21CSE11- EXPLORATORY DATA ANALYSIS

Unit – I

EXPLORATORY DATA ANALYSIS

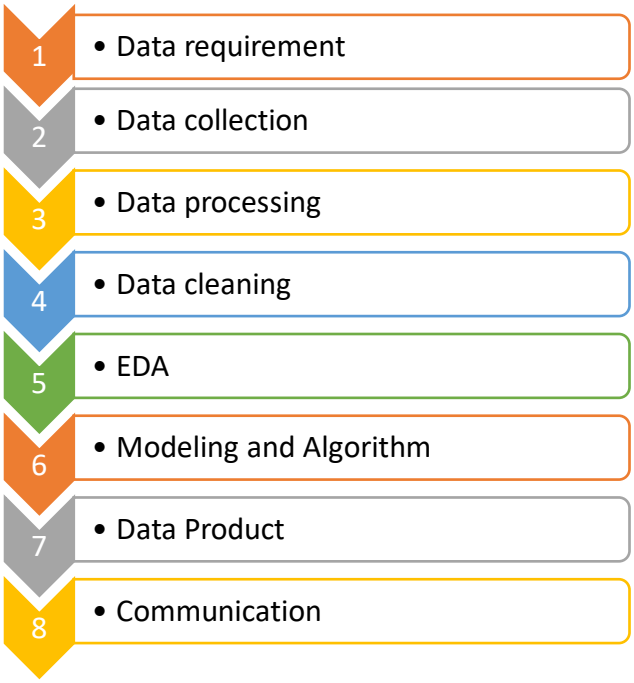
EDA fundamentals – Understanding data science – Significance of EDA – Making sense of data – Comparing EDA with classical and Bayesian analysis – Software tools for EDA - Visual Aids for EDA- Data transformation techniques-merging database, reshaping and pivoting, Transformation techniques.

1.1 EDA fundamentals

- EDA – The process is used to **generate meaningful and useful information from datasets.**
- EDA is a process of examining the available dataset to
  1. Discover patterns
  2. Spot anomalies
  3. Test hypotheses
  4. Check assumptions using statistical measures.

1.2 Understanding data science

- The stages of EDA



i. Data requirement
   - Various sources of data for an organization.
   - For example, an application tracking the sleeping pattern of patients suffering from dementia requires several types of sensors' data storage, such as sleep data, heart rate from the patient, electro-dermal activities, and user activities pattern.

ii. Data collection
   - Data collected from several sources must be stored in the correct format and transferred to the right information technology

iii. Data processing

- Preprocessing involves the process of pre-curating the dataset before actual analysis.
- Common tasks involve correctly exporting the dataset, placing them under the right tables, structuring them, and exporting them in the correct format.

iv. Data cleaning
- Preprocessed data is still not ready for detailed analysis. It must be correctly transformed for an incompleteness check, duplicates check, error check, and missing value check. These tasks are performed in the data cleaning stage, which involves responsibilities such as matching the correct record, finding inaccuracies in the dataset, understanding the overall data quality, removing duplicate items, and filling in the missing values.

v. EDA
- Exploratory data analysis, as mentioned before, is the stage where we actually start to understand the message contained in the data. It should be noted that several types of data transformation techniques might be required during the process of exploration.

vi. Modeling and Algorithm
- From a data science perspective, generalized models or mathematical formulas can represent or exhibit relationships among different variables, such as correlation or causation. These models or equations involve one or more variables that depend on other variables to cause an event.

vii. Data Product
- Any computer software that uses data as inputs, produces outputs, and provides feedback based on the output to control the environment is referred to as a data product. A data product is generally based on a model developed during data analysis, for example, a recommendation model that inputs user purchase history and recommends a related item that the user is highly likely to buy.

viii. Communication
- This stage deals with disseminating the results to end stakeholders to use the result for business intelligence. One of the most notable steps in this stage is data visualization. Visualization deals with information relay techniques such as tables, charts, summary diagrams, and bar charts to show the analyzed result.
- To be certain of the insights that the collected data provides and to make further decisions, data mining is performed where we go through distinctive analysis processes.

1.3 Significance of EDA

- Different fields of science, economics, engineering, and marketing accumulate and store data primarily in electronic databases.
- Appropriate and well-established decisions should be made using the data collected. It is practically impossible to make sense of datasets containing more than a handful of data points without the help of computer programs.
- Steps in EDA
    1. Problem definition: It is the driving force for a data analysis plan execution.

2. Data preparation: This step involves methods for preparing the dataset before actual analysis.
3. Data analysis: This is one of the most crucial steps that deals with descriptive statistics and analysis of the data. The main tasks involve summarizing the data, finding the hidden correlation and relationships among the data, developing predictive models, evaluating the models, and calculating the accuracies.

4. Development and representation of the results: This step involves presenting the dataset to the target audience in the form of graphs, summary tables, maps, and diagrams. Most of the graphical analysis techniques include scattering plots, character plots, histograms, box plots, residual plots, mean plots, and others.

1.4 Making sense of data

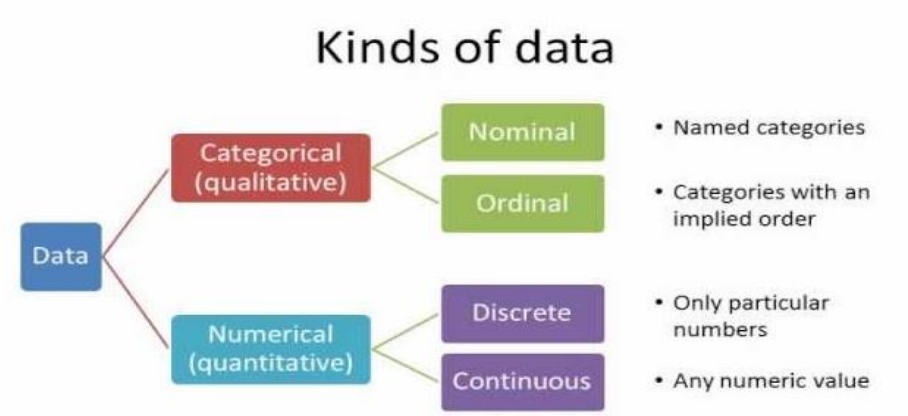
- It is crucial to identify the type of data under analysis.
- Different disciplines store different kinds of data for different purposes.
- For example, medical researchers store patients' data, universities store students' and teachers' data, and real estate industries storehouse and building datasets.
- A dataset contains many observations about a particular object.
- For instance, a dataset about patients in a hospital can contain many observations. A patient can be described by a patient identifier (ID), name, address, weight, date of birth, address, email, and gender.
- Each of these features that describes a patient is a variable.
- Each observation can have a specific value for each of these variables.
- For example, a patient can have the following:

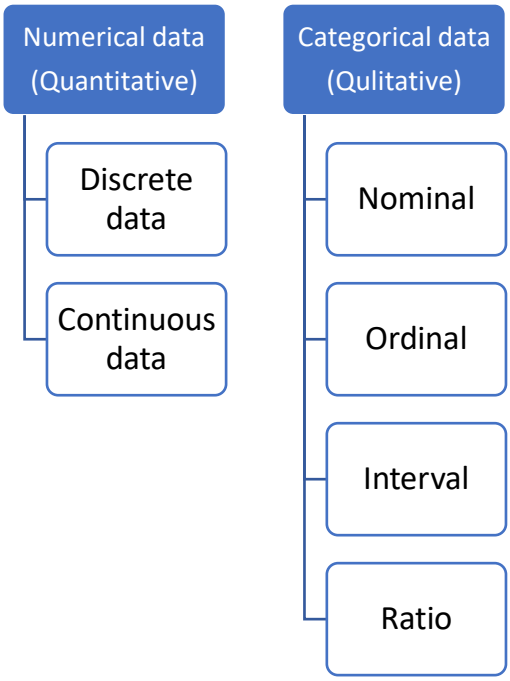
```
PATIENT_ID = 1001
Name = Yoshmi Mukhiya
Address = Mannsverk 61, 5094, Bergen, Norway
Date of birth = 10th July 2018
Email = yoshmimukhiya@gmail.com
Weight = 10
Gender = Female
```

- These datasets are stored in hospitals and are presented for analysis.
- Most of this data is stored in some sort of database management system in tables/schema.
- An example of a table for storing patient information is shown here:

PATIENT_ID	NAME	ADDRESS	DOB	EMAIL	Gender	WEIGHT
001	Suresh Kumar Mukhiya	Mannsverk, 61	30.12.1989	skmu@hvl.no	Male	68
002	Yoshmi Mukhiya	Mannsverk 61, 5094, Bergen	10.07.2018	yoshmimukhiya@gmail.com	Female	1
003	Anju Mukhiya	Mannsverk 61, 5094, Bergen	10.12.1997	anjumukhiya@gmail.com	Female	24
004	Asha Gaire	Butwal, Nepal	30.11.1990	aasha.gaire@gmail.com	Female	23
005	Ola Nordmann	Danmark, Sweden	12.12.1789	ola@gmail.com	Male	75

- To summarize the preceding table, there are four observations (001, 002, 003, 004, 005).
- Each observation describes variables (PatientID, name, address, dob, email, gender, and weight).
- Most of the dataset broadly falls into two groups—numerical data and categorical data.





1. Numerical data

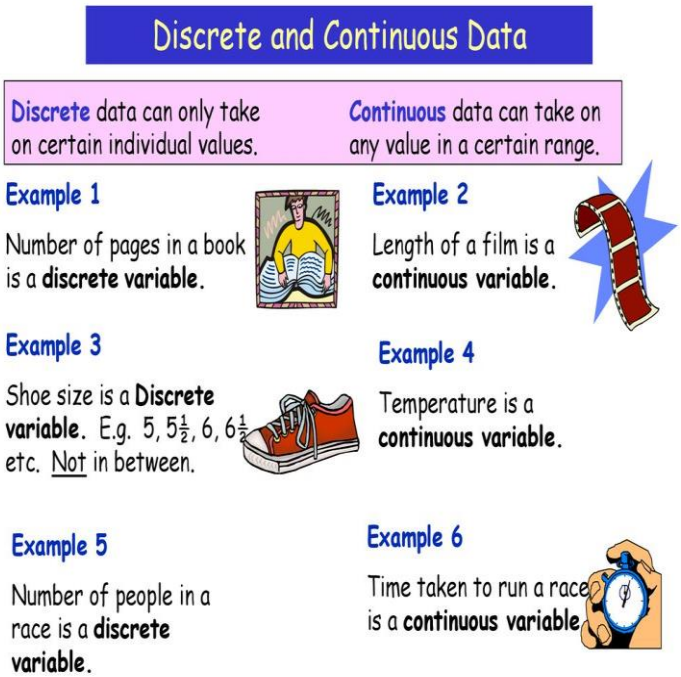
- Data that present in number form, and it doesn't include any language or descriptive form.
- For example, a person's age, height, weight, blood pressure, heart rate, temperature, number of teeth, number of bones, and the number of family members.
- Numerical data, also known as quantitative data.
- The numerical dataset can be either discrete or continuous types

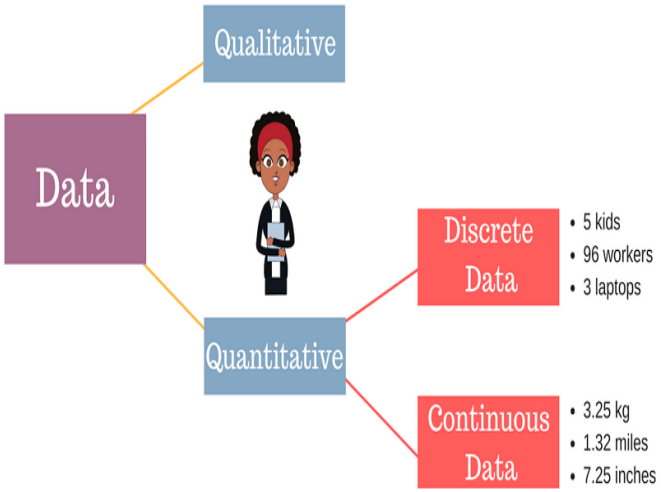
Discrete data:

- This is data that is countable and its values can be listed out.
- For example, if we flip a coin, the number of heads in 200 coin flips can take values from 0 to 200 (finite) cases.
- A variable that represents a discrete dataset is referred to as a discrete variable.
- The discrete variable takes a fixed number of distinct values.
- For example, the Country variable can have values such as Nepal, India, Norway, and Japan. It is fixed.
- The Rank variable of a student in a classroom can take values from 1, 2, 3, 4, 5, and so on.

Continuous data:

- A variable that can have an infinite number of numerical values within a specific range is classified as continuous data.
- Continuous data can follow an interval measure of scale or ratio measure of scale.
- A variable describing continuous data is a continuous variable. For example, what is the temperature of your city today?





Discrete and Continuous Data

Key characteristics of Discrete Data:

- The data can be counted.
- It is impossible to divide the values.
- The data cannot be measured.
- There are only a few possible values.
- A bar graph is used to visualize the data.

EXAMPLES OF DISCRETE DATA

Number of books

Number of people

Number of Billiard balls

Key characteristics of Continuous Data:

- Continuous variables are not counted.
- The data is measurable.
- An infinite number of possible values.
- Histograms are used to represent data graphically.

EXAMPLES OF CONTINUOUS DATA

Weight of a new born baby

Body temperature

Speed of a horse

START OF THE DAY

G6
Basic

It's the first day of spring. I'm enjoying the gentle breeze and swaying leaves. Mark the box to indicate whether the following are discrete or continuous data.

4. Height of the tree
   - ❑ Discrete
   - ❑ Continuous

5. Temperature of the day.
   - ❑ Discrete
   - ❑ Continuous

6. Height of the kite above the ground
   - ❑ Discrete
   - ❑ Continuous

3. Number of petals
   - ❑ Discrete
   - ❑ Continuous

1. Number of flowers
   - ❑ Discrete
   - ❑ Continuous

2. Speed of the bike
   - ❑ Discrete
   - ❑ Continuous

2. **Categorical data**

- This type of data **represents the characteristics of an object;**
- For example,
    1. gender, marital status, type of address, or categories of the movies.
    2. Blood type (A, B, AB, or O)
    3. Movie genres (Action, Adventure, Comedy, Crime, Drama, Fantasy, Historical, Horror, Mystery, Philosophical, Political, Romance, Saga, Satire, Science Fiction, Social, Thriller, Urban, or Western)
- A variable describing categorical data is referred to as a categorical variable.
- This data is often referred to as **qualitative datasets** in statistics.
- There are different types of categorical variables:
    1. **Binary categorical variable or dichotomous variable**
        - For example, when you create an experiment, the result is either success or failure. Hence, results can be understood as a binary categorical variable.
    2. **Polytomous variables:**
        - It can **take more than two possible values**.
        - For example, marital status can have several values, such as annulled, divorced, interlocutory, legally separated, married, polygamous, never married, domestic partners, unmarried, widowed, domestic partner, and unknown. Since marital status can take more than two possible values, it is a polytomous variable.

Measurement scales

- There are **four different types of measurement scales** described in statistics:
    1. **Nominal**
    2. **Ordinal**
    3. **Interval**
    4. **Ratio**

1. **Nominal**

- These are practiced for **labeling variables** without any quantitative value.
- The scales are generally referred to as labels.
- And these scales are mutually exclusive and **do not carry any numerical importance**
- **E.g.**

    What is your gender?

