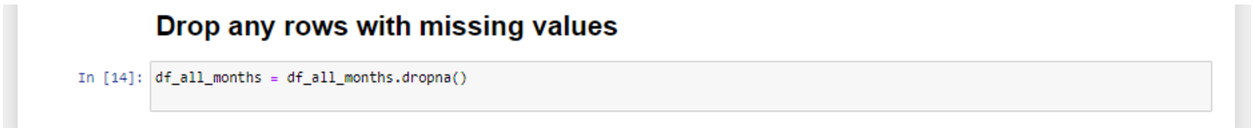


export the updated dataframe to a CSV file

```
In [24]: df_all_months.to_csv('all_data.csv', index=False)
```

df_all_months DataFrame is being saved as a CSV file with the name all_data.csv. The index=False argument tells pandas not to include the index column in the CSV file. If index=True, then the index column will be included in the CSV file as the first column.

- Write a python program to remove the NaN missing values from updated dataframe.

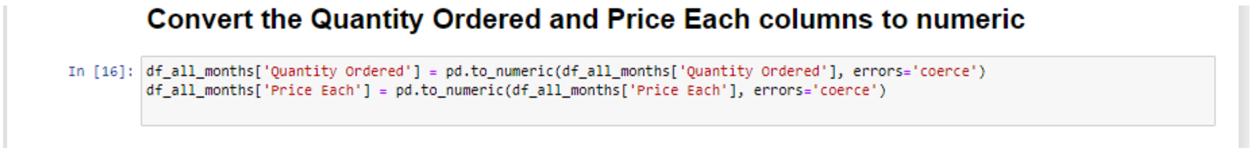


Drop any rows with missing values

```
In [14]: df_all_months = df_all_months.dropna()
```

This code is dropping any rows that have missing values NaN inside the df_all_months DataFrame. Dropna() is a pandas function that gets rid of any rows with missing values. By doing this, we are able to ensure that our information is clean and that we won't have any troubles with lacking data when we carry out our evaluation.

- Write a python program to convert Quantity Ordered and Price Each to numeric.



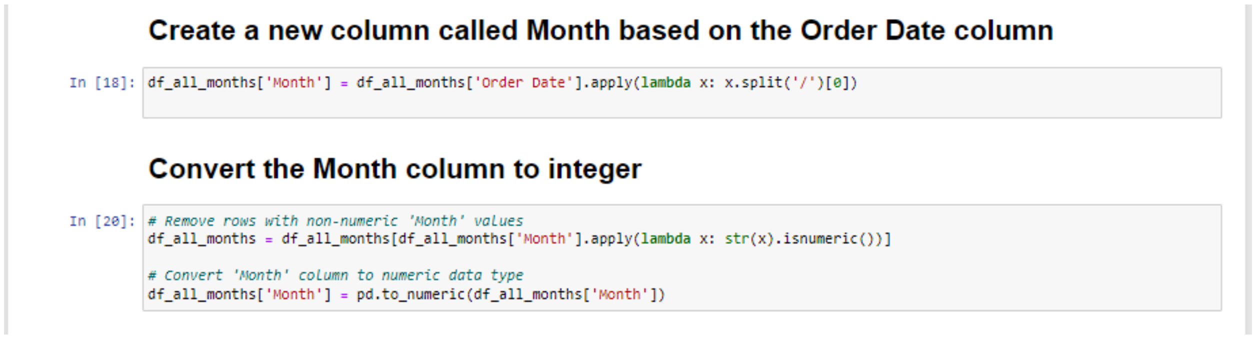
Convert the Quantity Ordered and Price Each columns to numeric

```
In [16]: df_all_months['Quantity Ordered'] = pd.to_numeric(df_all_months['Quantity Ordered'], errors='coerce')
         df_all_months['Price Each'] = pd.to_numeric(df_all_months['Price Each'], errors='coerce')
```

The first line converts the 'Quantity Ordered' column to numeric type, and any values that cannot be converted are replaced with NaN (not a number) using the 'coerce' option. Similarly, the second line converts the 'Price Each' column to numeric type and replaces any non-numeric values with NaN.

