

# Introduction to Python for Machine Learning

Madushanka Wadumulla  
PGC in IA  
Beng, IIESL, AMEI



# Outline

- Types of Matrices
- Transpose
- Determinant
- Inverse

# Python Programming Language

- High-level programming language.
- Free download software.
- Case sensitive
- Complete source code available.
- Environment for data analysis and graphics.
- Used to create web applications.
- Comprehensive platform, offering all manner of machine learning techniques.
- Runs on a wide array of platforms, including Windows, Linux, and Mac OS X.
- Easily import data from a wide variety of sources, including text files, database management systems, statistical packages, and specialized data repositories.

# Anaconda

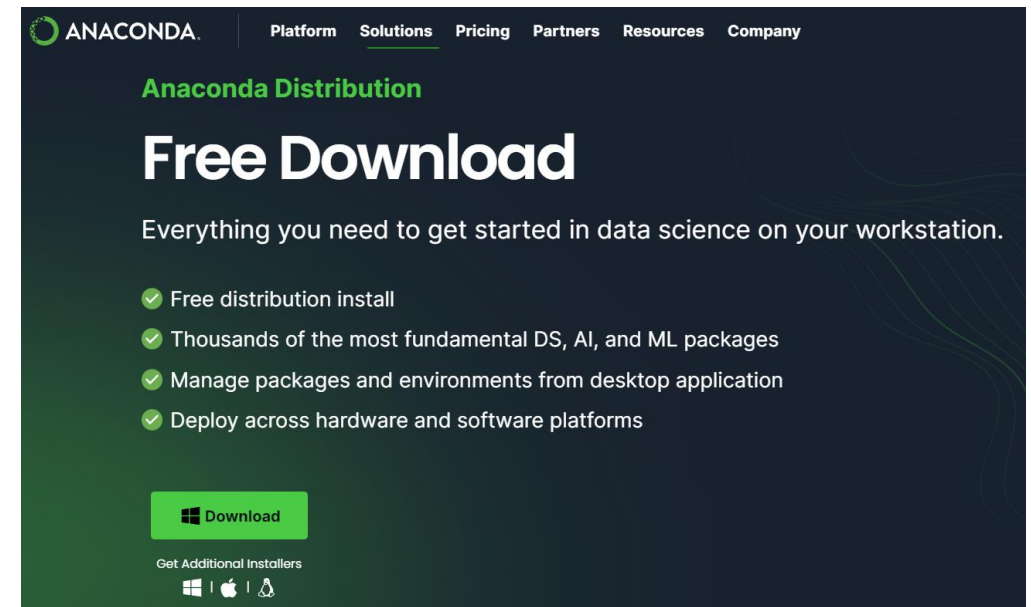
- An open-source distribution for Python and R, particularly popular for data analysis and scientific computing.
- Used for data science, machine learning and deep learning.
- Available for Windows, Linux, and Mac OS X.
- Includes many popular packages such as NumPy, SciPy, Pandas, Matplotlib, Seaborn, etc.
- Develop data science projects using our IDEs, including Jupyter Notebook, Spyder, Jupyter lab and RStudio.

# Install Python

Download the latest version of Python 3.12.2. from  
<https://www.python.org/downloads/>