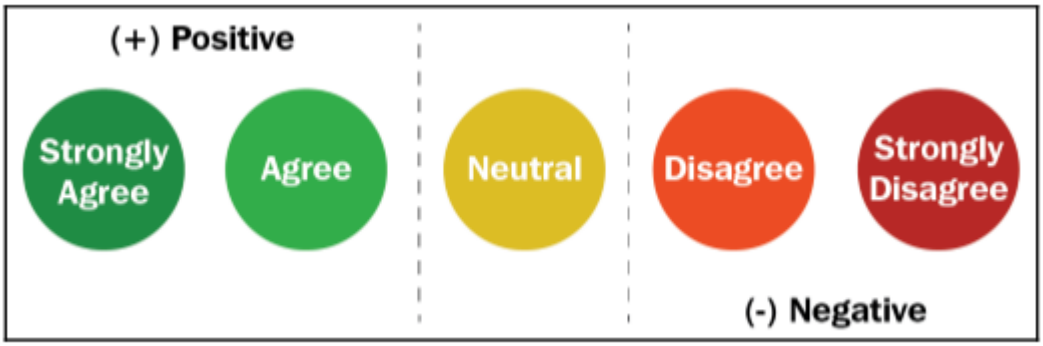
    - Male

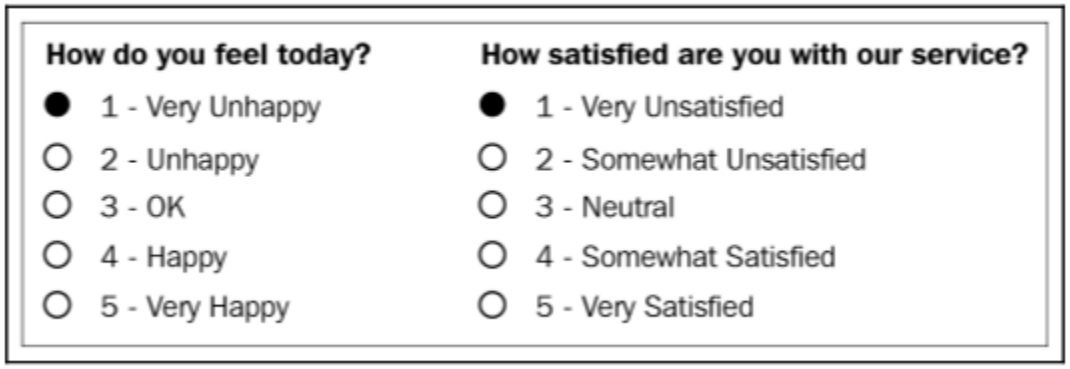
- Female
- Third gender/Non-binary
- I prefer not to answer
- Other

Other examples include the following:

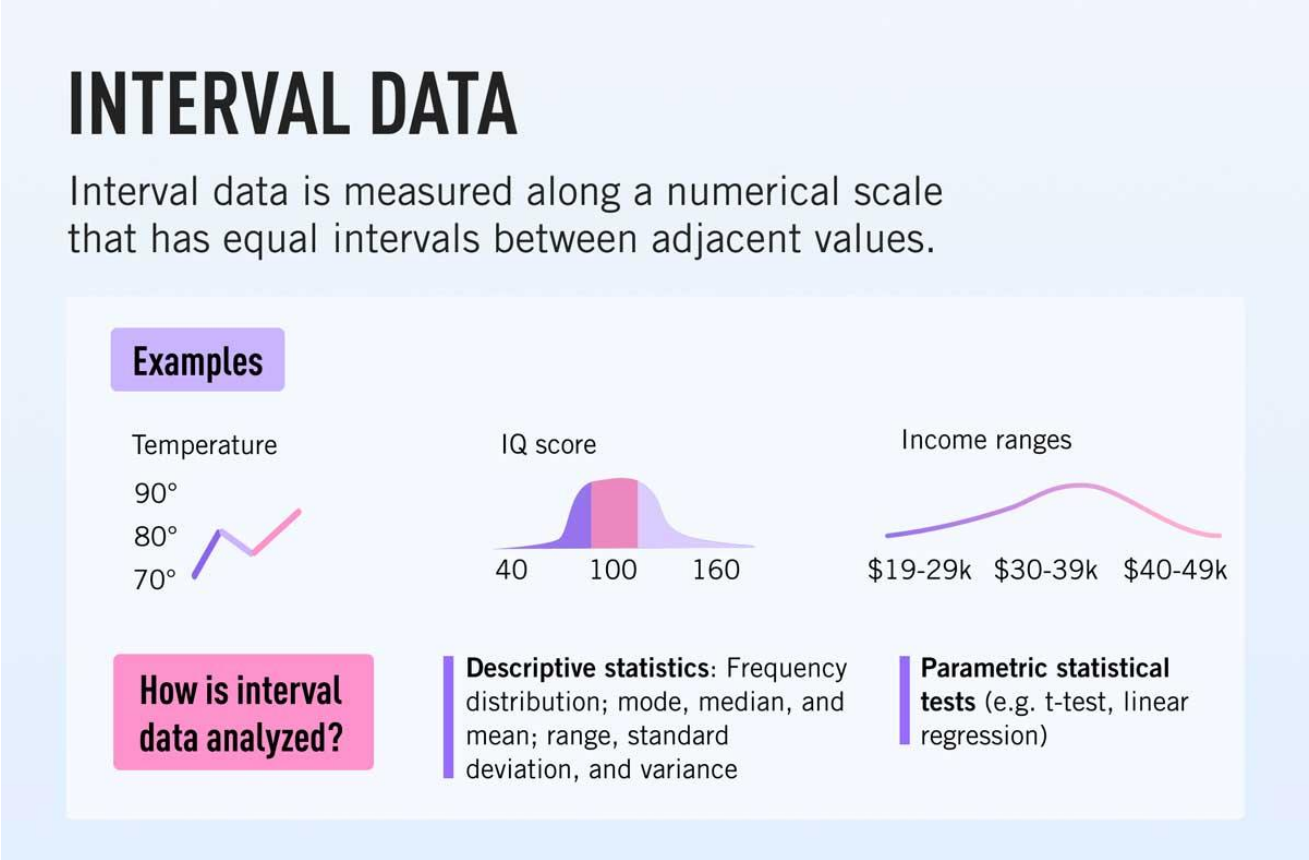
- The languages that are spoken in a particular country
- Biological species
- Parts of speech in grammar (noun, pronoun, adjective, and so on)
-
- Taxonomic ranks in biology (Archea, Bacteria, and Eukarya)

2. **Ordinal (order of ranking)**
   - ❖ The main difference in the ordinal and nominal scale is the order.
   - ❖ In ordinal scales, **the order of the values** is a significant factor. An easy tip to remember the ordinal scale is that it sounds like an order.
   - ❖ Have you heard about the **Likert scale**, which uses a variation of an ordinal scale?
   - ❖ Let's check an example of ordinal scale using the Likert scale: WordPress is making content managers' lives easier.
   - ❖ How do you feel about this statement? The following diagram shows the Likert scale.



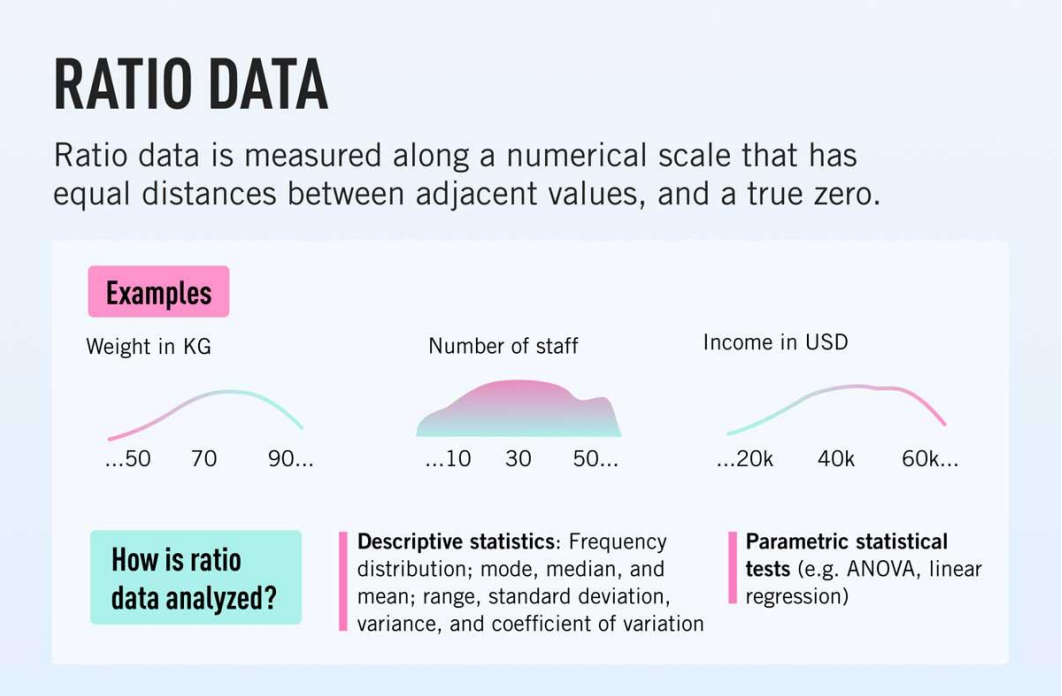


3. **Interval**
   - ❖ In interval scales, both the order and exact differences between the values are significant.
   - ❖ Interval scales are widely used in statistics, for example, in the measure of central tendencies—mean, median, mode, and standard deviations.
   - ❖ Examples include location in Cartesian coordinates and direction measured in degrees from magnetic north. The mean, median, and mode are allowed on interval data.



- ❖ Income categorized as ranges ($30-39k, $40-49k, $50-59k, and so on)

4. **Ratio**
   - ❖ Ratio scales contain order, exact values, and absolute zero, which makes it possible to be used in descriptive and inferential statistics.
   - ❖ These scales provide numerous possibilities for statistical analysis.
   - ❖ Mathematical operations, the measure of central tendencies, and the measure of dispersion and coefficient of variation can also be computed from such scales.



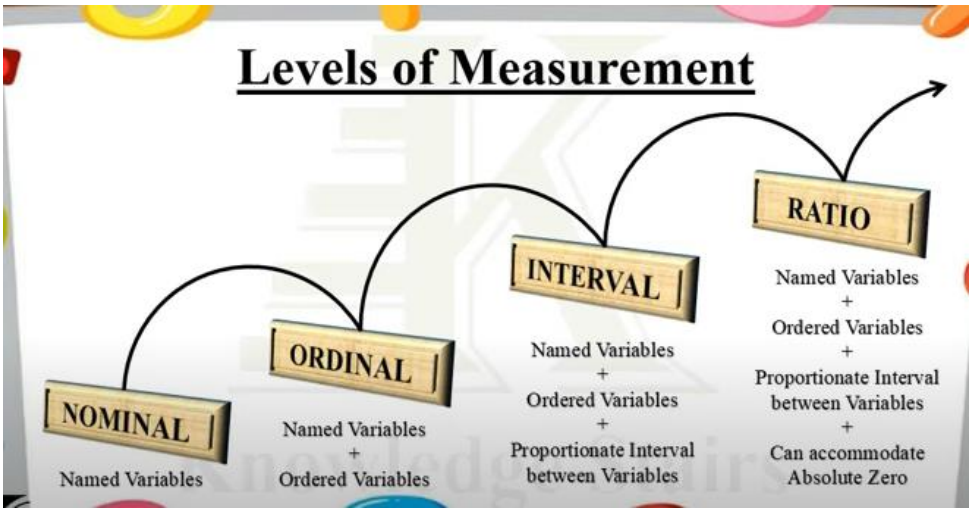
Examples of ratio data

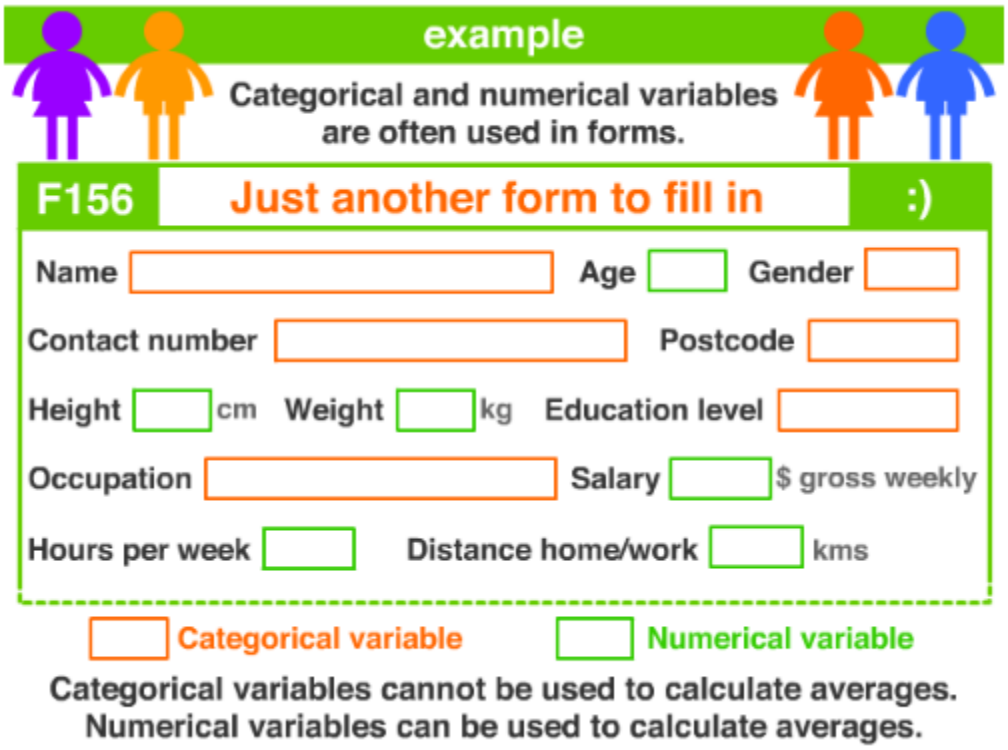
Ratio variables can be discrete (i.e. expressed in finite, countable units) or continuous (potentially taking on infinite values). Here are some examples of ratio data:

- Weight in grams (continuous)
- Number of employees at a company (discrete)
- Speed in miles per hour (continuous)
- Length in centimeters (continuous)
- Age in years (continuous)
- Income in dollars (continuous)
- Sales made in one month (discrete)

❖ Examples include a measure of energy, mass, length, duration, electrical energy, plan angle, and volume. The following table gives a summary of the data types and scale measures:

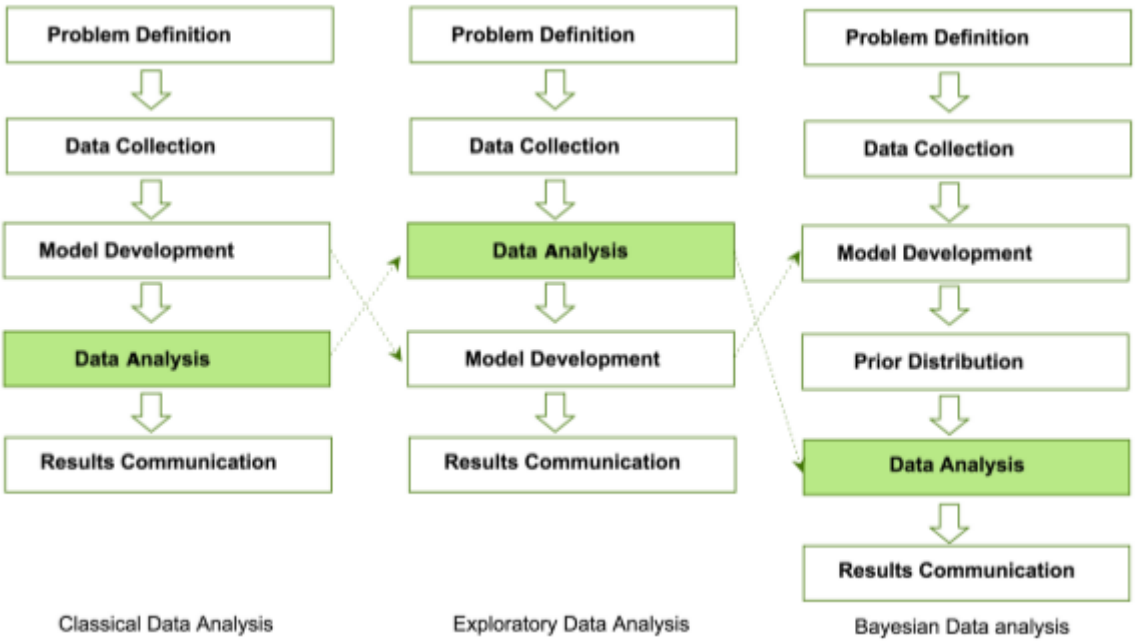
Provides:	Nominal	Ordinal	Interval	Ratio
The "order"of values is known		✓	✓	✓
"Counts," aka "Frequency of Distribution"	✓	✓	✓	✓
Mode	✓	✓	✓	✓
Median		✓	✓	✓
Mean			✓	✓
Can quantify the difference between each value			✓	✓
Can add or subtract values			✓	✓
Can multiple and divide values				✓
Has "true zero"				✓





1.5 Comparing EDA with classical and Bayesian analysis

- There are several approaches to data analysis.
- The most popular ones are the following:
  1. **Classical data analysis:** For the classical data analysis approach, the problem definition and data collection step are followed by model development, which is followed by analysis and result communication.
  2. **Exploratory data analysis approach:** For the EDA approach, it follows the same approach as classical data analysis except the model imposition and the data analysis steps are swapped. The main focus is on the data, its structure, outliers, models, and visualizations. Generally, in EDA, we do not impose any deterministic or probabilistic models on the data.
  3. **Bayesian data analysis approach:** The Bayesian approach incorporates prior probability distribution knowledge into the analysis steps as shown in the following diagram. Well, simply put, prior probability distribution of any quantity expresses the belief about that particular quantity before considering some evidence. Are you still lost with the term prior probability distribution? Andrew Gelman has a very descriptive paper about prior probability distribution.
- The following diagram shows three different approaches for data analysis illustrating the difference in their execution steps:



Techniques for Analyzing data

CLASSICAL DATA ANALYSIS

- Classical techniques are generally quantitative(set of statistical procedures that yield numeric or tabular output) in nature.
- Some of them include:

1.)ANOVA
2.)T-tests
3.)Chi-squared tests
4.)F tests

EXPLORATORY DATA ANALYSIS

- Most of EDA techniques are generally graphical.
- Some of them include:

1.)Scatter plots
2.)Histograms
3.)Box Plots
4.) Residual Plots

Classical Data Analysis

For the classical data analysis approach, the problem definition and data collection step are followed by model development, which is followed by analysis and result communication.

Exploratory Data Analysis approach

- For the EDA approach, it follows the same approach as classical data analysis except the model imposition and the data analysis steps are swapped.
- The main focus is on the data, its structure, outliers, models, and visualizations.
- Generally, in EDA, we do not impose any deterministic or

Bayesian Data Analysis Approach

- The Bayesian approach incorporates prior probability distribution knowledge into the analysis steps.
- The prior probability distribution of any quantity expresses the belief about that particular quantity before considering some evidence.

CLASSICAL STATISTICAL APPROACH

$$\mathbb{P}(HHH \mid \theta = \tfrac{3}{4}) = \left(\tfrac{3}{4}\right)^3$$

$$\mathbb{P}(HHH \mid \theta = \tfrac{1}{4}) = \left(\tfrac{1}{4}\right)^3$$

← THE PARAMETER $\theta = \frac{3}{4}$ GIVES THE MAXIMUM LIKELIHOOD FOR THE DATA SO WE GO WITH $\hat{\theta} = \frac{3}{4}$

BAYESIAN STATISTICAL APPROACH

EITHER $\theta = \frac{3}{4}$, OR $\theta = \frac{1}{4}$

$$\mathbb{P}(\theta = \tfrac{3}{4}) = \tfrac{1}{2} \quad \& \quad \mathbb{P}(\theta = \tfrac{1}{4}) = \tfrac{1}{2}$$

← PRIOR DISTRIBUTION FOR θ

∆ WE WANT TO CALCULATE

$$\mathbb{P}(\theta = \tfrac{3}{4} \mid HHH) \quad \& \quad \mathbb{P}(\theta = \tfrac{1}{4} \mid HHH)$$

WE CALCULATE THESE WITH BAYES FORMULA:

$$\mathbb{P}(\theta = \tfrac{3}{4} \mid HHH) = \mathbb{P}(HHH \mid \theta = \tfrac{3}{4}) \cdot \frac{\mathbb{P}(\theta = \frac{3}{4})}{\mathbb{P}(HHH)}$$

$$\mathbb{P}(B \mid A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)} = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} \times \frac{\mathbb{P}(B)}{\mathbb{P}(A)} = \frac{\mathbb{P}(A \mid B) \cdot \mathbb{P}(B)}{\mathbb{P}(A)}$$

$$\mathbb{P}(B \mid A) = \frac{\mathbb{P}(A \mid B) \cdot \mathbb{P}(B)}{\mathbb{P}(A)}$$ (BAYES' FORMULA)

EXAMPLE: WHAT IS THE PROBABILITY OF GETTING TWO ACES WHEN TWO CARDS ARE DRAWN FROM A PACK?
WHAT IS THE PROBABILITY THE 2nd CARD IS AN ACE? (IGNORING THE FIRST CARD).