By converting the data type of these columns to numeric type, we can perform mathematical operations on them and analyze the data in a more meaningful way.

- Create a new column named Month from Ordered Date of updated dataframe and convert it to integer as data type



Create a new column called Month based on the Order Date column

```
In [18]: df_all_months['Month'] = df_all_months['Order Date'].apply(lambda x: x.split('/')[0])
```

Convert the Month column to integer

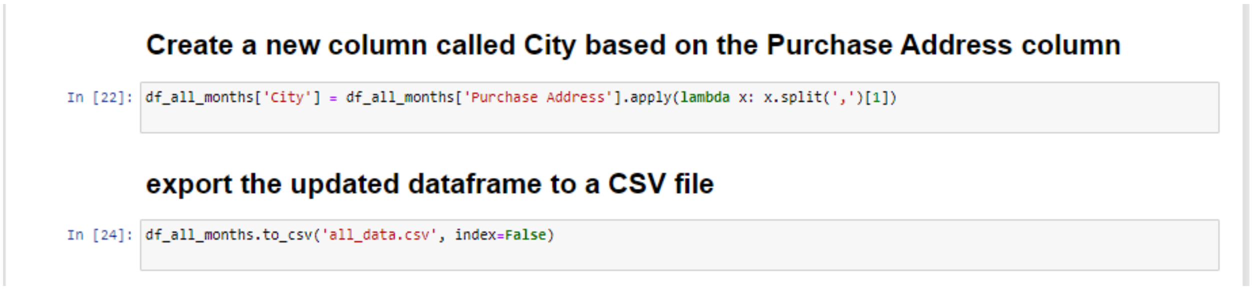
```
In [20]: # Remove rows with non-numeric 'Month' values
         df_all_months = df_all_months[df_all_months['Month'].apply(lambda x: str(x).isnumeric())]

         # Convert 'Month' column to numeric data type
         df_all_months['Month'] = pd.to_numeric(df_all_months['Month'])
```

A new column 'Month' is created in the 'df_all_months' DataFrame by means of extracting the month price from the 'Order Date' column. The follow() function is used to use a lambda function to each fee of the 'Order Date' column. The lambda characteristic splits the 'Order Date' value the use of '/' because the separator and returns the primary element of the ensuing list, that's the month value.

Next the rows with non-numeric values inside the 'Month' column are eliminated from the DataFrame the use of the apply() function with any other lambda feature. The str.Isnumeric() approach is used to test if every cost of the 'Month' column is numeric or now not. The resulting boolean values are used to clear out the DataFrame to maintain most effective the rows with numeric month values.

Finally, the 'Month' column is converted to a numeric data kind the use of the pd.To_numeric() function. This is finished to ensure that the values in the column are treated as numeric statistics, so that mathematical operations may be completed on them later.

- Create a new column named City from Purchase Address based on the value in updated dataframe.



Create a new column called City based on the Purchase Address column

```
In [22]: df_all_months['City'] = df_all_months['Purchase Address'].apply(lambda x: x.split(',')[1])
```

export the updated dataframe to a CSV file

```
In [24]: df_all_months.to_csv('all_data.csv', index=False)
```

extracts the city name from the Purchase Address column in the df_all_months DataFrame and creates a new City column to store the extracted values.

It uses the apply() method to apply a lambda function to each row of the Purchase Address column. The lambda function splits the address string by comma (,) and selects the second element of the resulting list, which is the city name.