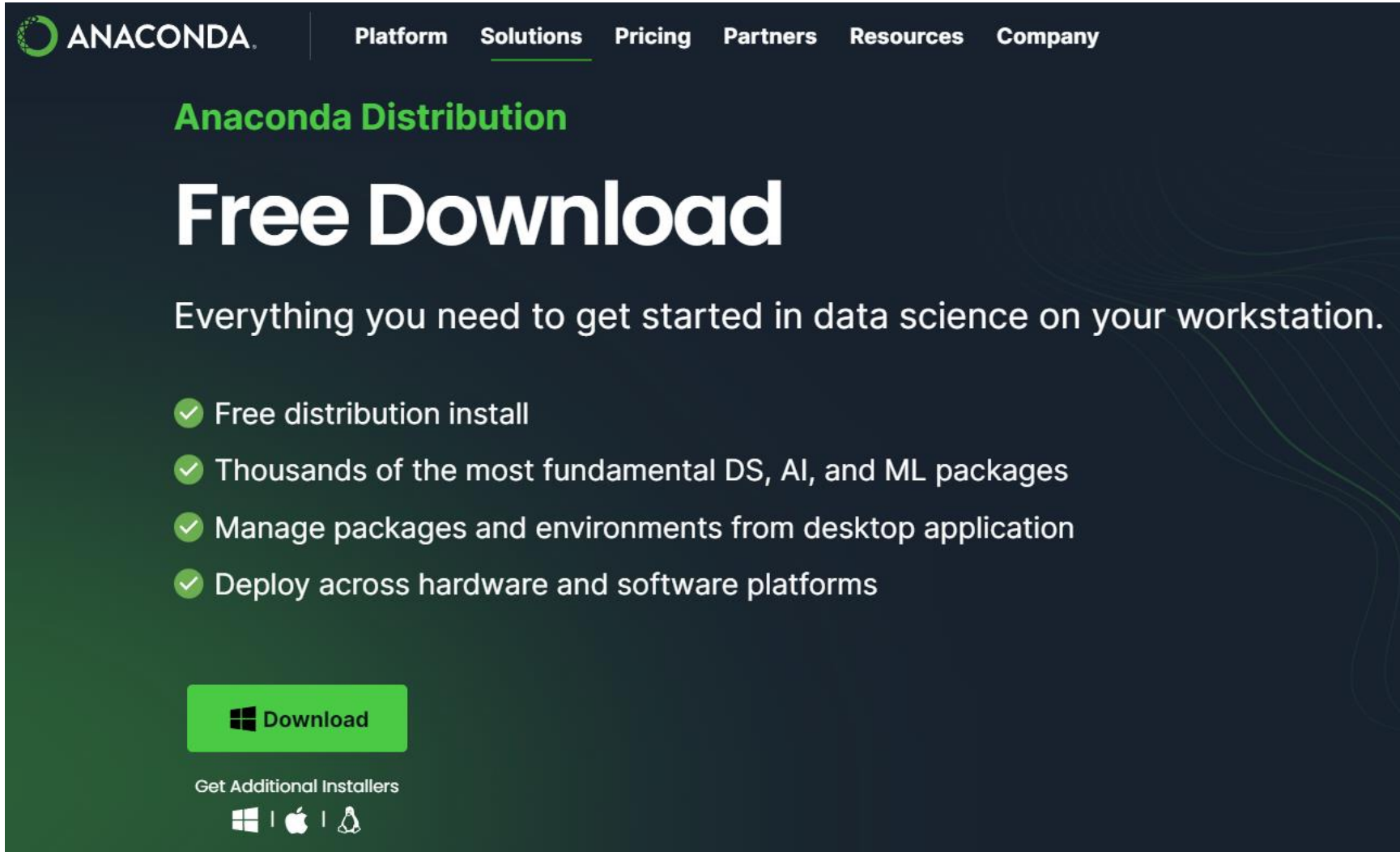
OR

Download the latest version of Python Anaconda from  
<https://www.anaconda.com/download>



# Installation of Anaconda Navigator

Step 1: Go to the Anaconda Website and choose a Python graphical installer



The screenshot shows the Anaconda website's 'Distribution' page. At the top, the Anaconda logo is on the left, and navigation links for Platform, Solutions (underlined), Pricing, Partners, Resources, and Company are on the right. Below the navigation bar, the text 'Anaconda Distribution' is in green, followed by 'Free Download' in large white letters. A subtitle reads 'Everything you need to get started in data science on your workstation.' Below this is a list of four features, each preceded by a green checkmark: 'Free distribution install', 'Thousands of the most fundamental DS, AI, and ML packages', 'Manage packages and environments from desktop application', and 'Deploy across hardware and software platforms'. At the bottom, there is a green 'Download' button with a Windows icon, and a link 'Get Additional Installers' with icons for Windows, macOS, and Linux.