ANSWER: PREVIOUS WAY $\binom{52}{2}, \binom{4}{2}$ $\mathbb{P}(AA) = \frac{\binom{4}{2}}{\binom{52}{2}} = \frac{6}{1326}$

NEW METHOD $A_1 = \{1\text{st card ace}\}$, $A_2 = \{2\text{nd ace}\}$

$\mathbb{P}(A_1 \cap A_2) = \mathbb{P}(A_2 | A_1) \cdot \mathbb{P}(A_1) = \frac{4}{52} \cdot \frac{3}{51}$

$$\mathbb{P}(A_2) = \mathbb{P}(A_2|A_1)\mathbb{P}(A_1) + \mathbb{P}(A_2|A_1^c)\mathbb{P}(A_1^c)$$

$$= \frac{4}{52}\cdot\frac{3}{51} + \frac{4}{51}\cdot\frac{48}{52} = \frac{4}{52}\cdot\left[\frac{3}{51} + \frac{48}{51}\right] = \frac{1}{13}$$

WE CALCULATE THESE WITH BAYES FORMULA:

$$\mathbb{P}(\theta=\tfrac{3}{4}|HHH) = \mathbb{P}(HHH|\theta=\tfrac{3}{4})\cdot\frac{\mathbb{P}(\theta=\frac{3}{4})}{\mathbb{P}(HHH)}$$

$(\frac{3}{4})^3$; $\mathbb{P}(\theta=\frac{3}{4}) = \frac{1}{2}$

- SINCE $B = (A\cap B)\cup(A^c\cap B)$

$$\mathbb{P}(B) = \mathbb{P}(A\cap B) + \mathbb{P}(A^c\cap B)$$

$$\mathbb{P}(B) = \mathbb{P}(B|A)\mathbb{P}(A) + \mathbb{P}(B|A^c)\mathbb{P}(A^c)$$

$$P(HHH) = \underbrace{P(HHH \mid \theta=\tfrac{3}{4})}_{(\frac{3}{4})^3}\underbrace{P(\theta=\tfrac{3}{4})}_{\frac{1}{2}} + \underbrace{P(HHH \mid \theta=\tfrac{1}{4})}_{(\frac{1}{4})^3}\underbrace{P(\theta=\tfrac{1}{4})}_{\frac{1}{2}} = \frac{28}{128}$$

1.6 Software tools for EDA

- There are several software tools that are available to facilitate EDA. Here, we are going to outline some of the open source tools:

1. **Python:** This is an open source programming language widely used in data analysis, data mining, and data science (https://www.python.org/). For this book, we will be using Python.
2. **R programming language:** R is an open source programming language that is widely utilized in statistical computation and graphical data analysis (https:// www.r-project.org).
3. **Weka:** This is an open source data mining package that involves several EDA tools and algorithms (https://www.cs.waikato.ac.nz/ml/weka/).
4. **KNIME:** This is an open source tool for data analysis and is based on Eclipse (https://www.knime.com/).

Getting started with EDA

Python programming	Fundamental concepts of variables, string, and data types Conditionals and functions Sequences, collections, and iterations Working with files Object-oriented programming
NumPy	Create arrays with NumPy, copy arrays, and divide arrays Perform different operations on NumPy arrays Understand array selections, advanced indexing, and expanding Working with multi-dimensional arrays Linear algebraic functions and built-in NumPy functions
pandas	Understand and create `DataFrame` objects Subsetting data and indexing data Arithmetic functions, and mapping with pandas Managing index Building style for visual analysis
Matplotlib	Loading linear datasets Adjusting axes, grids, labels, titles, and legends Saving plots
SciPy	Importing the package Using statistical packages from SciPy Performing descriptive statistics Inference and data analysis

Numpy

- For importing numpy, we will use the following code:

```
import numpy as np
```

- For creating different types of numpy arrays, we will use the following code:

```
# importing numpy
import numpy as np
# Defining 1D array
my1DArray = np.array([1, 8, 27, 64])
print(my1DArray)
# Defining and printing 2D array
my2DArray = np.array([[1, 2, 3, 4], [2, 4, 9, 16], [4, 8, 18, 32]])
print(my2DArray)
#Defining and printing 3D array
my3Darray = np.array([[[ 1, 2 , 3 , 4],[ 5 , 6 , 7 ,8]], [[ 1, 2, 3,
4],[ 9, 10, 11, 12]]])
print(my3Darray)
```

- For displaying basic information, such as the data type, shape, size, and strides of a NumPy array, we will use the following code:

```
# Print out memory address
print(my2DArray.data)
# Print the shape of array
print(my2DArray.shape)
# Print out the data type of the array
print(my2DArray.dtype)
# Print the stride of the array.
print(my2DArray.strides)
```

- For creating an array using built-in NumPy functions, we will use the following code:

```
# Array of ones

ones = np.ones((3,4))

print(ones)

# Array of zeros

zeros = np.zeros((2,3,4),dtype=np.int16)

print(zeros)

# Array with random values

np.random.random((2,2))

# Empty array emptyArray = np.empty((3,2))

print(emptyArray)

# Full array fullArray = np.full((2,2),7)

print(fullArray)

# Array of evenly-spaced values

evenSpacedArray = np.arange(10,25,5)

print(evenSpacedArray)

# Array of evenly-spaced values

evenSpacedArray2 = np.linspace(0,2,9)

print(evenSpacedArray2)
```

- For NumPy arrays and file operations, we will use the following code:

```
# Save a numpy array into file

x = np.arange(0.0,50.0,1.0)

np.savetxt('data.out', x, delimiter=',')

# Loading numpy array from text

z = np.loadtxt('data.out', unpack=True)

print(z)

# Loading numpy array using genfromtxt method

my_array2 = np.genfromtxt('data.out',  skip_header=1,
filling_values=-999) print(my_array2) For inspecting NumPy arrays, we will
use the following code:

# Print the number of `my2DArray`'s dimensions print(my2DArray.ndim)

# Print the number of `my2DArray`'s elements print(my2DArray.size)

# Print information about `my2DArray`'s memory layout
print(my2DArray.flags)

# Print the length of one array element in bytes
print(my2DArray.itemsize)

# Print the total consumed bytes by `my2DArray`'s elements
print(my2DArray.nbytes)
```

- Broadcasting is a mechanism that permits NumPy to operate with arrays of different shapes when performing arithmetic operations:

```
# Rule 1: Two dimensions are operatable if they are equal

# Create an array of two dimension A =np.ones((6, 8))

# Shape of A

print(A.shape)

# Create another array

B = np.random.random((6,8))

# Shape of B

print(B.shape)

# Sum of A and B, here the shape of both the matrix is same.

print(A + B)
```

Secondly, two dimensions are also compatible when one of the dimensions of the array is 1. Check the example given here:

```
# Rule 2: Two dimensions are also compatible when one of them is 1

# Initialize `x`

x = np.ones((3,4))

print(x)

# Check shape of `x`

print(x.shape)

# Initialize `y`

y = np.arange(4) print(y)

# Check shape of `y`

print(y.shape)

# Subtract `x` and `y`

print(x - y)
```

Lastly, there is a third rule that says two arrays can be broadcast together if they are compatible in all of the dimensions. Check the example given here:

```
# Rule 3: Arrays can be broadcast together if they are compatible in all dimensions

x = np.ones((6,8))

y = np.random.random((10, 1, 8))

print(x + y)
```

The dimensions of x(6,8) and y(10,1,8) are different. However, it is possible to add them. Why is that? Also, change y(10,2,8) or y(10,1,4) and it will give ValueError. Can you find out why? (Hint: check rule 1).

- **For seeing NumPy mathematics at work, we will use the following example:**

```
# Basic operations (+, -, *, /, %)
x = np.array([[1, 2, 3], [2, 3, 4]])
y = np.array([[1, 4, 9], [2, 3, -2]])
# Add two array
add = np.add(x, y)
print(add)
# Subtract two array
sub = np.subtract(x, y)
print(sub)
# Multiply two array
mul = np.multiply(x, y)
print(mul)
# Divide x, y
div = np.divide(x,y)
print(div)
# Calculated the remainder of x and y
rem = np.remainder(x, y)
print(rem)
```

- **Let's now see how we can create a subset and slice an array using an index:**

```
x = np.array([10, 20, 30, 40, 50])
# Select items at index 0 and 1
print(x[0:2])
# Select item at row 0 and 1 and column 1 from 2D array
y = np.array([[ 1, 2, 3, 4], [ 9, 10, 11 ,12]])
print(y[0:2, 1])
# Specifying conditions
biggerThan2 = (y >= 2)
print(y[biggerThan2])
```

Next, we will use the pandas library to gain insights from data.

Pandas

1.Use the following to set default parameters:

```
import numpy as np
import pandas as pd
print("Pandas Version:", pd.__version__)
pd.set_option('display.max_columns', 500)
pd.set_option('display.max_rows', 500)
```

2.In pandas, we can create data structures in two ways: series and dataframes

Check the following snippet to understand how we can create a dataframe from series, dictionary, and n-dimensional arrays

The following code snippet shows how we can create a dataframe from a series:

```
series = pd.Series([2, 3, 7, 11, 13, 17, 19, 23])
print(series)
# Creating dataframe from Series
series_df = pd.DataFrame({
  'A': range(1, 5),
   'B': pd.Timestamp('20190526'),
   'C': pd.Series(5, index=list(range(4)), dtype='float64'),
 'D': np.array([3] * 4, dtype='int64'),
  'E': pd.Categorical(["Depression", "Social Anxiety", "Bipolar Disorder", "Eating Disorder"]),
  'F': 'Mental health',
  'G': 'is challenging'
 })
print(series_df)
```

The following code snippet shows how to create a dataframe for a dictionary:

```
# Creating dataframe from Dictionary
dict_df = [{'A': 'Apple', 'B': 'Ball'},{'A': 'Aeroplane', 'B':
'Bat', 'C': 'Cat'}]
dict_df = pd.DataFrame(dict_df)
print(dict_df)
```

The following code snippet shows how to create a dataframe from n-dimensional arrays:

```
# Creating a dataframe from ndarrays

sdf = {

        'County':['Østfold', 'Hordaland', 'Oslo', 'Hedmark', 'Oppland',
'Buskerud'],

        'ISO-Code':[1,2,3,4,5,6],

        'Area': [4180.69, 4917.94, 454.07, 27397.76, 25192.10,
14910.94],

        'Administrative centre': ["Sarpsborg", "Oslo", "City of Oslo",
"Hamar", "Lillehammer", "Drammen"]

            }

        sdf = pd.DataFrame(sdf)

        print(sdf)
```

3. Now, let's load a dataset from an external source into a pandas DataFrame. After3. that, let's see the first 10 entries:

```
    columns = ['age', 'workclass', 'fnlwgt', 'education',

      'education_num',

         'marital_status', 'occupation', 'relationship', 'ethnicity',
'gender','capital_gain','capital_loss','hours_per_week','country_of
_origin','income']

        df =

        pd.read_csv('http://archive.ics.uci.edu/ml/machine-learning-
databas es/adult/adult.data',names=columns)

       df.head(10)
```

If you run the preceding cell, you should get an output similar to the following screenshot:

4.The following code displays the rows, columns, data types, and memory used by the dataframe:

```
df.info()
```

The output of the preceding code snippet should be similar to the following:

Output:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 32561 entries, 0 to 32560

Data columns (total 15 columns):

age 32561 non-null int64 workclass 32561 non-null object

fnlwgt 32561 non-null int64

education 32561 non-null object

education_num 32561 non-null int64

marital_status 32561 non-null object

occupation 32561 non-null object

relationship 32561 non-null object

ethnicity 32561 non-null object

gender 32561 non-null object

capital_gain 32561 non-null int64

capital_loss 32561 non-null int64

hours_per_week 32561 non-null int64

country_of_origin 32561 non-null object

income 32561 non-null object

dtypes: int64(6), object(9)

memory usage: 3.7+ MB

5.Let's now see how we can select rows and columns in any dataframe:

```
# Selects a row
df.iloc[10]
# Selects 10 rows
df.iloc[0:10]
# Selects a range of rows
df.iloc[10:15]
# Selects the last 2 rows
df.iloc[-2:]
# Selects every other row in columns 3-5
df.iloc[::2, 3:5].head()
```

6.Let's combine NumPy and pandas to create a dataframe as follows:

```
import pandas as pd
import numpy as np
np.random.seed(24)
dFrame = pd.DataFrame({'F': np.linspace(1, 10, 10)})
dFrame = pd.concat([df, pd.DataFrame(np.random.randn(10, 5),
columns=list('EDCBA'))],
                   axis=1)
dFrame.iloc[0, 2] = np.nan
dFrame
```

It should produce a dataframe table similar to the following screenshot:

7.Let's style this table using a custom rule. If the values are greater than zero, wechange the color to black (the default color); if the value is less than zero, we change the color to red; and finally, everything else would be colored green. Let's define a Python function to accomplish that:

```
# Define a function that should color the values that are less than
0
def colorNegativeValueToRed(value):
```

```
 if value < 0:
     color = 'red'
  elif value > 0:
    color = 'black'
else:
   color = 'green'
  return 'color: %s' % color
```

8.Now, let's pass this function to the dataframe. We can do this by using the style8. method provided by pandas inside the dataframe:

```
s = df.style.applymap(colorNegativeValueToRed,
 subset=['A','B','C','D','E'])
s
```

It should display a colored dataframe as shown in the following screenshot:

It should be noted that the applymap and apply methods are computationally expensive as they apply to each value inside the dataframe. Hence, it will take some time to execute. Have patience and await execution.

9.Now, let's go one step deeper. We want to scan each column and highlight the 9. maximum value and the minimum value in that column:

```
def highlightMax(s):
    isMax = s == s.max()
    return ['background-color: orange' if v else '' for v in isMax]
def highlightMin(s):
     isMin = s == s.min()
     return ['background-color: green' if v else '' for v in isMin]
```

We apply these two functions to the dataframe as follows:

```
df.style.apply(highlightMax).apply(highlightMin).highlight_null(nul
 l_color='red')
```

The output should be similar to the following screenshot:

The output should be similar to the following screenshot:

	F	E	D	C	B	A
0	1	1.32921	nan	-0.31628	-0.99081	-1.07082
1	2	-1.43871	0.564417	0.295722	-1.6264	0.219565
2	3	0.678805	1.88927	0.961538	0.104011	-0.481165
3	4	0.850229	1.45342	1.05774	0.165562	0.515018
4	5	-1.33694	0.562861	1.39285	-0.063328	0.121668
5	6	1.2076	-0.00204021	1.6278	0.354493	1.03753
6	7	-0.385684	0.519818	1.68658	-1.32596	1.42898
7	8	-2.08935	-0.12982	0.631523	-0.586538	0.29072
8	9	1.2641	0.290035	-1.97029	0.803906	1.03055
9	10	0.118098	-0.0218533	0.0468407	-1.62875	-0.392361

10.Are you still not happy with your visualization? Let's try to use another Pythonlibrary called seaborn and provide a gradient to the table:

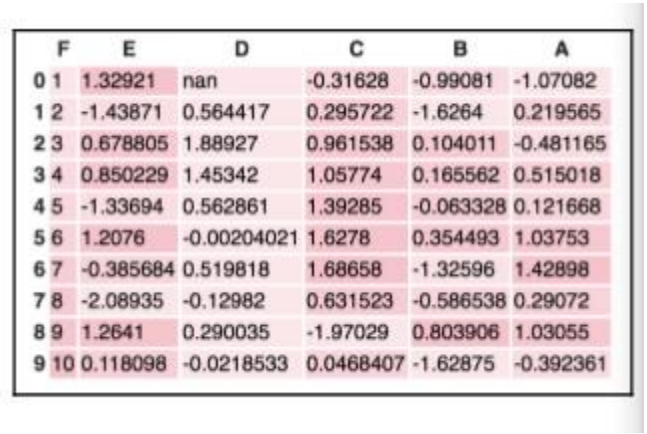
```
import seaborn as sns

colorMap = sns.light_palette("pink", as_cmap=True)

styled = df.style.background_gradient(cmap=colorMap)

 styled
```

The dataframe should have an orange gradient applied to it:



	F	E	D	C	B	A
0	1	1.32921	nan	-0.31628	-0.99081	-1.07082
1	2	-1.43871	0.564417	0.295722	-1.6264	0.219565
2	3	0.678805	1.88927	0.961538	0.104011	-0.481165
3	4	0.850229	1.45342	1.05774	0.165562	0.515018
4	5	-1.33694	0.562861	1.39285	-0.063328	0.121668
5	6	1.2076	-0.00204021	1.6278	0.354493	1.03753
6	7	-0.385684	0.519818	1.68658	-1.32596	1.42898
7	8	-2.08935	-0.12982	0.631523	-0.586538	0.29072
8	9	1.2641	0.290035	-1.97029	0.803906	1.03055
9	10	0.118098	-0.0218533	0.0468407	-1.62875	-0.392361

There are endless possibilities. How you present your result depends on you. Keep in mind that when you present your results to end stakeholders (your managers, boss, or nontechnical persons), no matter how intelligently written your code is, it is worthless to them if they cannot make sense of your program. It is widely accepted that better-visualized results are easy to market.