Data Analysis

Introduction

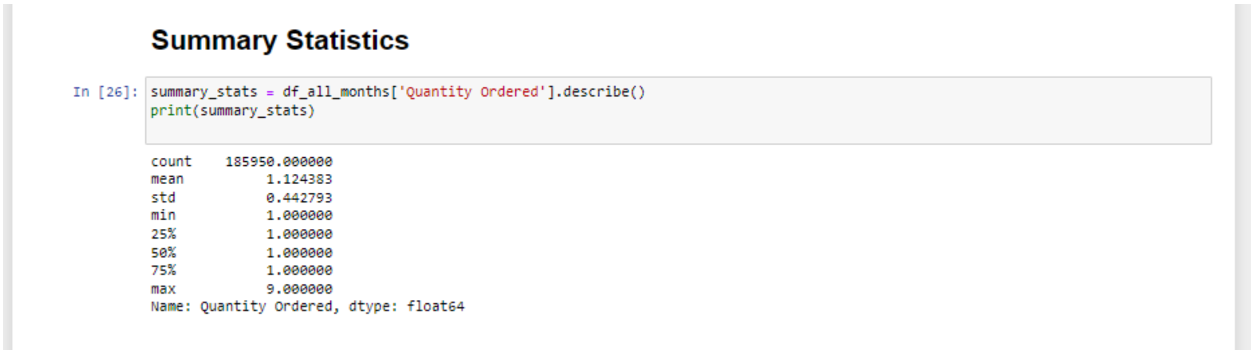
Data analysis is the procedure of inspecting, reworking, and modeling data with the purpose of coming across beneficial statistics, drawing conclusions, and helping decision-making. It involves using various strategies and gear to organize, discover, and examine data to be able to extract insights and expertise from it.

Data analysis may be divided into numerous levels, together with:

- Data cleaning and education: This involves casting off errors and inconsistencies from the facts, reworking it right into a format this is suitable for analysis, and choosing the applicable records for the analysis.
- Exploratory facts evaluation: This entails visualizing and summarizing the statistics to perceive styles, relationships, and tendencies. This can encompass techniques consisting of descriptive statistics, records visualization, and clustering.
- Statistical inference: This entails using statistical strategies to draw conclusions and make predictions based totally on the statistics. This can consist of techniques which include speculation testing and regression evaluation.
- Machine gaining knowledge of: This includes the usage of algorithms to research patterns inside the facts and make predictions or classifications. This can include strategies which include selection trees, neural networks, and help vector machines.
- Data interpretation and communique: This includes providing the results of the analysis in a clear and meaningful way, and the use of them to assist choice-making.

Data evaluation may be utilized in a variety of fields, along with business, finance, healthcare, and social sciences. It is a effective tool for understanding complicated facts and making knowledgeable decisions based totally at the insights it provides.

- Write a Python program to show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of any chosen variable.



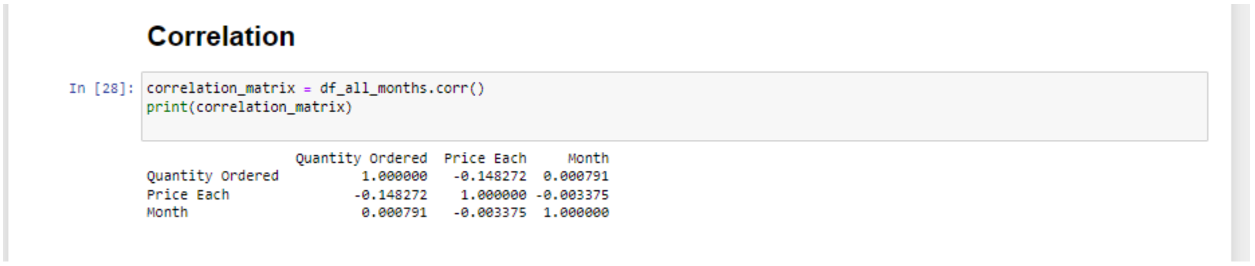
Summary Statistics

```
In [26]: summary_stats = df_all_months['Quantity Ordered'].describe()
         print(summary_stats)
```

```
count    185950.000000
mean          1.124383
std           0.442793
min           1.000000
25%           1.000000
50%           1.000000
75%           1.000000
max           9.000000
Name: Quantity Ordered, dtype: float64
```

computes the descriptive statistics of the 'Quantity Ordered' column in the dataframe 'df_all_months', and then prints the summary statistics. The summary statistics include the count (number of non-null values), mean, standard deviation, minimum value, 25%, median 50% , 75% and maximum value of the 'Quantity Ordered' column.

- Write a Python program to calculate and show correlation of all variables.



Correlation

```
In [28]: correlation_matrix = df_all_months.corr()
         print(correlation_matrix)
```

```
                  Quantity Ordered  Price Each     Month
Quantity Ordered          1.000000   -0.148272  0.000791
Price Each               -0.148272    1.000000 -0.003375
Month                     0.000791   -0.003375  1.000000
```

Calculates the correlation matrix of the df_all_months DataFrame, which contains sales statistics for some of months.