ANACONDA


Platform Solutions Pricing Partners Resources Company

Anaconda Distribution




## Free Download

Everything you need to get started in data science on your workstation.

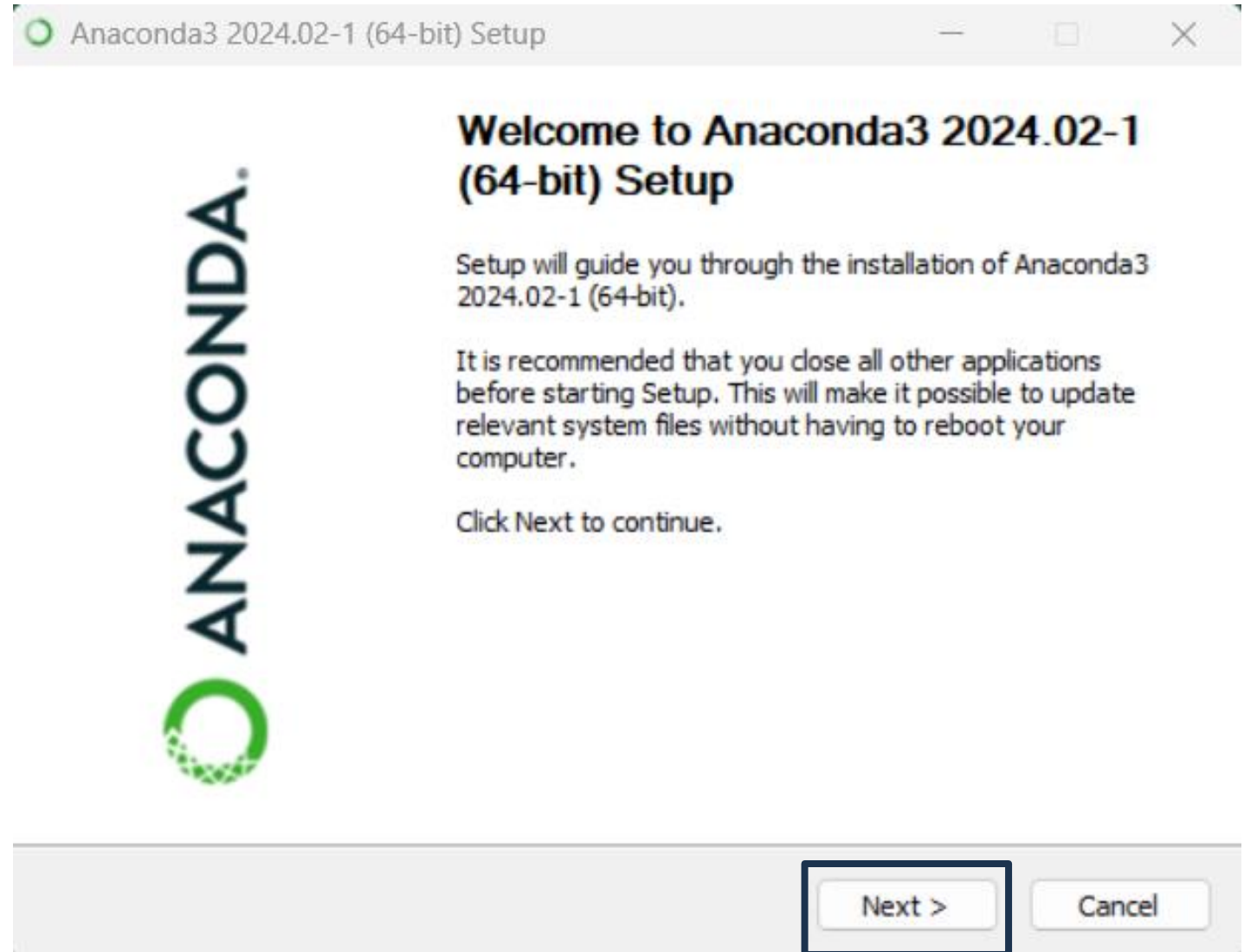
- ✓ Free distribution install
- ✓ Thousands of the most fundamental DS, AI, and ML packages
- ✓ Manage packages and environments from desktop application
- ✓ Deploy across hardware and software platforms