SciPy

SciPy is a scientific library for Python and is open source. We are going to use this library in the upcoming chapters. This library depends on the NumPy library, which provides an efficient n-dimensional array manipulation function. We are going to learn more about these libraries in the upcoming chapters. My intention here is just to inform you to get prepared to face other libraries apart from NumPy and pandas. If you want to get started early, check for scipy.stats from the SciPy library.

1.7 Visual Aids for EDA

- As data scientists, two important goals in our work would be **to extract knowledge from the data and to present the data to stakeholders.**
- Presenting results to stakeholders is very complex in the sense that our audience may not have enough technical know-how to understand programming jargon and other technicalities. Hence, visual aids are very useful tools.
- In this chapter, we will focus on different types of visual aids that can be used with our datasets.
- We are going to learn about different types of techniques that can be used in the visualization of data.

In this chapter, we will cover the following topics:

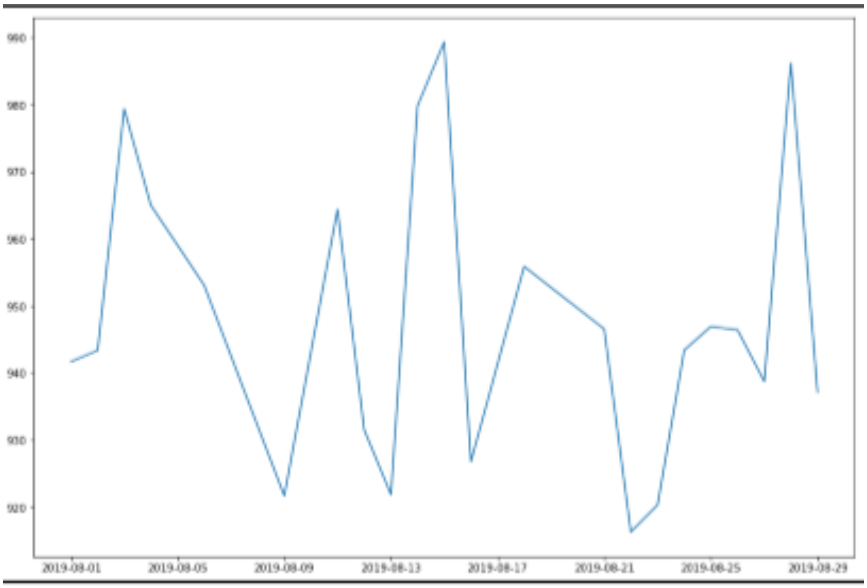
1. Line chart
2. Bar chart
3. Scatter plot
4. Area plot and stacked plot
5. Pie chart Table chart
6. Polar chart
7. Histogram
8. Lollipop chart
9. Choosing the best chart
10. Other libraries to explore

Technical requirements

- **Python libraries such as pandas, seaborn, and matplotlib installed.**

1. Line chart

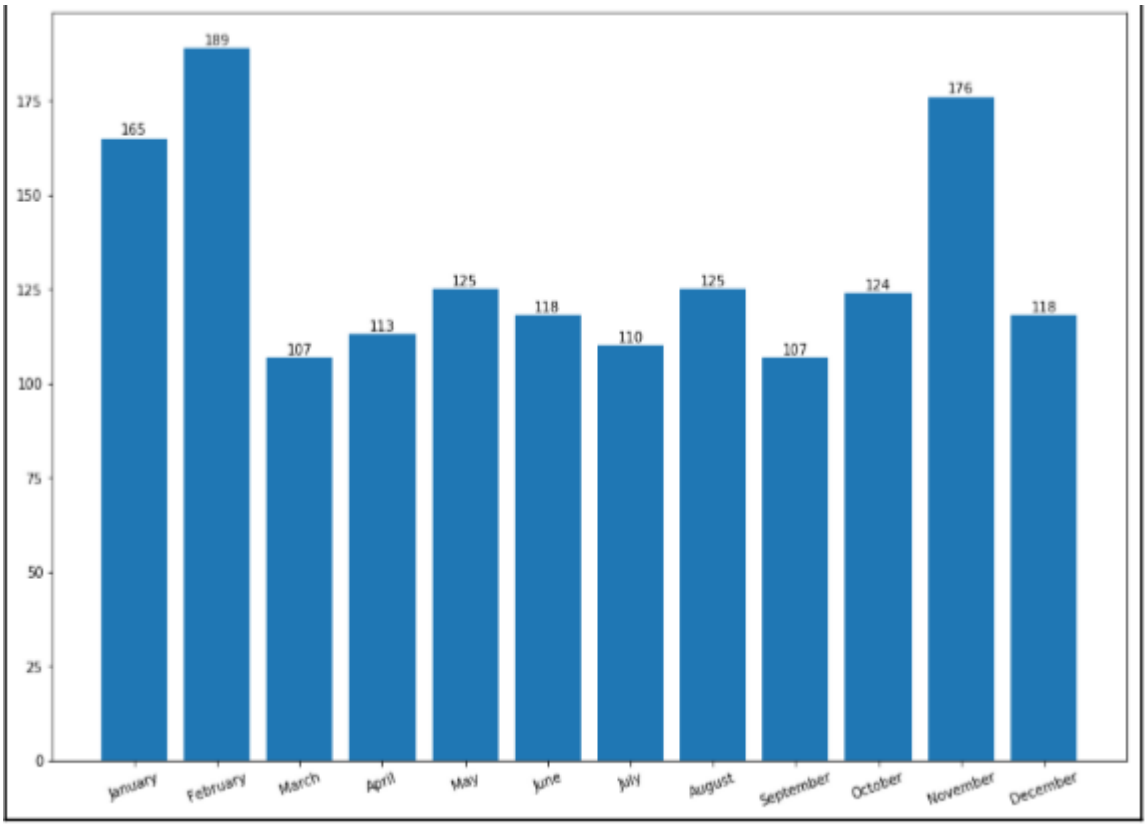
- A **line chart** is used to illustrate the relationship between two or **more continuous variables**.
- We are going to use the matplotlib library and the stock price data to plot time series lines.
- First of all, let's understand the dataset.
- We have created a function using the faker Python library to generate the dataset.
- It is the simplest possible dataset you can imagine, with just two columns. The first column is Date and the second column is Price, indicating the stock price on that date.
- Let's generate the dataset by calling the helper method. In addition to this, we have saved the CSV file. You can optionally load the CSV file using the pandas (read_csv) library and proceed with visualization.

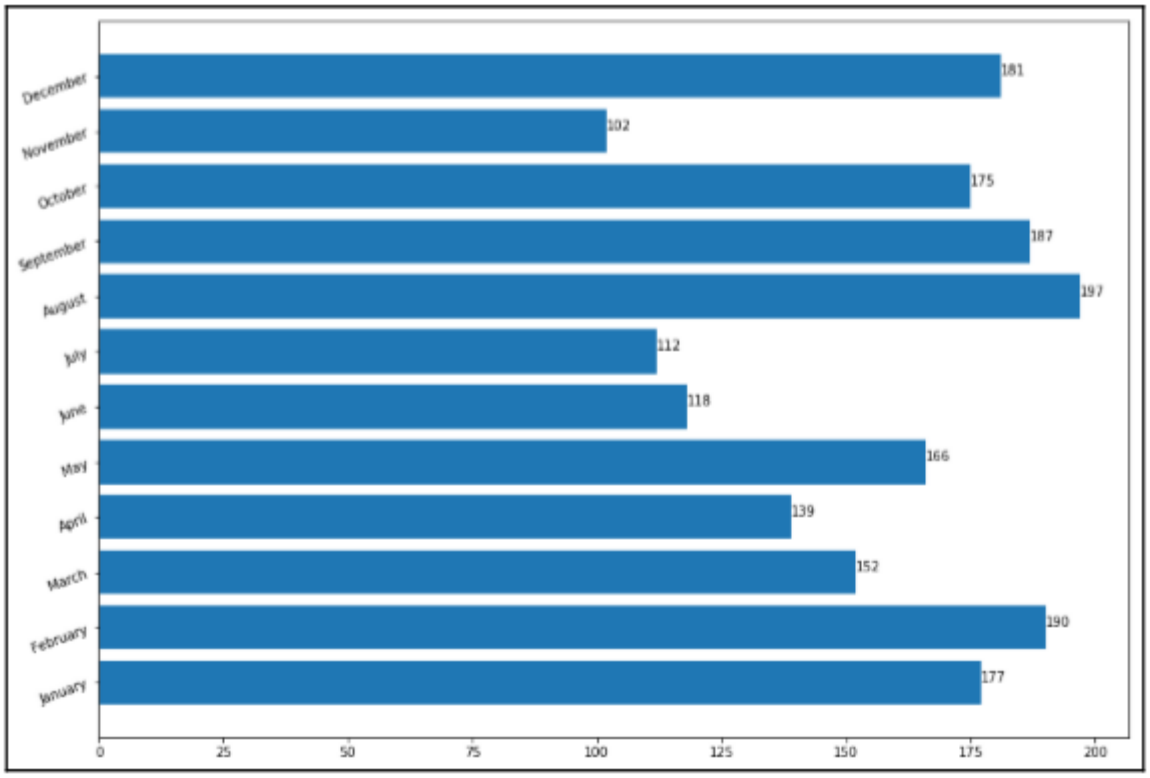


2. Bar charts

- This is one of the most common types of visualization that almost everyone must have encountered. **Bar**s can be drawn **horizontally or vertically** to represent **categorical variables.**
- Bar charts are frequently used to distinguish objects between distinct collections in order to track variations over time.
- In most cases, bar charts are very convenient when the changes are large.

The bar chart is as follows:



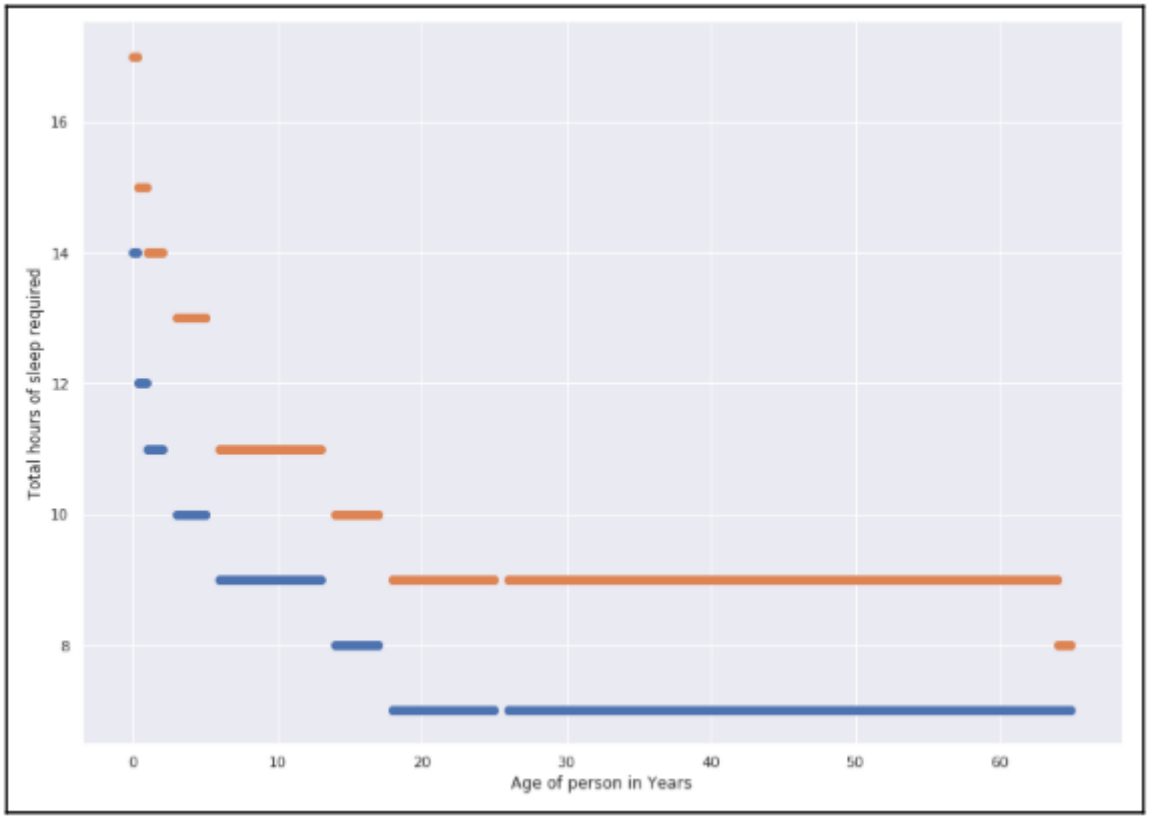


3. Scatter plot

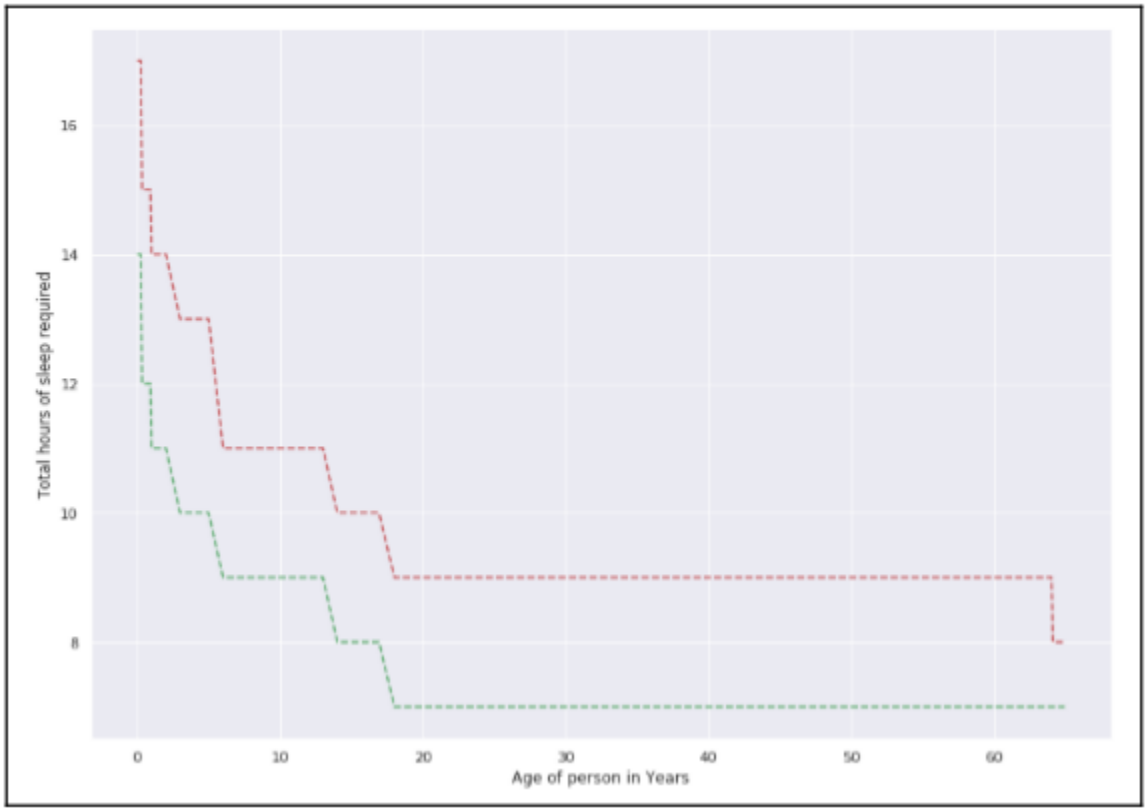
- Scatter plots are also called scatter graphs, scatter charts, scattergrams, and scatter diagrams.
- They use a Cartesian coordinates system to display values of typically two variables for a set of data.
- scatter plots are used when we **need to show the relationship between two variables**, and hence are sometimes referred to as **correlation plots.**
- When should we use a scatter plot? Scatter plots can be constructed in the following two situations:

    1. When one continuous variable is dependent on another variable, which is under the control of the observer
    2. When both continuous variables are independent

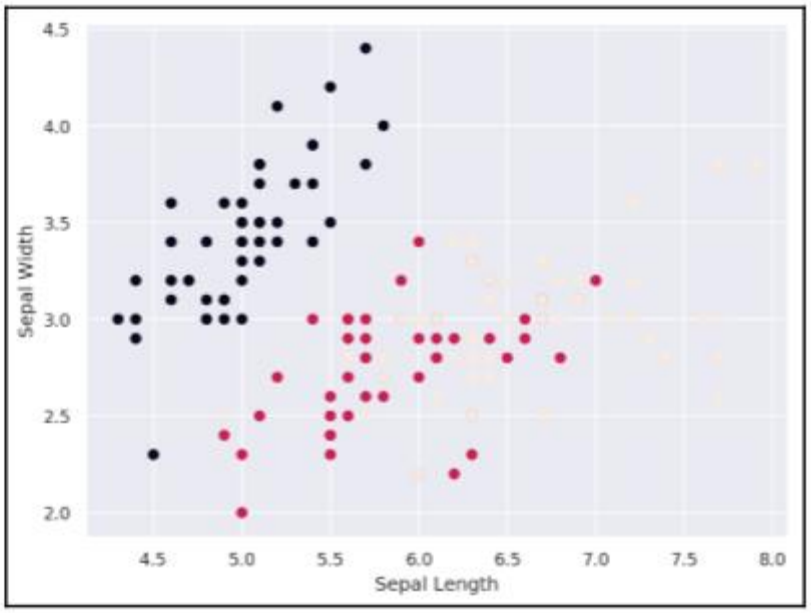
- Some examples in which scatter plots are suitable are as follows:
    1. Research studies have successfully established that the number of hours of sleep required by a person depends on the age of the person.
    2. The average income for adults is based on the number of years of education.



