Descriptive Analytics (Data Summarization)

Ms Swapnil Shrivastava

C-DAC Bangalore

Content

- Data Summarization Overview
- Frequency Distribution
  - Categorical and Continuous data
  - Histogram
  - Relative and Cumulative Relative Frequency
- Percentiles and Quartiles
- Cross Tabulations
- Summary
- References

Sorting Data Values

- sort_values() takes the column names, based on which the records need to be sorted.
- By default, sorting is done in ascending order.
- To sort the DataFrame records in descending order, pass False to ascending parameter.
- Sort Purchase Order dataset records by Supplier and Cost per order.

Grouping and Aggregating

- This would group records based on column values and then apply aggregated operations such as maximum, minimum etc:
- Use the group by method for grouping and methods like mean() and max() for aggregating.
- To create a DataFrame we can call reset_index() on the returned data structure.
- Find the average cost per order for each Supplier.

Filtering Records

- DataFrame records can be filtered using a condition as indexing mechanism.
- Those records for which the condition returns True are selected to be part of the resulting DataFrame.
- Suppose we wish to identify all records in the Purchase Orders dataset whose item cost is atleast $200.

Handling Missing Values

- In real world, the datasets are not clean and may have missing values.
- We must know how to find and deal with these missing values.
- One of the strategies to deal with missing values is to remove them from the dataset.
- dropna() method is used to remove rows with null values.

Joining DataFrames

- To combine columns from multiple DataFrames into one single DataFrame.
- In this case both DataFrames need to have a common column.
- The merge() method is called from one of the DataFrames and other DataFrame is passed as a parameter.
- Create two dataframes using groupby method and join them using merge functionality.

Cost per order example

	Supplier	Cost per order_x	Cost per order_y
0	Alum Sheeting	53478.750000	127500.0
1	Durrable Products	36593.519231	121000.0
2	Fast-Tie Aerospace	14963.333333	30625.0
3	Hulkey Fasteners	41256.250000	82875.0
4	Manley Valve	11167.159091	81937.5
5	Pylon Accessories	6420.000000	7425.0
6	Spacetime Technologies	11108.895833	17250.0
7	Steelpin Inc.	29111.666667	96750.0

```
supplier_cost_mean=df.groupby("Supplier ")["Cost per order"].mean().reset_index()
supplier_cost_max=df.groupby("Supplier ")["Cost per order"].max().reset_index()

supplier_cost_summary=supplier_cost_mean.merge(supplier_cost_max,on="Supplier ")
print(supplier_cost_summary)
```

Frequency Distribution

- A frequency distribution is a table that shows the number of observations in each of several non overlapping groups.
- Categorical variables naturally define the groups in a frequency distribution.
- In Purchase Order Excel file, calculate frequency distribution of the items for which orders were placed.
- We need to find the number of orders that were placed for each item category.

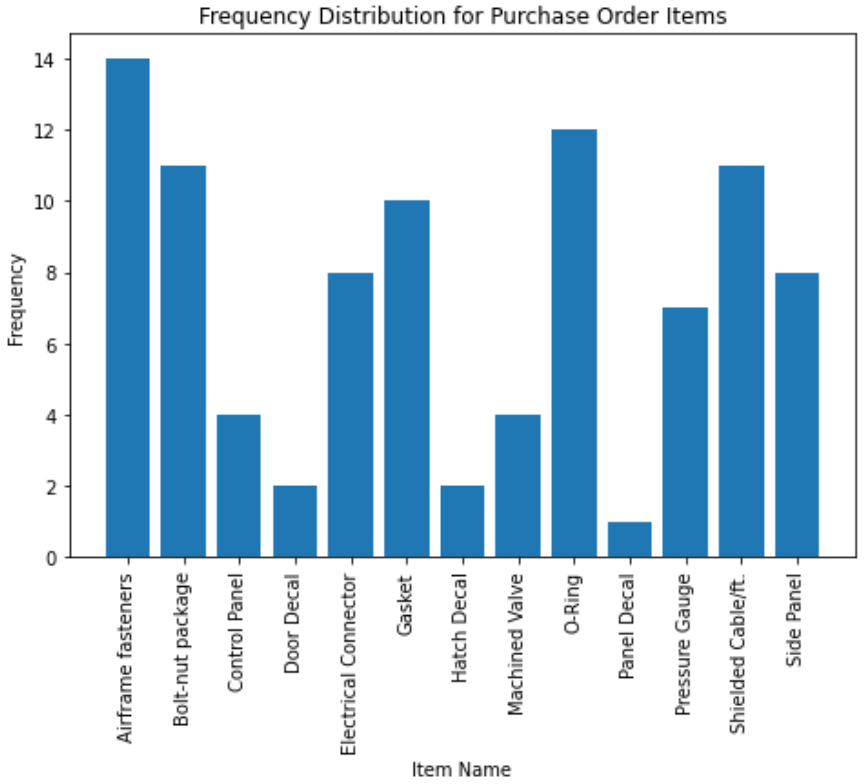
Frequency Distribution for Items in the Purchase Order dataset

```
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats

df=pd.read_excel("F:\Business
Analytics\Data_Files\Purchase
Orders.xlsx", skiprows=2)
freq_dist=df["Item
Description"].value_counts().sort_index()
print(freq_dist)
```