 Download

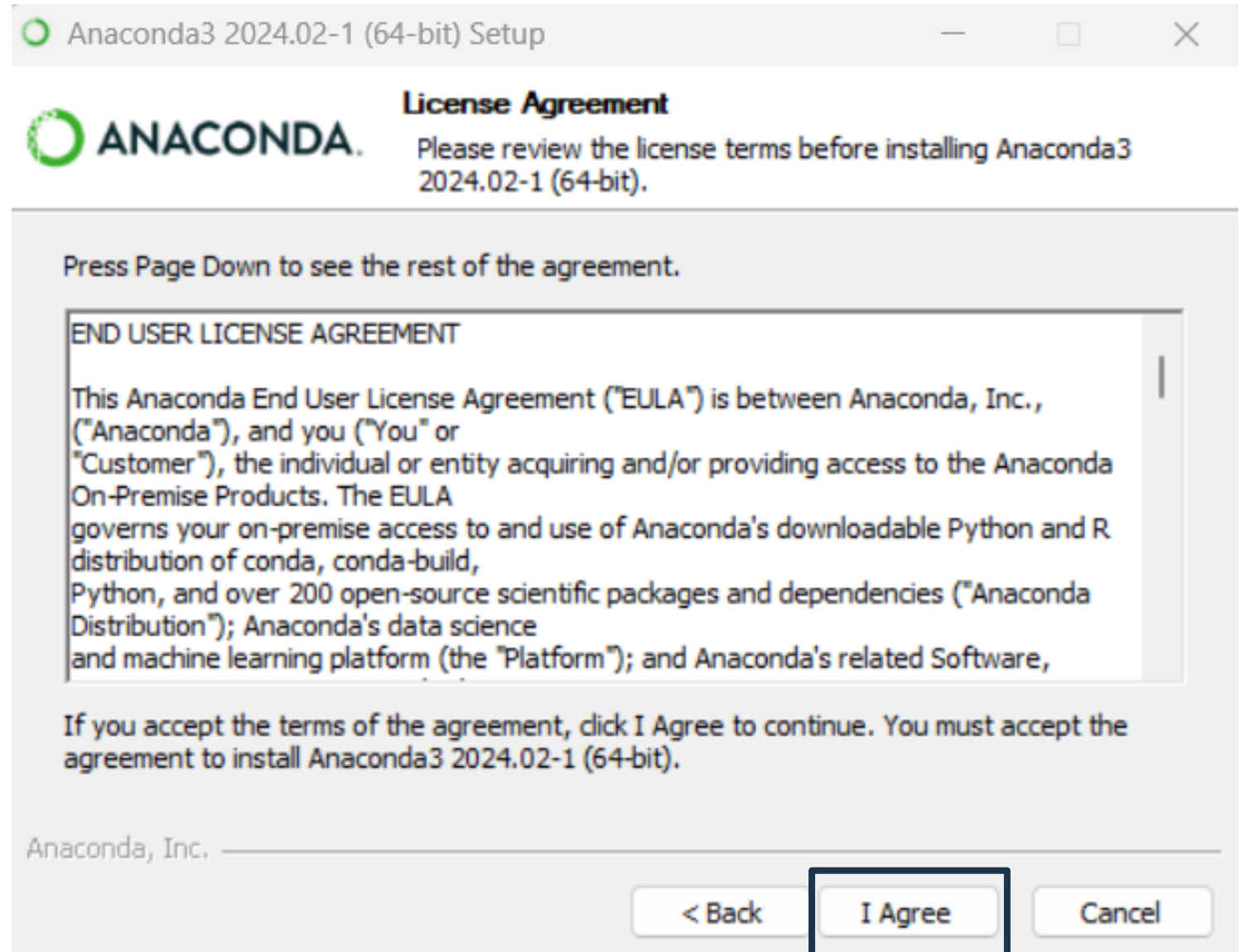
Get Additional Installers

 |  | 

# Installation of