A line chart of the same data is as follows:



The scatter plot generated by the preceding code is as follows:

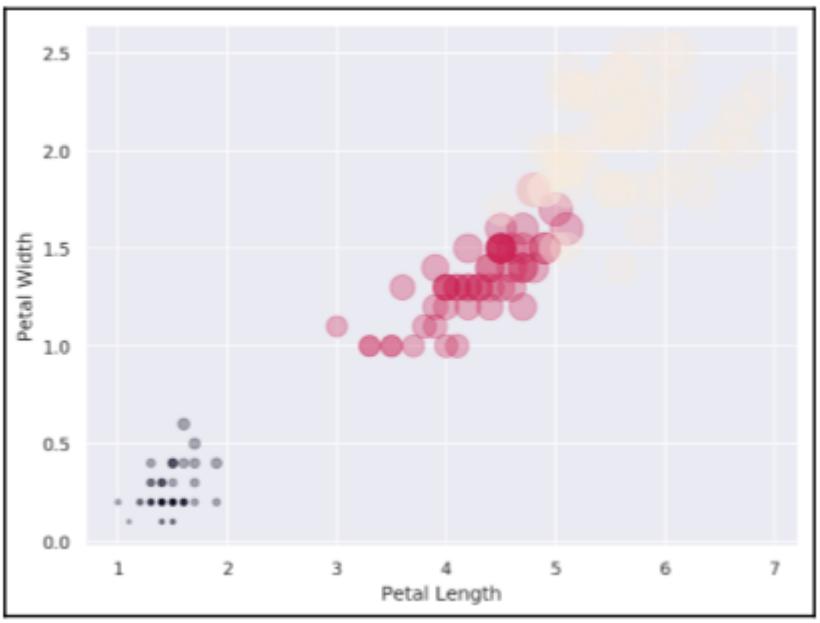


- We can three different types of points and that there are three different clusters.
- However, it is not clear which color represents which species of Iris. Thus, we are going to learn how to create legends in the Scatter plot using seaborn section.

Bubble chart

- A bubble plot is a manifestation of the scatter plot where each data point on the graph is shown as a bubble. Each bubble can be illustrated with a different color, size, and appearance.

The bubble chart generated by the preceding code is as follows:

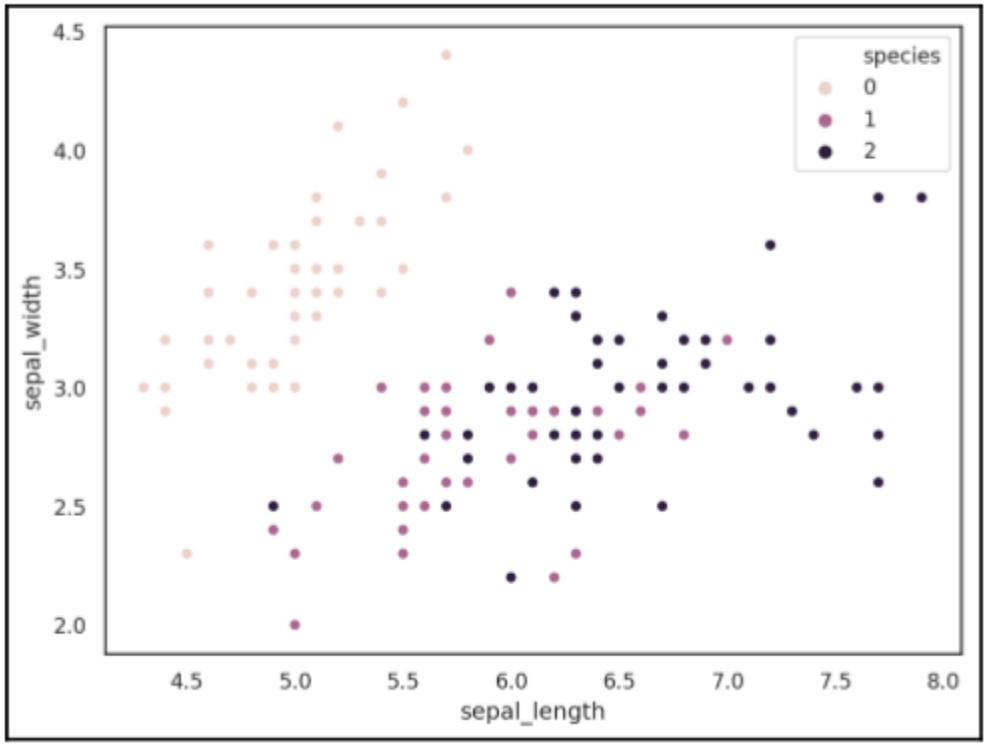


- it is not clear from the graph which color represents which species of Iris. But we can clearly see three different clusters, which clearly indicates for each specific species or cluster there is a relationship between Petal Length and Petal Width.

Scatter plot using seaborn

- A scatter plot can also be generated using the seaborn library. Seaborn makes the graph visually better. We can illustrate the relationship between x and y for distinct subsets of the data by utilizing the size, style, and hue parameters of the scatter plot in seaborn.

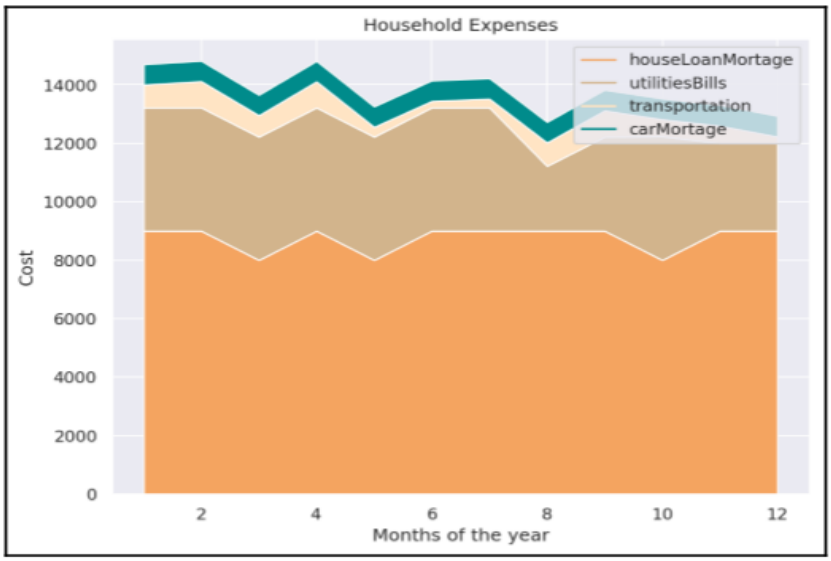
The scatter plot generated from the preceding code is as follows:



- In the preceding plot, we can clearly see there are three species of flowers indicated by three distinct colors. It is more clear from the diagram how different specifies of flowers vary in terms of the sepal width and the length.

4. Area plot and stacked plot

- The stacked plot owes its name to the fact that it represents the area under a line plot and that several such plots can be stacked on top of one another, giving the feeling of a stack.
- The stacked plot can be useful when we want to visualize the **cumulative effect of multiple variables** being plotted on the y axis.



- Now the most important part is the ability to interpret the graph. In the preceding graph, it is clear that the house mortgage loan is the largest expense since the area under the curve for the house mortgage loan is the largest.
- Secondly, the area of utility bills stack covers the second-largest area, and so on.
- The graph clearly disseminates meaningful information to the targeted audience. Labels, legends, and colors are important aspects of creating a meaningful visualization.

5. Pie chart Table chart

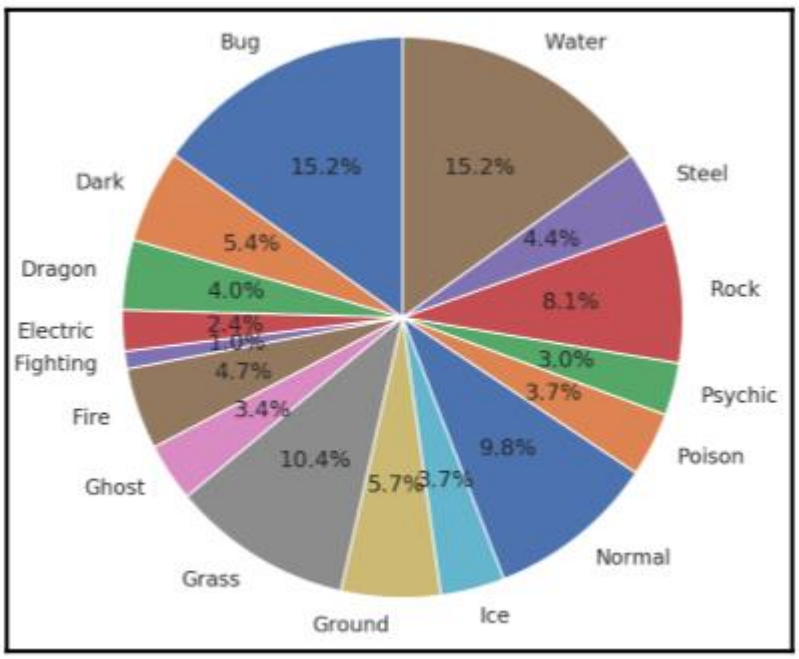
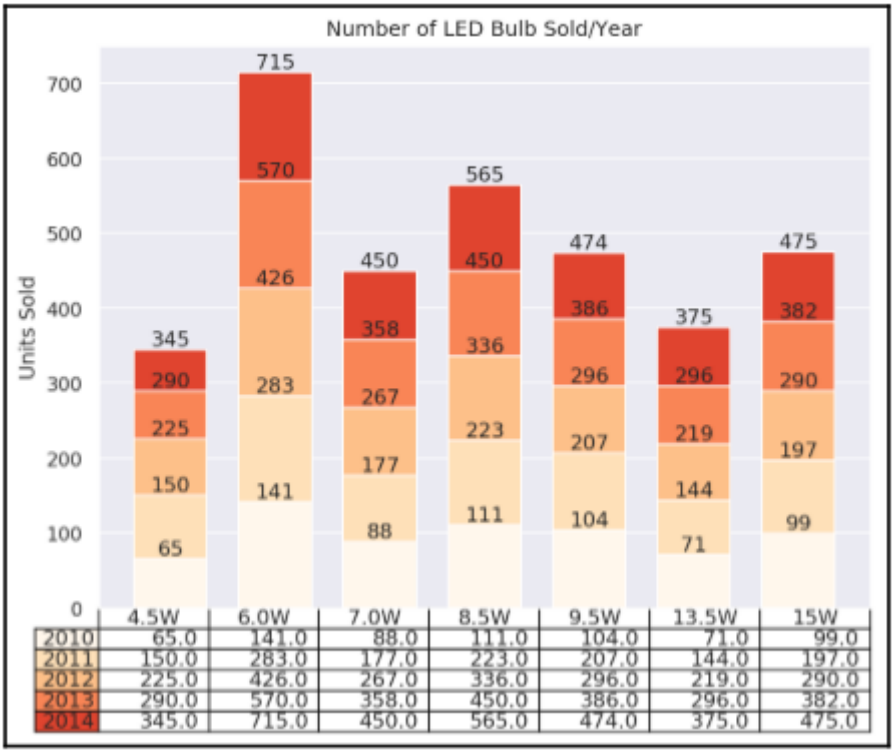


Table Chart

- A table chart combines a bar chart and a table.
- Consider standard LED bulbs that come in different wattages. The standard Philips LED bulb can be 4.5 Watts, 6 Watts, 7 Watts, 8.5 Watts, 9.5 Watts, 13.5 Watts, and 15 Watts. Let's assume there are two categorical variables, the year and the wattage, and a numeric variable, which is the number of units sold in a particular year.

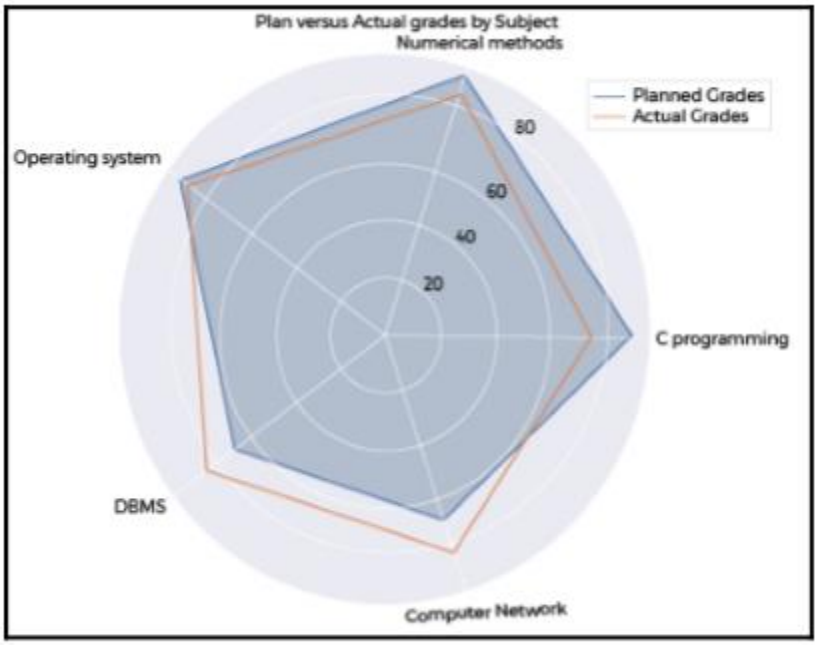


	4.5W	6.0W	7.0W	8.5W	9.5W	13.5W	15W
2010	65.0	141.0	88.0	111.0	104.0	71.0	99.0
2011	150.0	283.0	177.0	223.0	207.0	144.0	197.0
2012	225.0	426.0	267.0	336.0	296.0	219.0	290.0
2013	290.0	570.0	358.0	450.0	386.0	296.0	382.0
2014	345.0	715.0	450.0	565.0	474.0	375.0	475.0

In this chart, in the year 2014, 345 units of the 4.5-Watt bulb were sold. Similarly, the same information can be deduced from the preceding table plot.

Polar chart

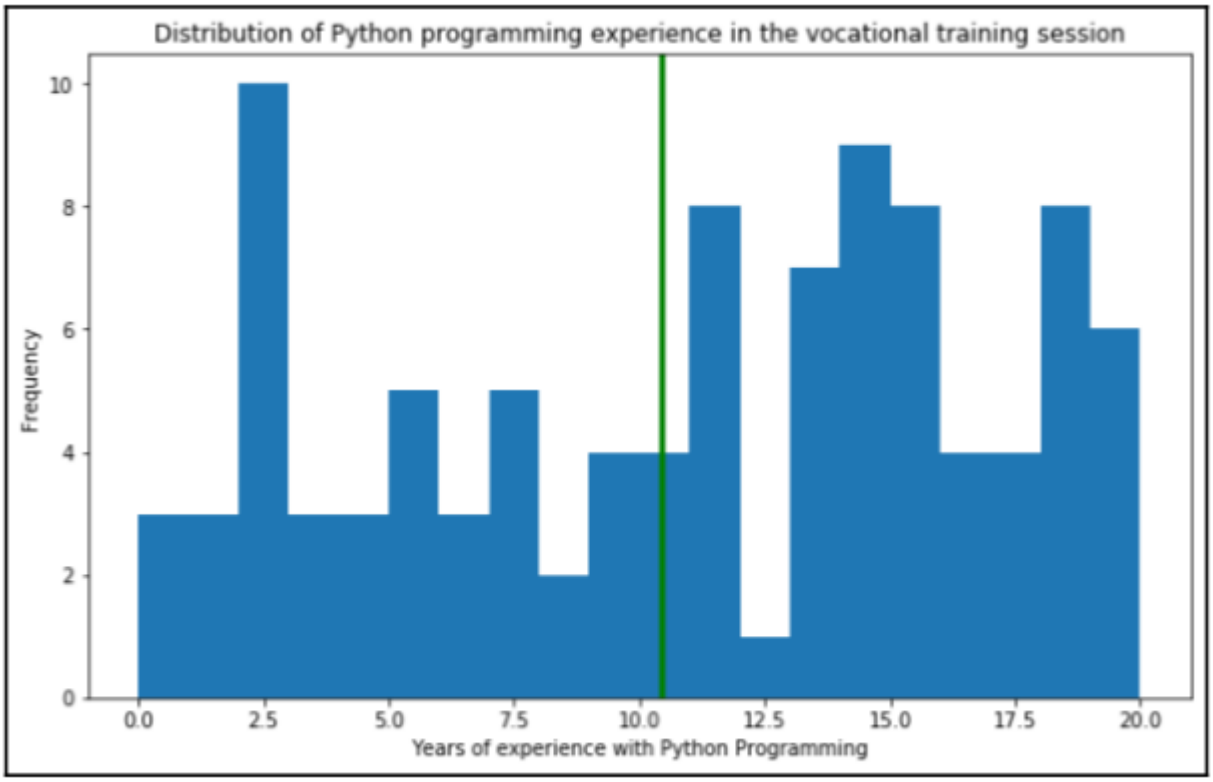
- A polar chart is a diagram that is plotted on a polar axis.
- Its coordinates are angle and radius, as opposed to the Cartesian system of x and y coordinates.
- Sometimes, it is also referred to as a spider web plot.



- The legend makes it clear which line indicates the planned grades (the blue line in the screenshot) and which line indicates actual grades (the orange line in the screenshot). This gives a clear indication of the difference between the predicted and actual grades of a student to the target audience.