The corr() characteristic computes the pairwise correlation of columns inside the DataFrame. It returns a new DataFrame that suggests the correlation coefficients between all pairs of columns in the authentic DataFrame.

The ensuing correlation_matrix DataFrame suggests the correlation coefficients between all pairs of columns in df_all_months, in which values close to 1 imply a strong wonderful correlation and values near -1 imply a sturdy poor correlation. The diagonal of the matrix is continually 1 due to the fact a variable is perfectly correlated with itself.

Data Exploration

Introduction

Data exploration is a procedure of analyzing and understanding facts as a way to benefit insights into its traits, patterns, and relationships. It includes the use of diverse strategies to research and visualize facts, with the goal of figuring out developments, anomalies, and different patterns that may be hidden within the facts.

Data exploration is an vital first step in records evaluation, because it lets in analysts to benefit a deeper understanding of the data and formulate hypotheses which could manual further evaluation. It involves obligations including facts cleaning, statistics profiling, and statistics visualization.

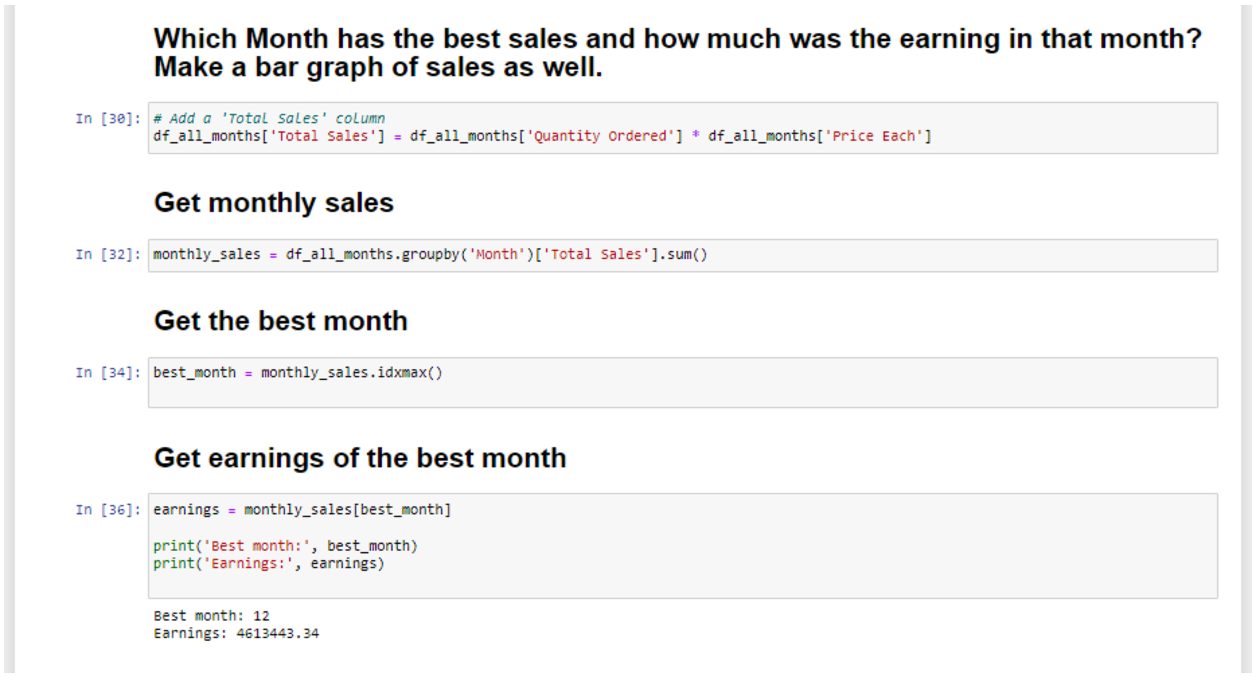
Data cleaning entails identifying and correcting errors or inconsistencies within the records, which includes missing values, duplicates, or outliers. This step is important because misguided or incomplete facts can skew the analysis effects.