Item Description	Frequency
Airframe fasteners	14
Bolt-nut package	11
Control Panel	4
Door Decal	2
Electrical Connector	8
Gasket	10
Hatch Decal	2
Machined Valve	4
O-Ring	12
Panel Decal	1
Pressure Gauge	7
Shielded Cable/ft.	11
Side Panel	8

Column Chart for Frequency Distribution of items purchased

```
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
ax.bar(freq_dist.index.tolist(),freq_dist.tolist())
plt.xticks(rotation=90)
plt.ylabel('Frequency')
plt.xlabel('Item Name')
plt.title('Frequency Distribution for Purchase Order Items')

plt.show()
```



Relative Frequency Distribution

- It is to express frequencies as a fraction, or proportion of the total.
- If a data set has n observations, the relative frequency of category I is computed as

$$\text{relative frequency of category i} = \frac{\text{frequency of category i}}{n}$$

- A relative frequency distribution is a tabular summary of the relative frequencies of all categories.

Relative Frequency Distribution for Items Purchased

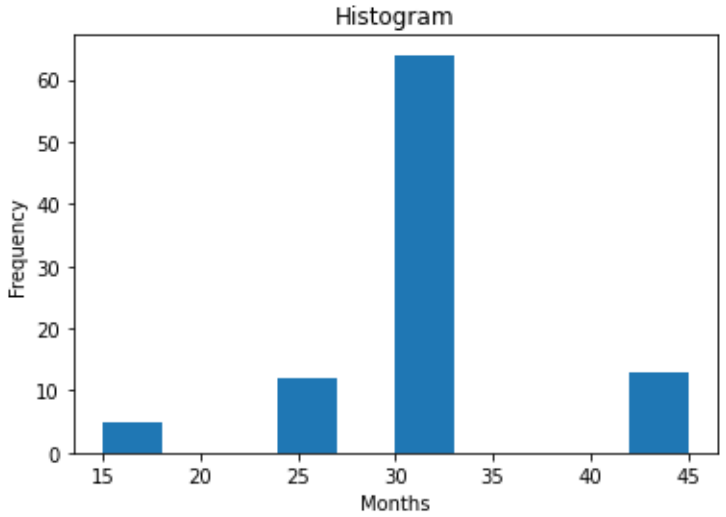
	Item Description	Frequency	Relative Frequency
0	Airframe fasteners	14	0.148936
1	Bolt-nut package	11	0.117021
2	Control Panel	4	0.042553
3	Door Decal	2	0.021277
4	Electrical Connector	8	0.085106
5	Gasket	10	0.106383
6	Hatch Decal	2	0.021277
7	Machined Valve	4	0.042553
8	O-Ring	12	0.127660
9	Panel Decal	1	0.010638
10	Pressure Gauge	7	0.074468
11	Shielded Cable/ft.	11	0.117021
12	Side Panel	8	0.085106

```
freq_dist_df=freq_dist.to_frame().reset_index()
count=len(df)
freq_dist_df["Relative Frequency"]= freq_dist_df[freq_dist_df.columns[1]]/count
print(freq_dist_df)
```