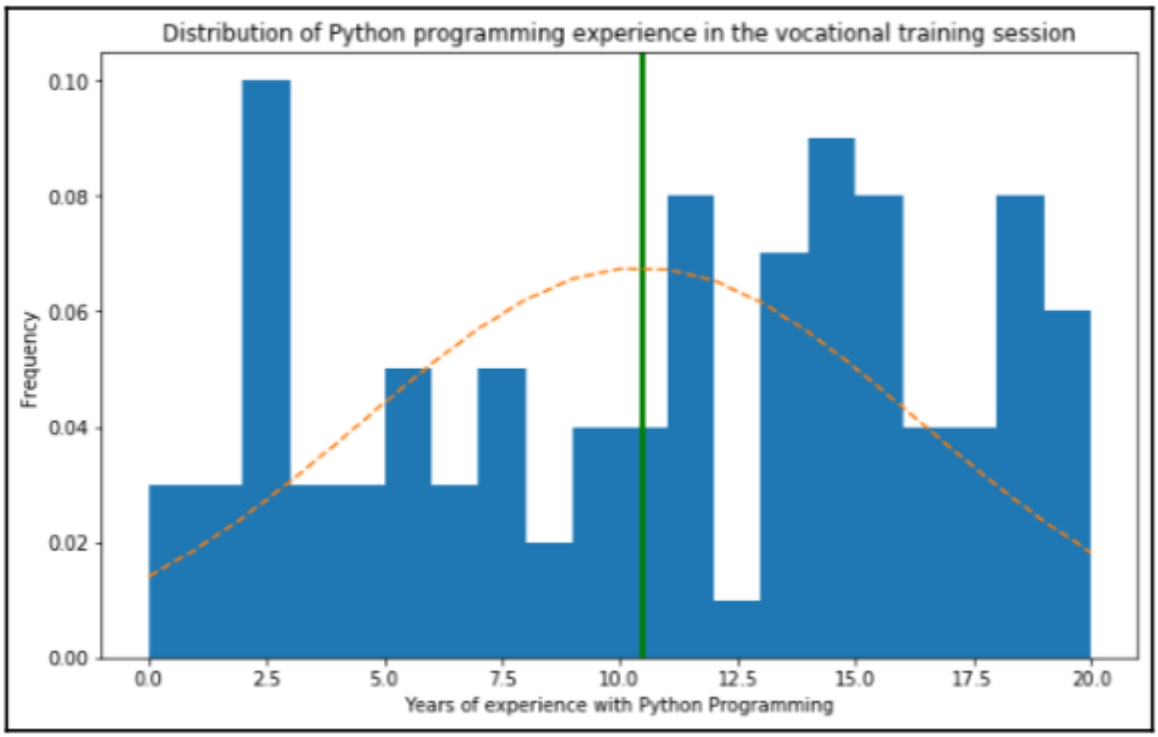
6. Histogram

- Histogram plots are used to depict the distribution of any continuous variable. These types of plots are very popular in statistical analysis.



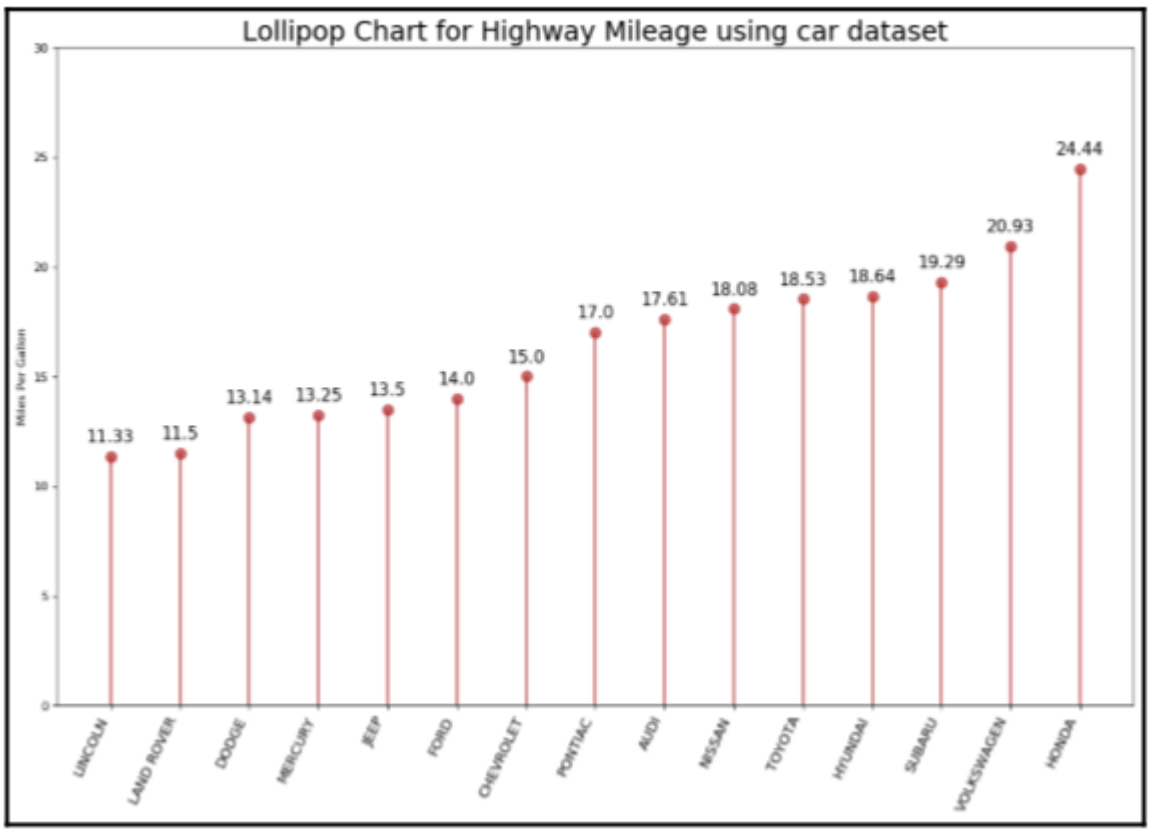
- from the graph, we can say that the average experience of the participants is around 10 years.

And the generated histogram with the normal distribution is as follows:



7. Lollipop chart

- A lollipop chart can be used to display ranking in the data. It is similar to an ordered bar chart.



- The line and the circle on the top gives a nice illustration of different types of cars and their associated miles per gallon consumption

8. Choosing the best chart

- continuous variables – histogram
- show ranking, an ordered - bar chart
- Our purpose should be to illustrate abstract information in a clear way.

The following table shows the different types of charts based on the purposes:

Purpose	Charts
Show correlation	Scatter plot Correlogram Pairwise plot Jittering with strip plot Counts plot Marginal histogram Scatter plot with a line of best fit Bubble plot with circling
Show deviation	Area chart Diverging bars Diverging texts Diverging dot plot Diverging lollipop plot with markers

Show distribution	Histogram for continuous variable Histogram for categorical variable Density plot Categorical plots Density curves with histogram Population pyramid Violin plot Joy plot Distributed dot plot Box plot
Show composition	Waffle chart Pie chart Treemap Bar chart

Show change	Time series plot Time series with peaks and troughs annotated Autocorrelation plot Cross-correlation plot Multiple time series Plotting with different scales using the secondary y axis Stacked area chart Seasonal plot Calendar heat map Area chart unstacked
Show groups	Dendrogram Cluster plot Andrews curve Parallel coordinates
Show ranking	Ordered bar chart Lollipop chart Dot plot Slope plot Dumbbell plot

9. Other libraries to explore

Ploty (https://plot.ly/python/): This is a web-application-based toolkit for visualization. Its API for Jupyter Notebook and other applications makes it very powerful to represent 2D and 3D charts.

Ggplot (http://ggplot.yhathq.com/): This is a Python implementation based on the Grammar of Graphics library from the R programming language.

Altair (https://altair-viz.github.io/): This is built on the top of the powerful Vega-Lite visualization grammar and follows very declarative statistical visualization library techniques. In addition to that, it has a very descriptive and simple API.

1.8 Data transformation Techniques

- Data transformation is a set of techniques used to convert data from one format or structure to another format or structure.
- The following are some examples of transformation activities:
    1. Data deduplication involves the identification of duplicates and their removal. Key restructuring involves transforming any keys with built-in meanings to the generic keys.
    2. Data cleansing involves extracting words and deleting **out-of-date, inaccurate**, and incomplete information from the source language without extracting the meaning or information to enhance the accuracy of the source data.
    3. Data validation is a process of formulating rules or algorithms that help in validating different types of data against some known issues.
    4. Format revisioning involves converting from one format to another.
    5. Data derivation consists of creating a set of rules to generate more information from the data source.
    6. Data aggregation involves searching, extracting, summarizing, and preserving important information in different types of reporting systems.
    7. Data integration involves converting different data types and merging them into a common structure or schema.
    8. Data filtering involves identifying information relevant to any particular user.
    9. Data joining involves establishing a relationship between two or more tables.

- The main reason for transforming the data is **to get a better representation** such that the transformed data is compatible with other data.

1.8.1 Merging Database

Case 1:

StudentID	ScoreSE
1	89
3	39
5	50
7	97
9	20
...	...
...	...
27	73
29	92

StudentID	ScoreSE
2	98
4	93
6	44
8	77
10	69
...	...
...	...
28	56
30	27

We can do that by using the pandas concat() method:

```
dataframe = pd.concat([dataFrame1, dataFrame2], ignore_index=True) dataframe
```

The output of the preceding code is a single dataframe combining both of the tables. These tables would be merged into a single one as shown in the following screenshot:

StudentID	ScoreSE
1	89
3	39
5	50
7	97
9	20
...	...
...	...
27	73
29	92
2	98
4	93
6	44
8	77
10	69
...	...
...	...
28	56
30	27

```
pd.concat([dataFrame1, dataFrame2], axis=1)
```

The output of the preceding code is shown in the following screenshot:

	StudentID	Score	StudentID	Score
0	1	89	2	98
1	3	39	4	93
2	5	50	6	44
3	7	97	8	77
4	9	22	10	69
5	11	66	12	56
6	13	31	14	31
7	15	51	16	53
8	17	71	18	78
9	19	91	20	93
10	21	56	22	56
11	23	32	24	77
12	25	52	26	33
13	27	73	28	56
14	29	92	30	27

Note the difference in the output. When we specify axis=1, the concatenation happens on a side-by-side basis.

Case 2:

Check the following dataframes:

StudentID	ScoreSE
9	22
11	66
13	31
15	51
17	71
...	...
...	...
27	73
29	92

StudentID	ScoreSE
2	98
4	93
6	44
8	77
10	69
...	...
...	...
28	56
30	27

StudentID	ScoreML
1	39
3	49
5	55
7	77
9	52
...	...
...	...
27	23
29	49

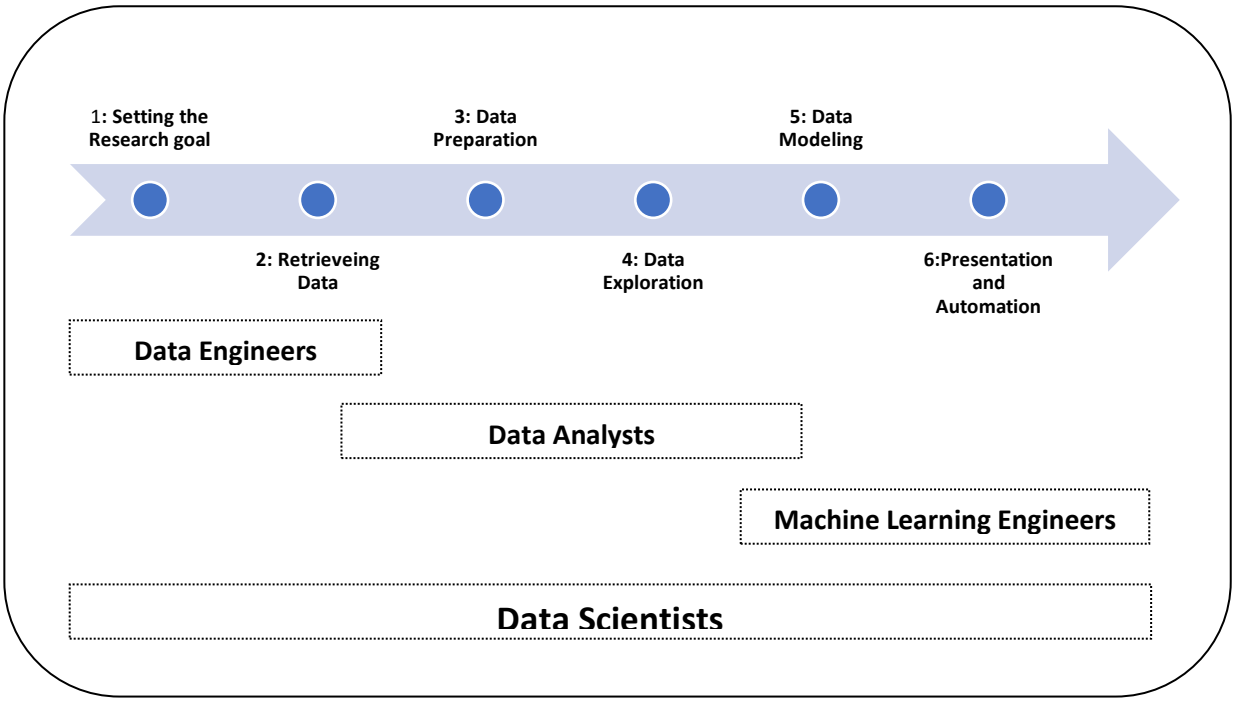
StudentID	ScoreML
2	98
4	93
6	44
8	77
10	69
...	...
...	...
28	56
30	27

1.8.2 Reshaping and Pivoting

1.8.3 Transformation Techniques

1.9 Data Science Process

- The data science process typically consists of six steps.
  i. Setting the research goal
  ii. Retrieving Data
  iii. Data Preparation
  iv. Data Exploration
  v. Data modeling or Model Building
  vi. Presentation and Automation



Types of Data Analysis

- **Descriptive Analysis:** It tries to understand **what happened** in the past by analysing the stored data.

• **Diagnostic Analysis:** It focuses on understanding **why something** has happened. It is literally the diagnosis of a problem
• **Predictive Analysis:** It tries to understand **what could happen** in the future using past data analysis.
• **Prescriptive Analysis:** It allows you to make **recommendations for the future.** This is the final step in the analytics part of the process.

1.3.1 Setting the research goal

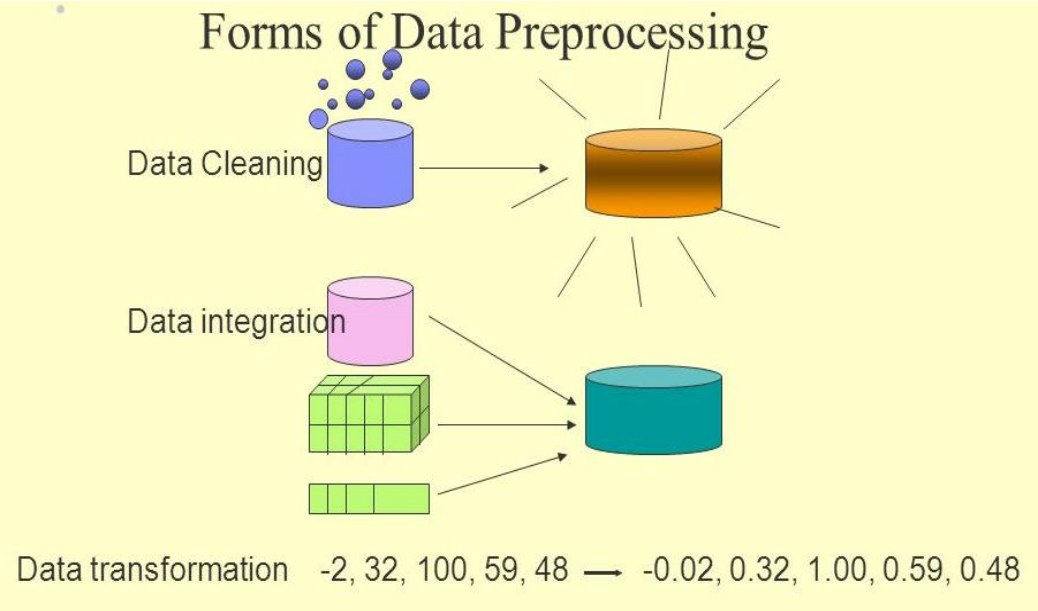
- Data science is mostly applied in the context of an organization.
- When the business to perform a data science project, first prepare a project charter.
- It contains information such as what to research, how the company benefits from that, what data and resources to need, a timetable, and deliverables.

1.3.2 Retrieving Data

- The second step is to collect data.
- In this step, which data to need and where to find it and to ensure that checking the existence of, quality, and access to the data.
- Data can also be delivered by third-party companies and takes many forms ranging from Excel spreadsheets to different types of databases.

1.3.3. Data Preparation

- Data collection is an error-prone process. In this phase, have to enhance the quality of the data and prepare it for use in subsequent steps. \
- This phase consists of three subphases:
  i. **Data Cleansing** removes false values from a data source and inconsistencies across data sources.
  ii. **Data Integration** enriches data sources by combining information from multiple data sources.
  iii. **Data Transformation** ensures that the data is in a suitable format to use in models.



1.3.4. Data Exploration

- It is also known as Exploratory Data Analysis (EDA).
- Data exploration is concerned with building a deeper understanding of the data.

- To understand how variables interact with each other, the distribution of the data, and whether there are outliers.
- To achieve this, use descriptive statistics, visual techniques, and simple modeling.

1.3.5. Data modeling or Model Building

- To use the models, domain knowledge, and insights about the data found in the previous steps to answer the research question.
- To select a technique from the fields of statistics, machine learning, operations research, and so on.
- Building a model is an iterative process that involves selecting the variables for the model, executing the model, and model diagnostics.

1.3.6. Presentation and Automation

- Finally present the results to the business.
- These results can take many forms, ranging from presentations to research reports.
- Sometimes need to automate the execution of the process because the business will want to use the insights gained in another project or enable an operational process to use the outcome from the model.