Data profiling entails producing summary information and visualizations to get an overall image of the records, consisting of the variety of values, the distribution of records, and the presence of outliers.

Data visualization entails developing graphical representations of the statistics, inclusive of scatterplots, histograms, or boxplots, to visually perceive patterns and relationships within the information. Visualizations also can assist to talk the results of the evaluation in a clean and understandable manner.

Overall, records exploration is a essential step in the facts evaluation manner, as it affords a foundation for further analysis and facilitates to ensure that the results are accurate and significant.

- Which Month has the best sales? and how much was the earning in that month? Make a bar graph of sales as well.



Which Month has the best sales and how much was the earning in that month? Make a bar graph of sales as well.

```
In [30]: # Add a 'Total Sales' column
         df_all_months['Total Sales'] = df_all_months['Quantity Ordered'] * df_all_months['Price Each']
```

Get monthly sales

```
In [32]: monthly_sales = df_all_months.groupby('Month')['Total Sales'].sum()
```

Get the best month

```
In [34]: best_month = monthly_sales.idxmax()
```

Get earnings of the best month

```
In [36]: earnings = monthly_sales[best_month]

         print('Best month:', best_month)
         print('Earnings:', earnings)

         Best month: 12
         Earnings: 4613443.34
```

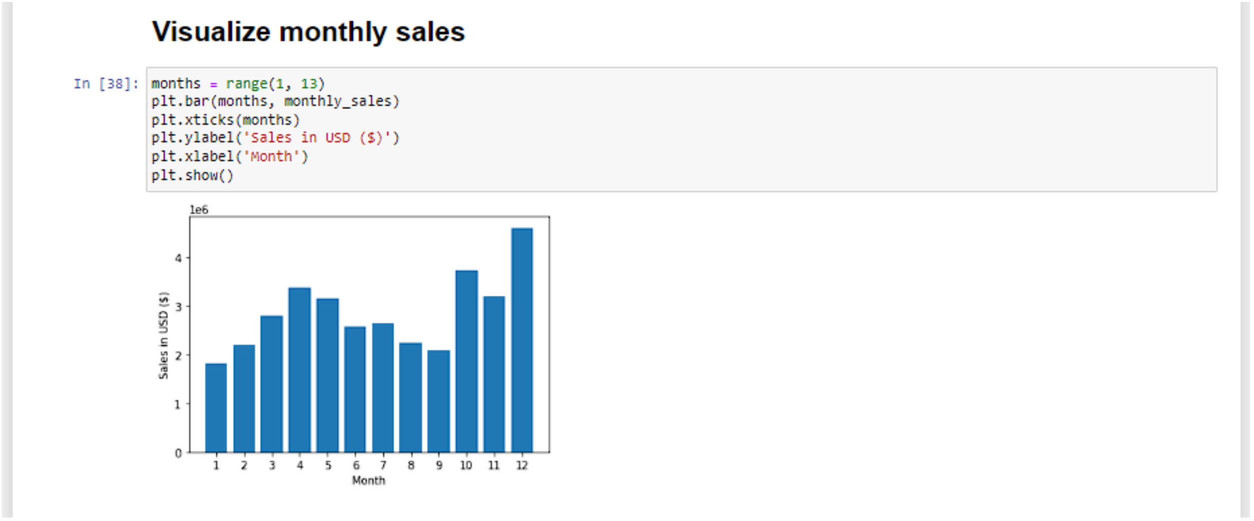
Calculates the entire sales and income for every month and identifies the month with the very best income.

- A brand new column called 'Total Sales' is brought to the DataFrame via multiplying the 'Quantity Ordered' and 'Price Each' columns. Then, the DataFrame is grouped through month the usage of the 'groupby' approach, and the sum of the 'Total Sales' column is calculated for each month using the 'sum' technique.

- The 'idxmax' approach is used to become aware of the month with the best overall sales, and the '[]' notation is used to retrieve the corresponding profits cost.

Effects are published to the console the use of the 'print' characteristic. The 'best_month' variable consists of the month with the very best profits, and the 'profits' variable includes the corresponding profits value.

- Which city has sold the highest product?