Histogram

- A histogram is a plot that shows the frequency distribution of a set of continuous variable.
- Histogram gives an insight into the underlying distribution pf the variable, outlier, skewness etc:
- Find frequency distribution of A/P terms in Purchase Order dataset.

```
plt.hist(df["A/P Terms (Months)"])
plt.ylabel('Frequency')
plt.xlabel('Months')
plt.title('Histogram')
```



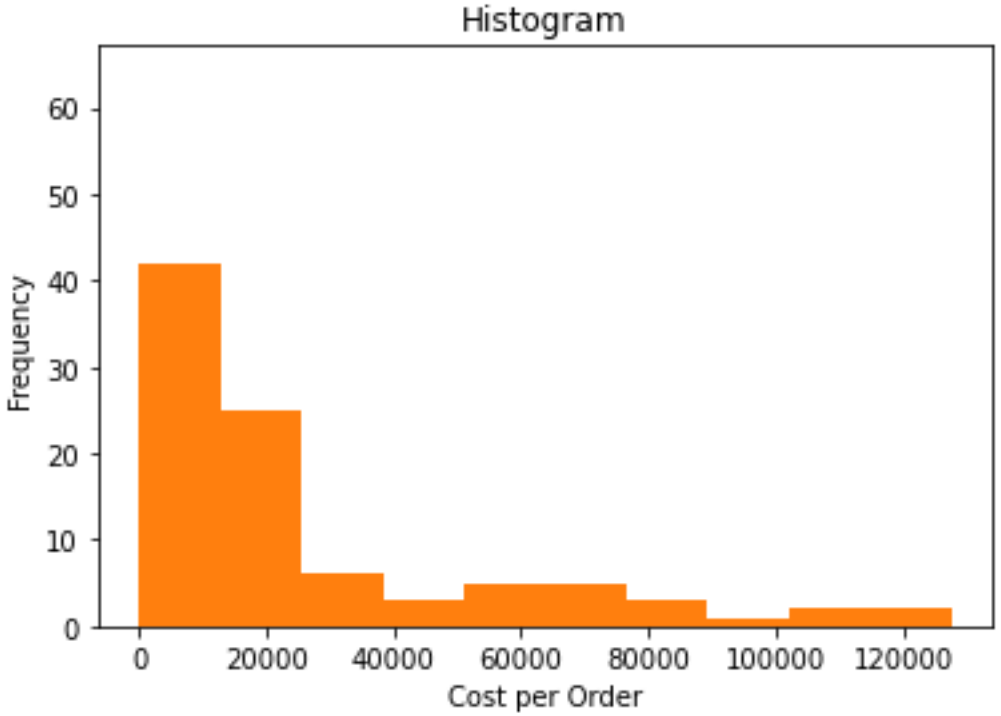
Frequency Distribution for Continuous Data

- It requires that we define by specifying
  - The number of groups
  - The width of each group, and
  - The upper and lower limits of each group
- Lower limit of the first group (LL) is a whole number smaller than the minimum data value
- Upper limit of the last group (UL) as a whole number larger than the maximum data value.