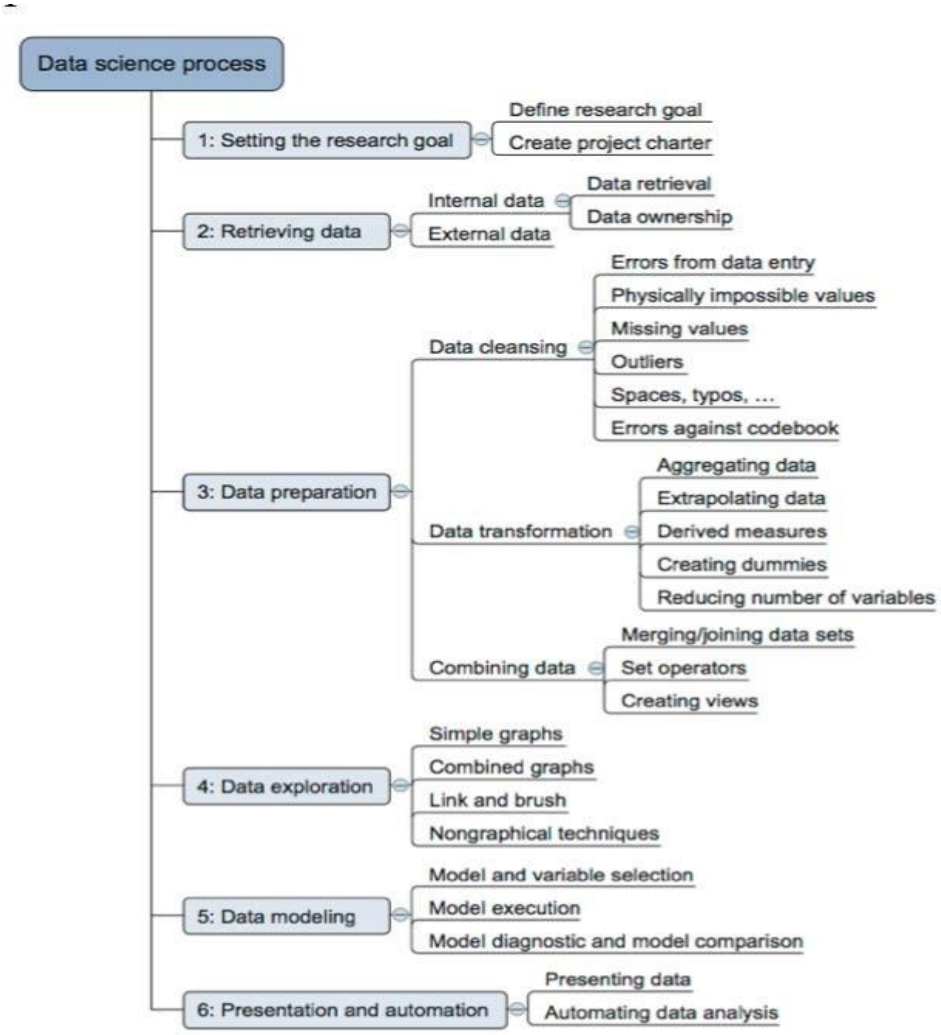
1.10 Overview of the data science process

Summary of the data science process

1. The first step of this process is setting a research goal. The main purpose here is making sure all the stakeholders understand the what, how, and why of the project. In every serious project this will result in a project charter.
2. The second phase is data retrieval. If data available for analysis then finding the suitable data and getting access to the data from the data owner. The result is data in its raw form, which probably needs polishing and transformation before it becomes usable.
3. Now the data are the raw data, it's time to prepare it. This includes transforming the data from a raw form into data that's directly usable in the models. To achieve this, to detect and correct different kinds of errors in the data, combine data from different data sources, and transform it. Once successfully completed, the data can visualize and modeling.
4. The fourth step is data exploration. The goal of this step is to gain a deep understanding of the data. Look for patterns, correlations, and deviations

based on visual and descriptive techniques. Gain from this phase will enable to start modeling.

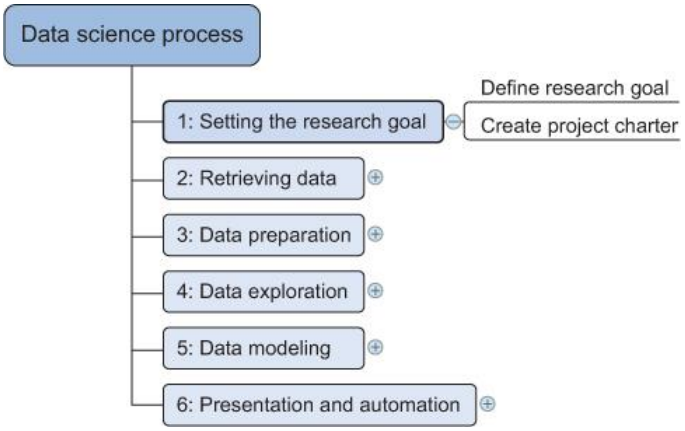
5. Finally the most important part: model building or data modeling to be done. It is now that to gain the insights or make the predictions stated in the project charter. Now is the time to bring out the heavy guns, but remember research has taught us that often (but not always) a combination of simple models tends to outperform one complicated model. If this phase is completed correctly, it is almost done.
6. The last step of the data science model is presenting the results and automating the analysis, if needed. One goal of a project is to change a process and/or make better decisions. The importance of this step is more apparent in projects on a strategic and tactical level. Certain projects require to perform. The business process over and over again, so automating the project will save time.



The six steps of the data science process

1.11 Defining research goals

- A project starts by understanding the what, the why, and the how the project.
  i. What does the company expect you to do?
  ii. Why does management place such a value on your research?
  iii. How to complete the research?
- Answering these three questions (what, why, how) is the goal of the first phase, so that everybody knows what to do and can agree on the best course of action.



Step 1: Setting the research goal

- The outcome should be a clear research goal, a good understanding of the context, well-defined deliverables, and a plan of action with a timetable.
- This information is then best placed in a project charter. The length and formality can, of course, differ between projects and companies.
- In this early phase of the project, people skills and business acumen are more important than great technical prowess, which is why this part will often be guided by more senior personnel.

1.5.1 Spend time understanding the goals and context of the research

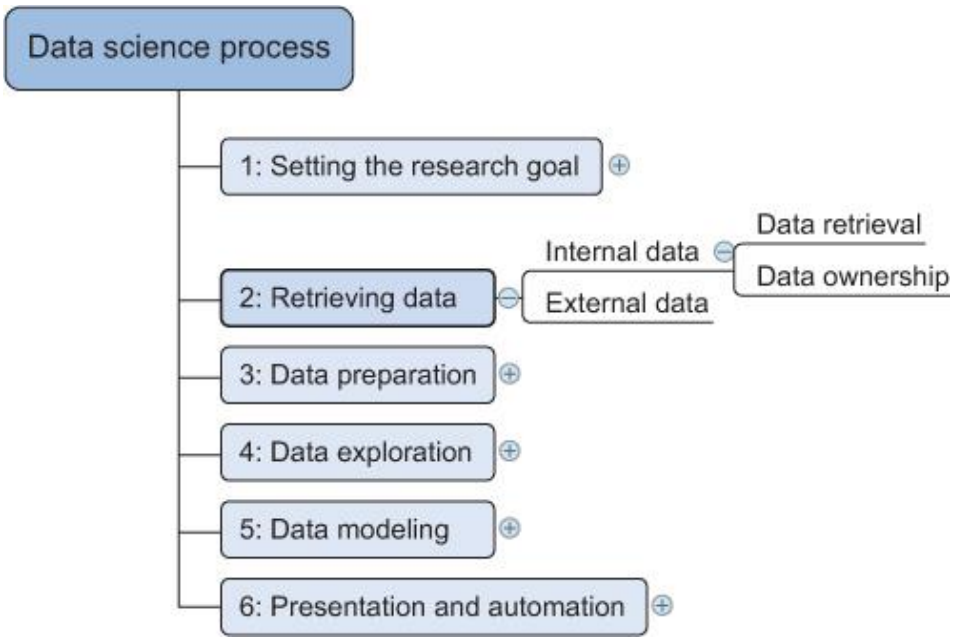
- The research goal that states the purpose of the assignment in a clear and focused manner
- Understanding the business goals and context is critical for project success.
- Continue asking questions and devising examples until to grasp the exact business expectations, identify how the project fits in the bigger picture, appreciate how to research is going to change the business, and understand how to use the results.

1.5.2 Create a project charter

- A project charter requires teamwork, and the input covers at least the following:
  - A clear research goal
  - The project mission and context
  - How you're going to perform your analysis
  - What resources you expect to use
  - Proof that it's an achievable project, or proof of concepts
  - Deliverables and a measure of success
  - A timeline
- The client can use this information to make an estimation of the project costs and the data and people required for the project to become a success.

1.12 Retrieving data

- In this step, need to retrieve the required data. Sometimes need to go to the field and design a data collection process.
- But most of the time, many companies will have already collected and stored the data. If not, even high-quality data freely available for public and commercial use.



Step 2: Retrieving data

- Data can be stored in many forms, ranging from simple text files to tables in a database. Data is often like a diamond in the rough: it needs polishing to be useful.

1.6.1 Start with data stored within the company

- This data can be stored in official data repositories such as databases, data marts, data warehouses, and data lakes maintained by a team of IT professionals.
- The primary goal of a database is data storage, while a data warehouse is designed for reading and analyzing that data.
- A data mart is a subset of the data warehouse and geared toward serving a specific business unit.
- While data warehouses and data marts are home to preprocessed data, data lakes contain data in its natural or raw format. But the possibility exists that the data still resides in Excel files on the desktop of a domain expert.
- Finding data even within the own company can sometimes be a challenge. As companies grow, their data becomes scattered around many places.
- Knowledge of the data may be dispersed as people change positions and leave the company. Documentation and metadata aren't always the top priority of a delivery manager.
- Getting access to data is another difficult task. Organizations understand the value and sensitivity of data and often have policies in place so everyone has access to what they need and nothing more

1.6.2 Don't be afraid to shop around

- If data isn't available inside the organization, look outside the organization's walls. Many companies specialize in collecting valuable information.

Open data site	Description
Data.gov	The home of the US Government's open data
https://open-data.europa.eu/	The home of the European Commission's open data
Freebase.org	An open database that retrieves its information from sites like Wikipedia, MusicBrains, and the SEC archive
Data.worldbank.org	Open data initiative from the World Bank
Aiddata.org	Open data for international development
Open.fda.gov	Open data from the US Food and Drug Administration

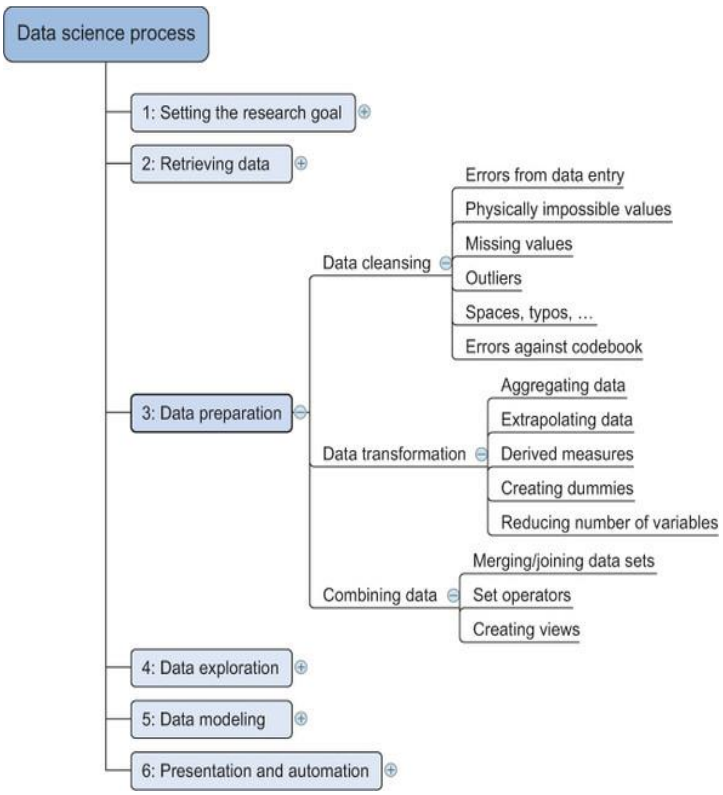
Table : A list of open-data provides that should get

1.6.3 Do data quality checks now to prevent problems later

- Doing data correction and cleansing is very important one. Investigate the data during the import, data preparation, and exploratory phases.
- During data retrieval, have to check if the data is equal to the data in the source document and have the right data types.
- During data preparation, have to do a more elaborate check. The focus is on the content of the variables. To get rid of typos and other data entry errors and bring the data to a common standard among the data sets. For example, you might correct USQ to USA and United Kingdom to UK.
- During the exploratory phase, learn something from the data. Now the data to be clean and look at the statistical properties such as distributions, correlations, and outliers.
- For instance, when you discover outliers in the exploratory phase, they can point to a data entry error.

1.13 Data preparation

- It is nothing but cleansing, integrating, and transforming data.
- The data received from the data retrieval phase is likely to be "a diamond in the rough." Now it is time to sanitize and prepare it for use in the modeling and reporting phase.
- Doing so is tremendously important because the models will perform better and will lose less time trying to fix strange output. It can't be mentioned nearly enough times: garbage in equals garbage out. The model needs the data in a specific format, so data transformation will always come into play.



Step 3: Data preparation

- It's a good habit to correct data errors as early on in the process as possible. However, this isn't always possible in a realistic setting, so you'll need to take corrective actions in your program.

1.7.1 Cleansing data

- Data cleansing is a sub process of the data science process that focuses on removing errors and data becomes a true and consistent representation of the processes it originates from.

- By “true and consistent representation” to imply that at least two types of errors exist.
- The first type is the *interpretation error*: such as when you take the value in your data for granted, like saying that a person’s age is greater than 300 years.
- The second type of error points to *inconsistencies* between data sources or against the company’s standardized values
- An example of this class of errors is putting “Female” in one table and “F” in another when they represent the same thing: that the person is female. Another example is that you use Pounds in one table and Dollars in another.

1.7.2 Correct errors as early as possible

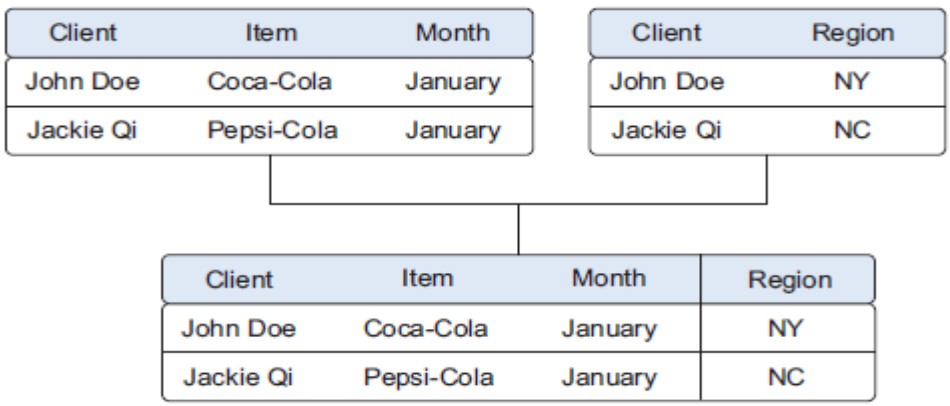
- A good practice is to mediate data errors as early as possible in the data collection chain and to fix as little as possible inside your program while fixing the origin of the problem. Retrieving data is a difficult task, and organizations spend millions of dollars on it in the hope of making better decisions. The data collection process is error prone, and in a big organization it involves many steps and teams.

1.7.3 Combining data from different data sources

- Your data comes from several different places, and in this substep we focus on integrating these different sources. Data varies in size, type, and structure, ranging from databases and Excel files to text documents.

The Different Ways of Combining Data

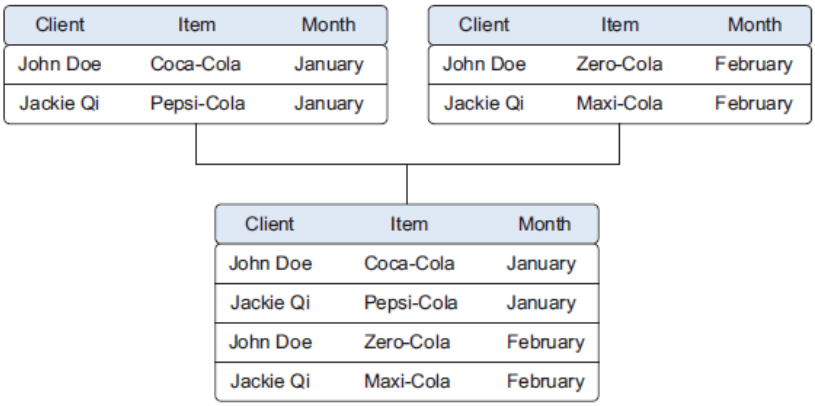
i. Joining Tables:
   - Joining tables allows you to combine the information of one observation found in one table with the information that you find in another table. The focus is on enriching a single observation.



Joining two tables on the Item and Region keys

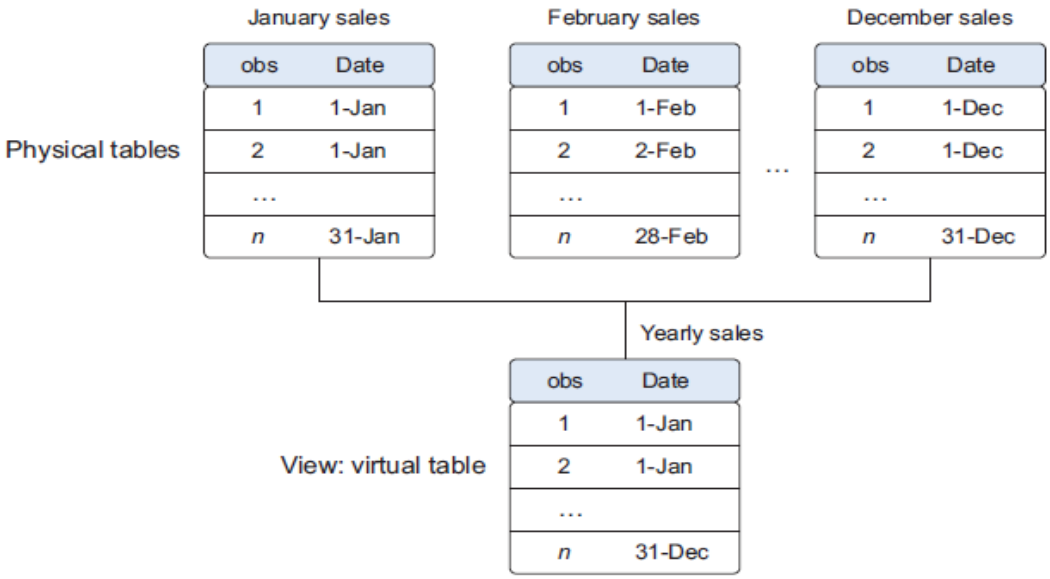
ii. Appending Tables:

- Appending or stacking tables is effectively adding observations from one table to another table.