Visualize monthly sales

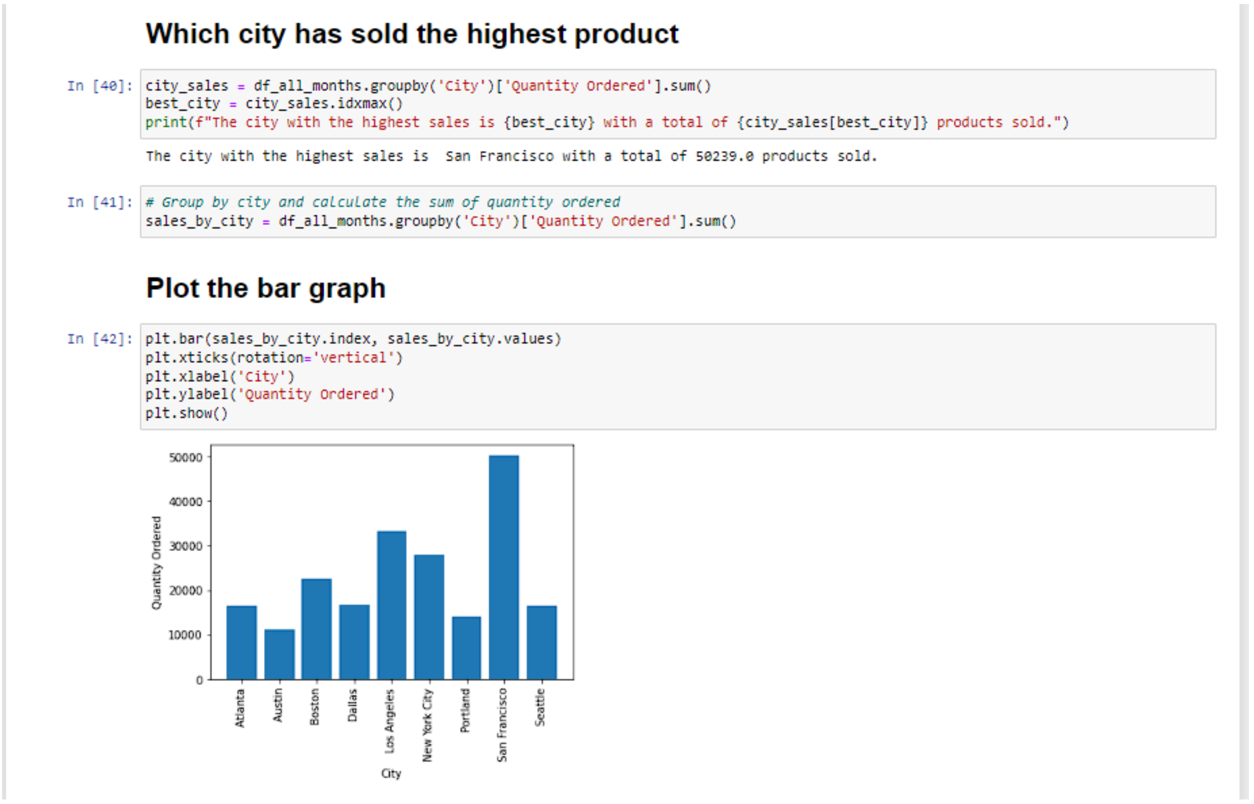
```
In [38]: months = range(1, 13)
         plt.bar(months, monthly_sales)
         plt.xticks(months)
         plt.ylabel('Sales in USD ($)')
         plt.xlabel('Month')
         plt.show()
```

Makes use of Matplotlib to create a bar chart that suggests the full sales for each month within the statistics.

- The first line defines a number of numbers from 1 to 12 that represents the months.
- The 2nd line creates a vertical bar chart using the plt.Bar() function. The x-axis shows the months, at the same time as the y-axis suggests the full income in USD ($).
- The third line sets the tick labels for the x-axis to be the numbers inside the months list.
- The fourth line adds a label for the y-axis.
- The 5th line adds a label for the x-axis.

Finally, the plt.Display() characteristic is referred to as to display the chart.

- Which product was sold the most in overall? Illustrate it through bar graph.