$$\text{Group width} = \frac{UL - LL}{\text{number of groups}}$$

Constructing Histogram for Cost per Order

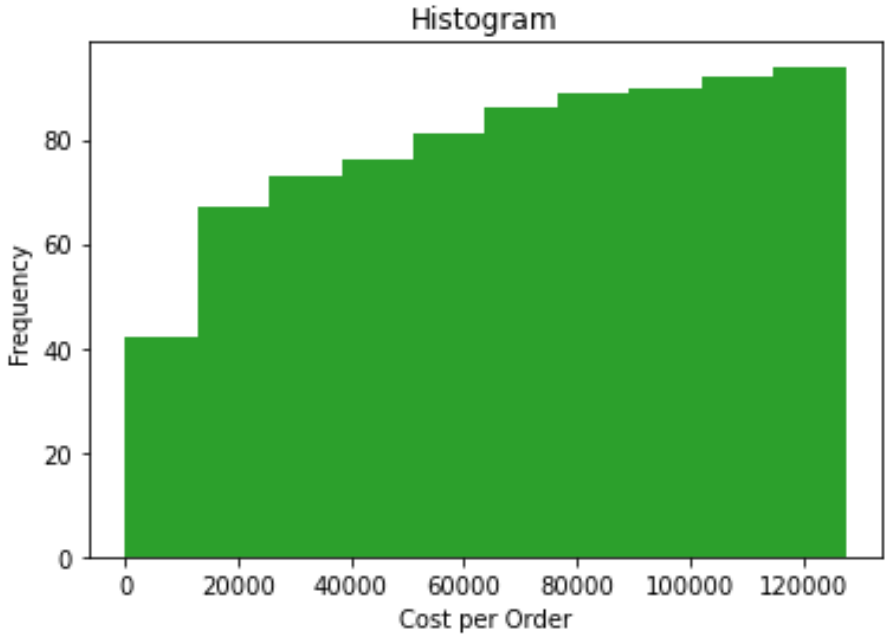
```
plt.hist(df['Cost per order'],bins=10)
plt.ylabel('Frequency')
plt.xlabel('Cost per Order')
plt.title('Histogram')
```



Cumulative Relative Frequency

- It is obtained by summing all the relative frequencies at or below each upper limit.
- It represents the proportion of observations that fall at or below a certain value.
- Cumulative relative frequency distribution is the tabular summary of cumulative relative frequencies.
- Ogive is a chart for the cumulative relative frequency.

Cumulative Relative Frequency for Cost per Order Data




```
freq_dist_df['Cumulative Frequency'] = freq_dist_df["Relative Frequency"].cumsum()

plt.hist(df['Cost per order'],bins=10, cumulative=True)
```

Percentiles

- Standardized tests used for college or graduate school entrance examinations (SAT, ACT, GMAT, GRE etc)
- Percentiles specify the percent of other test takers who scored at or below the score of a particular individual.
- The kth percentile is a value at or below which at least k percent of the observation lie.
- Scipy module stats.percentileofscore computes percentile of a given score.
- Scipy module stats.scoreatpercentile computes the score at a given percentile.
- Calculate the 90^{th} percentile for cost per order in Purchase Order dataset.

Quartiles

- Quartiles break the data into four parts
  - 25^{th} percentile is called the first quartile Q1. One fourth of data fall below Q1
  - 50^{th} percentile is called the second quartile. Half of the data fall below Q2.
  - 75^{th} percentile is called the third quartile Q3. Three fourth are below Q3.
  - 100^{th} percentile is called the fourth quartile Q4.
- Numpy percentile method could be used to compute quartiles.
- Compute the quartiles for cost per order data in the Purchase Orders database.

Cross Tabulations

- It is a tabular method that displays the number of observations in a data set for different subcategories of two categorical variables.
- It is often called contingency table
- The subcategories of the variables must mutually exclusive and exhaustive.
- Used in marketing research to provide insight into characteristics of different market segments using categorical variables such as gender education level, marital status and so on.

Exploring Data Using Crosstab

Python pandas cross tabulation features will help find occurrences for the combination of values for two columns.

```
import pandas as pd

df=pd.read_excel("F:\Business Analytics\Data_Files\Sales Transactions.xlsx", skiprows=2)
print(df)
print(pd.crosstab(df["Region"],df["Product"]))
```

Product	Book	DVD
Region		
East	56	42
North	43	42
South	62	37
West	100	90

Summary

- Making sense of large quantities of disparate data is necessary not only for gaining competitive advantage in today's business environment but also surviving data.
- Data Visualization is important for building decision models and for interpreting their results.
- Frequency distribution, histograms, and cross-tabulations are tabular and visual tools of descriptive statistics.

References

- James Evans, Business Analytics: Methods, Models and Decisions, Second Edition, Pearson Publication, 2017.
- Manaranjan Pradhan and U Dinesh Kumar, Machine Learning using Python, Wiley Publication, 2019.
- U Dinesh Kumar, Business Analytics- The Science of Data-Driven Decision Making, Wiley Publication, 2017.
- Python pandas: https://pandas.pydata.org/
- Matplotlib: Visualization with Python https://matplotlib.org/