Appending data from tables is a common operation but requires an equal structure in the tables being appended.

iii. Using Views To Simulate Data Joins And Appends:



A view helps you combine data without replication

1.7.3 Transforming data

- The data so it takes a suitable form for data modeling.
- Relationships between an input variable and an output variable aren't always linear.

Customer	Year	Gender	Sales
1	2015	F	10
2	2015	M	8
1	2016	F	11
3	2016	M	12
4	2017	F	14
3	2017	M	13

M F

Customer	Year	Sales	Male	Female
1	2015	10	0	1
1	2016	11	0	1
2	2015	8	1	0
3	2016	12	1	0
3	2017	13	1	0
4	2017	14	0	1

Turning variables into dummies is a data transformation that breaks a variable that has multiple classes into multiple variables, each having only two possible values: 0 or 1.

1.14 Exploratory Data analysis

- During exploratory data analysis you take a deep dive into the data (see figure 2.14). Information becomes much easier to grasp when shown in a picture, therefore you mainly use graphical techniques to gain an understanding of your data and the interactions between variables. This phase is about exploring data, so keeping your mind open and your eyes peeled is essential during the exploratory data analysis phase. The goal isn't to cleanse the data, but it's common that you'll still discover anomalies you missed before, forcing you to take a step back and fix them.

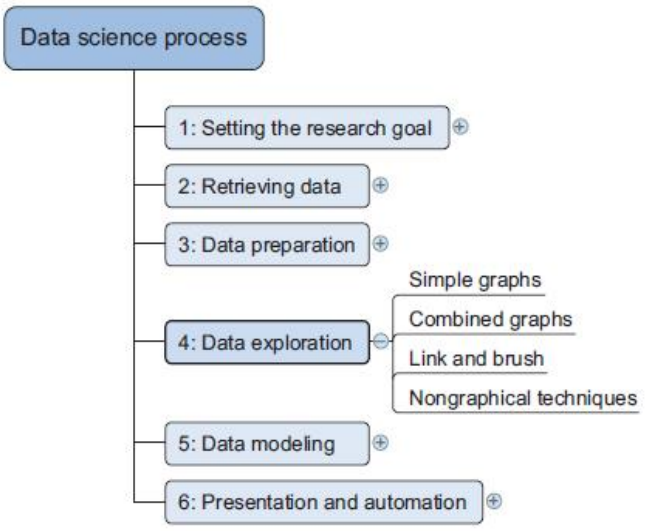


Figure 2.14 Step 4: Data exploration

- The visualization techniques you use in this phase range from simple line graphs or histograms, as shown in figure 2.15, to more complex diagrams such as Sankey and network graphs. Sometimes it's useful to compose a composite graph from simple graphs to get even more insight into the data. Other times the graphs can be animated or made interactive to make it easier and, let's admit it, way more fun. An example of an interactive Sankey diagram can be found at http://bost.ocks.org/ mike/sankey/.
- Mike Bostock has interactive examples of almost any type of graph. It's worth spending time on his website, though most of his examples are more useful for data presentation than data exploration.

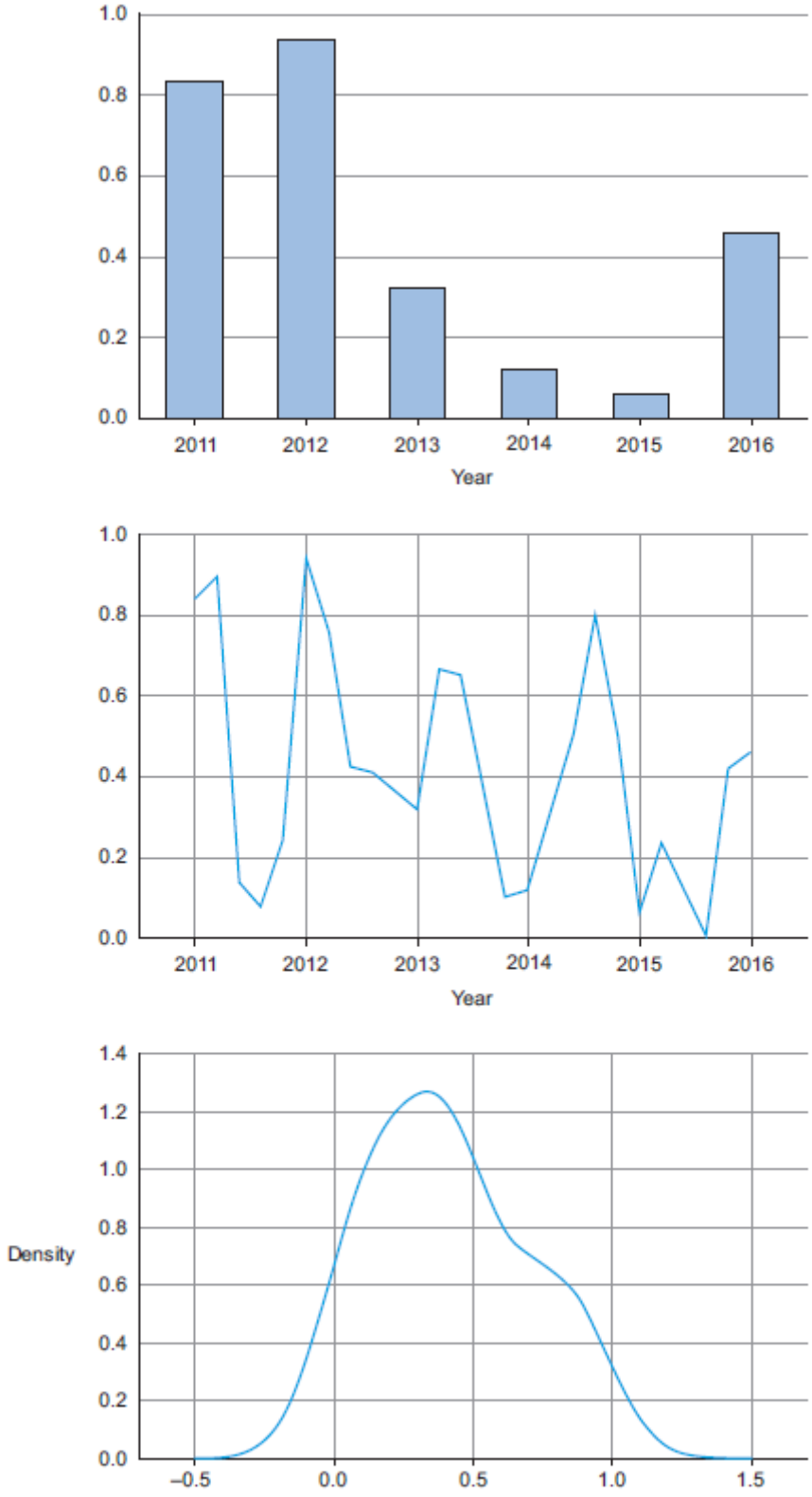
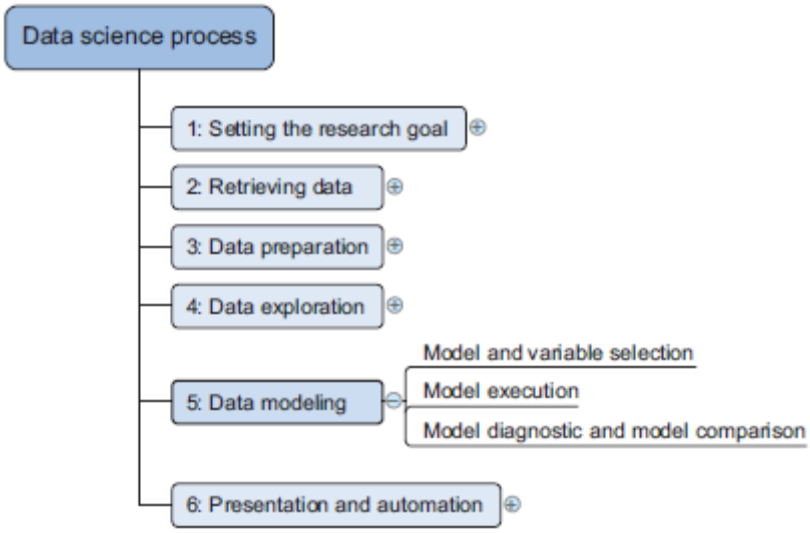


Figure 2.15 From top to bottom, a bar chart, a line plot, and a distribution are some of the graphs used in exploratory analysis.

1.15 Build the model

❖ With clean data in place and a good understanding of the content, you're ready to build models with the goal of making better predictions, classifying objects, or gaining an understanding of the system that you're modeling. This phase is much more focused than the exploratory analysis step, because you know what you're looking for and what you want the outcome to be. Figure shows the components of model building.



Data modeling

Building a model is an iterative process.

❖ In this phase, data science team needs to develop datasets for training, testing and production purposes.

❖ **Building a model** requires splitting of data into two sets, such as **_training set'** and **_testing set'** in the ratio of **80:20 or 70:30.**

❖ A set of supervised **(for labeled data)** and unsupervised **(for unlabeled data)** algorithms are available to choose from depending on the nature of input data and business outcome to predict.

❖ **These data sets enable data scientist to develop analytical method and train it, while holding aside some of data for testing the model.**

❖ The goal of making better predictions, classifying objects or gaining an understanding of the system need to build the models. This phase is much more focused than the exploratory analysis step.

Exploratory data analysis involves data attributes identification, data preprocessing and feature engineering.

- **A data attribute** is a single-value descriptor for a data point or data object.
- **Data preprocessing** involves identification of missing values, outliers that fill gaps by computing mean or median for quantitative attributes and

mode for qualitative attributes of data to improve the predictive power of model.

- **Feature engineering** improves the performance of the model by selecting the right features and preparing the features in a way that is suitable for the model.

- The techniques are borrowed from the field of machine learning, data mining and/or statistics.

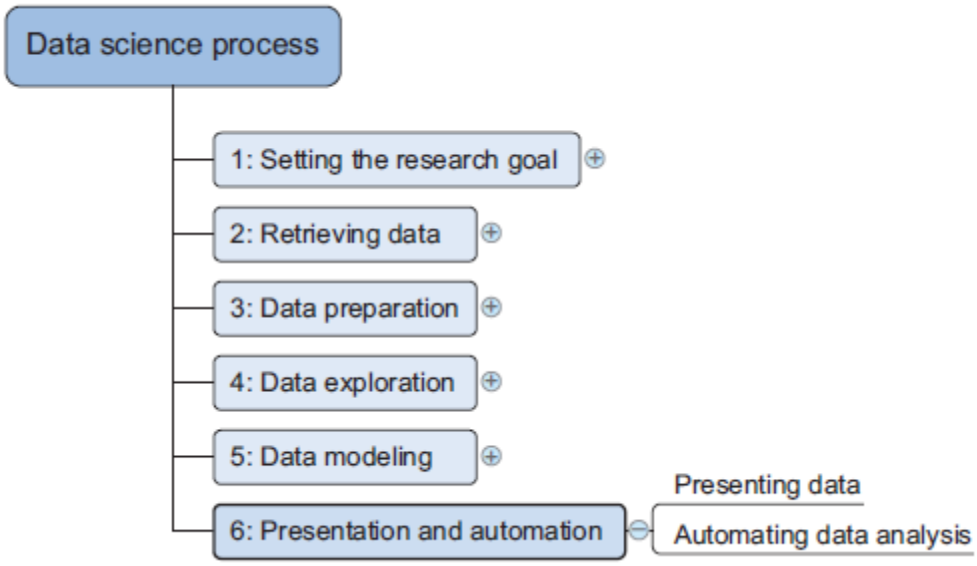
- The definition of good model includes **robustness and well-defined accuracy.**

Models consist of the following main steps:

- 1. Selection of a modeling technique and variables to enter the model
- 2. Execution of the model
- 3. Diagnosis and model comparison

1.16 Presenting findings and building applications

❖ After you've successfully analyzed the data and built a well-performing model, you're ready to present your findings to the world. This is an exciting part; all your hours of hard work have paid off and you can explain what you found to the stakeholders.



Presentation and automation

1.17 Data Mining

❖ Data mining is the process of discovering interesting patterns and knowledge from large amounts of data.

- The data sources can include databases, data warehouses, the Web, other information repositories, or data that are streamed into the system dynamically.
- Extracting the hidden information from the data.
- In addition, many other terms have a similar meaning to data mining—for example, knowledge mining from data, knowledge extraction, data/pattern analysis, data archaeology, and data dredging
- Knowledge discovery from data, or KDD- The knowledge discovery process

    1. Data cleaning (to remove noise and inconsistent data)

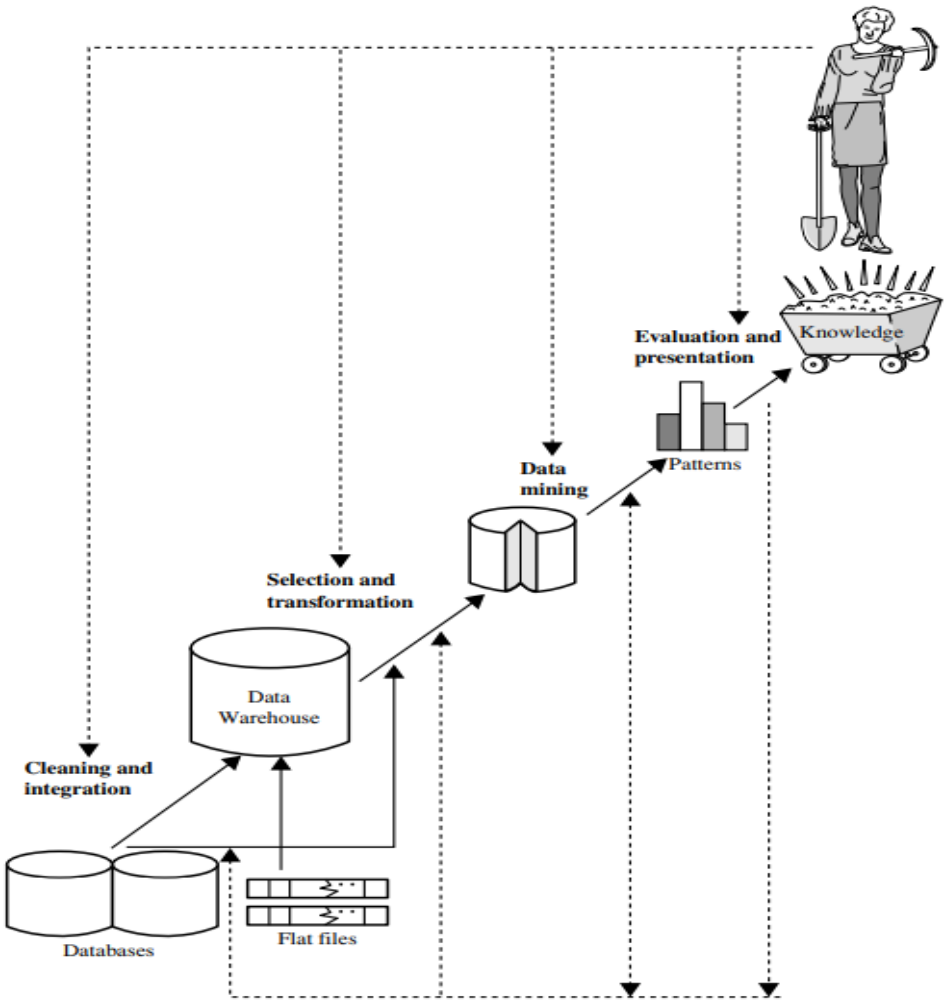
    2. Data integration (where multiple data sources may be combined)

    3. Data selection (where data relevant to the analysis task are retrieved from the database)

    4. Data transformation (where data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations)

    5. Data mining (an essential process where intelligent methods are applied to extract data patterns) 6. Pattern evaluation (to identify the truly interesting patterns representing knowledge based on interestingness measures)

    7. Knowledge presentation (where visualization and knowledge representation techniques are used to present mined knowledge to users)



Data mining as a step in the process of knowledge discovery.

1.18 Data Warehousing

- "A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision making process"
  - **Subject-oriented:** A data warehouse is organized around major subjects such as customer, supplier, product, and sales. Rather than concentrating on the day-to-day operations and transaction processing of an organization, a data warehouse focuses on the modeling and analysis of data for decision makers. Hence, data warehouses typically provide a simple and concise view of particular subject issues by excluding data that are not useful in the decision support process.
  - **Integrated:** A data warehouse is usually constructed by integrating multiple heterogeneous sources, such as relational databases, flat files, and online transaction records. Data cleaning and data integration techniques are applied to ensure consistency in naming conventions, encoding structures, attribute measures, and so on.

- **Time-variant:** Data are stored to provide information from an historic perspective (e.g., the past 5–10 years). Every key structure in the data warehouse contains, either implicitly or explicitly, a time element.
- **Nonvolatile:** A data warehouse is always a physically separate store of data transformed from the application data found in the operational environment. Due to this separation, a data warehouse does not require transaction processing, recovery, and concurrency control mechanisms. It usually requires only two operations in data accessing: initial loading of data and access of data.