Which city has sold the highest product

```
In [40]: city_sales = df_all_months.groupby('City')['Quantity Ordered'].sum()
         best_city = city_sales.idxmax()
         print(f"The city with the highest sales is {best_city} with a total of {city_sales[best_city]} products sold.")

         The city with the highest sales is  San Francisco with a total of 50239.0 products sold.
```

```
In [41]: # Group by city and calculate the sum of quantity ordered
         sales_by_city = df_all_months.groupby('City')['Quantity Ordered'].sum()
```

Plot the bar graph

```
In [42]: plt.bar(sales_by_city.index, sales_by_city.values)
         plt.xticks(rotation='vertical')
         plt.xlabel('City')
         plt.ylabel('Quantity Ordered')
         plt.show()
```

First agencies the sales facts by using metropolis using the groupby feature and calculates the full quantity of merchandise offered in every town the usage of the sum function. It then reveals the city with the highest sales by using using the idxmax function on the resulting Series.

After printing the town with the highest sales, the code creates a bar chart showing the amount of merchandise ordered in each metropolis. This is finished with the aid of plotting the values of the sales_by_city series on the y-axis and the index values (i.E., the metropolis names) on the x-axis. The rotation='vertical' parameter rotates the x-axis labels vertically for better clarity. Finally, the x- and y-axis labels are added to the plot the usage of the xlabel and ylabel capabilities.

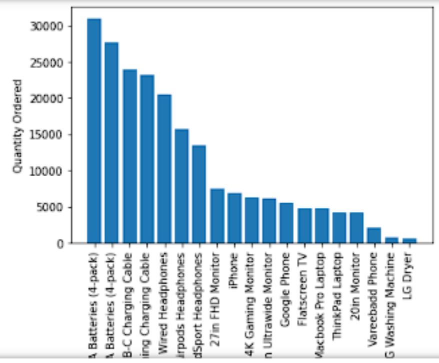
- Write a Python program to show histogram plot of any chosen variables. Use proper labels in the graph.

Which product was sold the most in overall? Illustrate it through bar graph.

```
In [44]: product_sales = df_all_months.groupby('Product')['Quantity Ordered'].sum().sort_values(ascending=False)
```

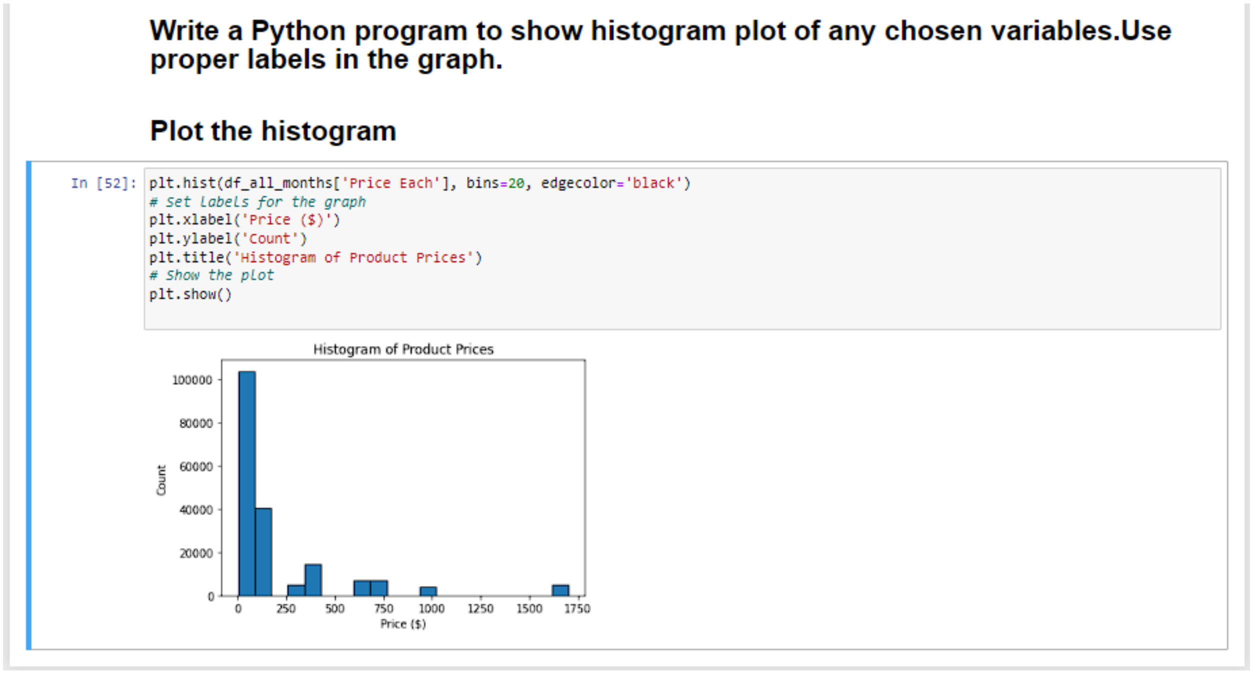
Plot the bar graph

```
In [45]: plt.bar(product_sales.index, product_sales.values)
         plt.xticks(rotation='vertical')
         plt.xlabel('Product')
         plt.ylabel('Quantity Ordered')
         plt.show()
```



Growing a bar chart of the full amount of every product ordered throughout all months. The groupby() technique is used to group the information through product, and then the sum() approach is used to calculate the overall amount ordered for each product. The resulting Series is then sorted in descending order the usage of the sort_values() method. Finally, a bar chart is created the use of the plt.Bar() feature, with the product names as the x-axis labels and the overall quantity ordered as the y-axis values. The plt.Xticks(rotation='vertical') feature is used to rotate the x-axis labels for higher visibility. The x-axis represents the products and the y-axis represents the amount ordered.

- Write a Python program to show histogram plot of any chosen variables. Use proper labels in the graph.



Write a Python program to show histogram plot of any chosen variables.Use proper labels in the graph.

Plot the histogram

In [52]:
```
plt.hist(df_all_months['Price Each'], bins=20, edgecolor='black')
# Set labels for the graph
plt.xlabel('Price ($)')
plt.ylabel('Count')
plt.title('Histogram of Product Prices')
# Show the plot
plt.show()
```

Creates a histogram of the 'Price Each' column within the df_all_months dataframe using Matplotlib. The histogram is divided into 20 packing containers and the brink colour of the bars is about to black. The x-axis label is ready to 'Price ($)', the y-axis label is set to 'Count', and the title of the plot is set to 'Histogram of Product Prices'. Finally, the plot is displayed using the plt.Display() function.

Conclusion

Data understanding is a vital step in the facts mining technique, because it enables ensure that the facts getting used for analysis is accurate, applicable, and meaningful. By examining and exploring the data, records scientists can gain a deeper know-how of its shape, content, and relationships. This lets in them to identify styles, trends, and outliers within the statistics, as well as check the excellent of the statistics and pick out any issues that can effect the evaluation.

During the information expertise section, information scientists usually accumulate the information from inner or outside resources and describe it in phrases of its size, layout, and structure. They additionally behavior an preliminary exploration of the facts to identify any ability problems or regions of interest. This can involve producing summary data, visualizations, and different exploratory information analysis techniques.

Another important assignment throughout the statistics information section is information pleasant evaluation. This entails assessing the pleasant of the statistics and identifying any troubles that could effect the analysis. This can consist of identifying lacking or incomplete data, data access mistakes, or other problems.

Once data excellent has been assessed, the facts training phase can start. This involves getting ready the facts for analysis by using cleaning, transforming, and integrating the information. This can involve duties consisting of facts cleaning, function choice, and function engineering. By well making ready the facts, data scientists can ensure that the analysis is based on accurate and dependable records.

In end, data know-how is a critical step inside the records mining technique that facilitates make certain that the data getting used for evaluation is accurate, applicable, and significant. By gaining a deeper information of the information, records scientists could make greater informed decisions and generate extra accurate insights, leading to higher business consequences.

Bibliography

Answers, P.—Q. a. (2023, 04 29). *Pandas Tutorial*. Retrieved from w3school : https://www.w3schools.com/python/pandas/default.asp

python. (2023, 04 28). *Using Python on Windows*. Retrieved from Python: https://docs.python.org/3/using/windows.html