Anaconda Navigator



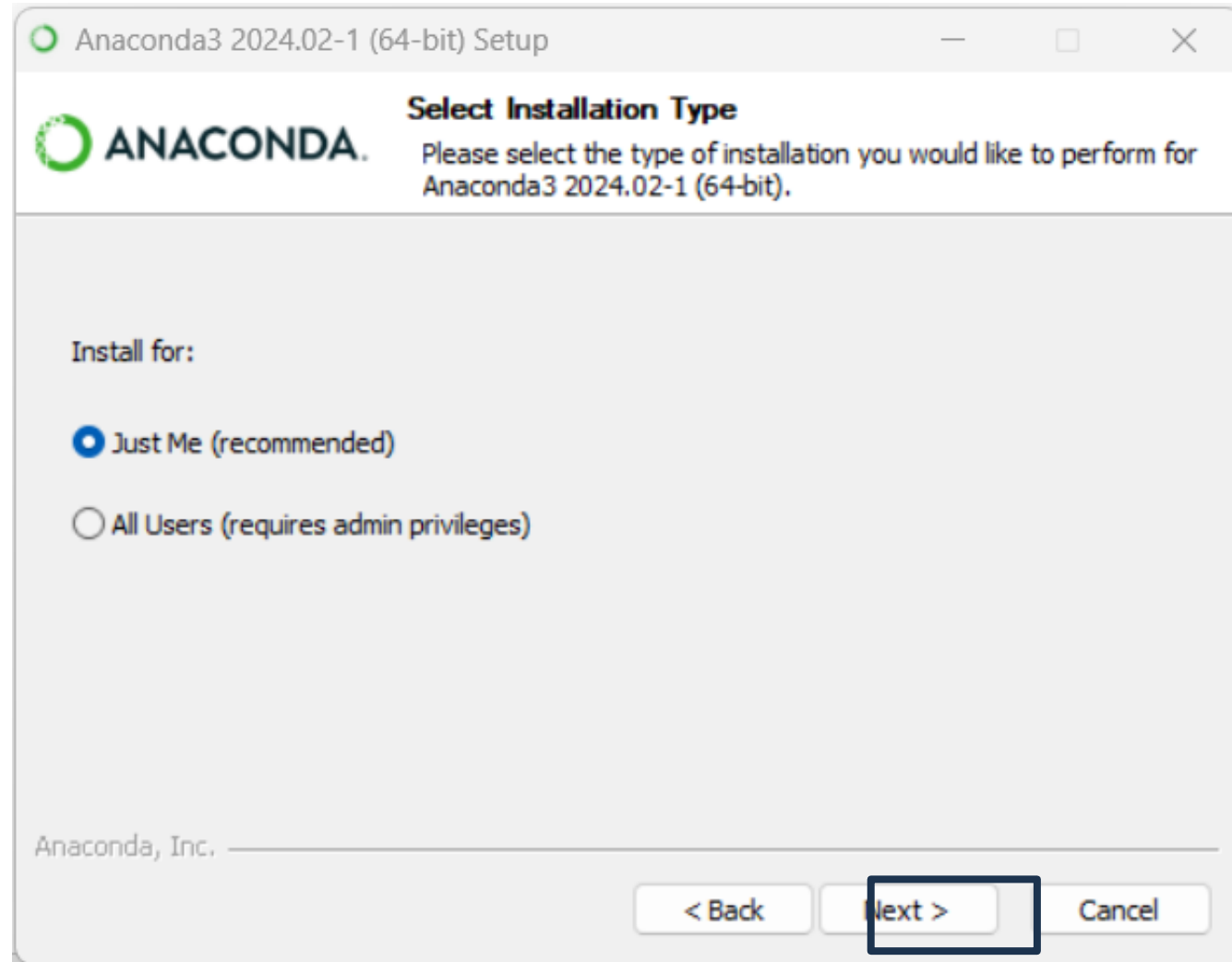
# Installation of Anaconda Navigator



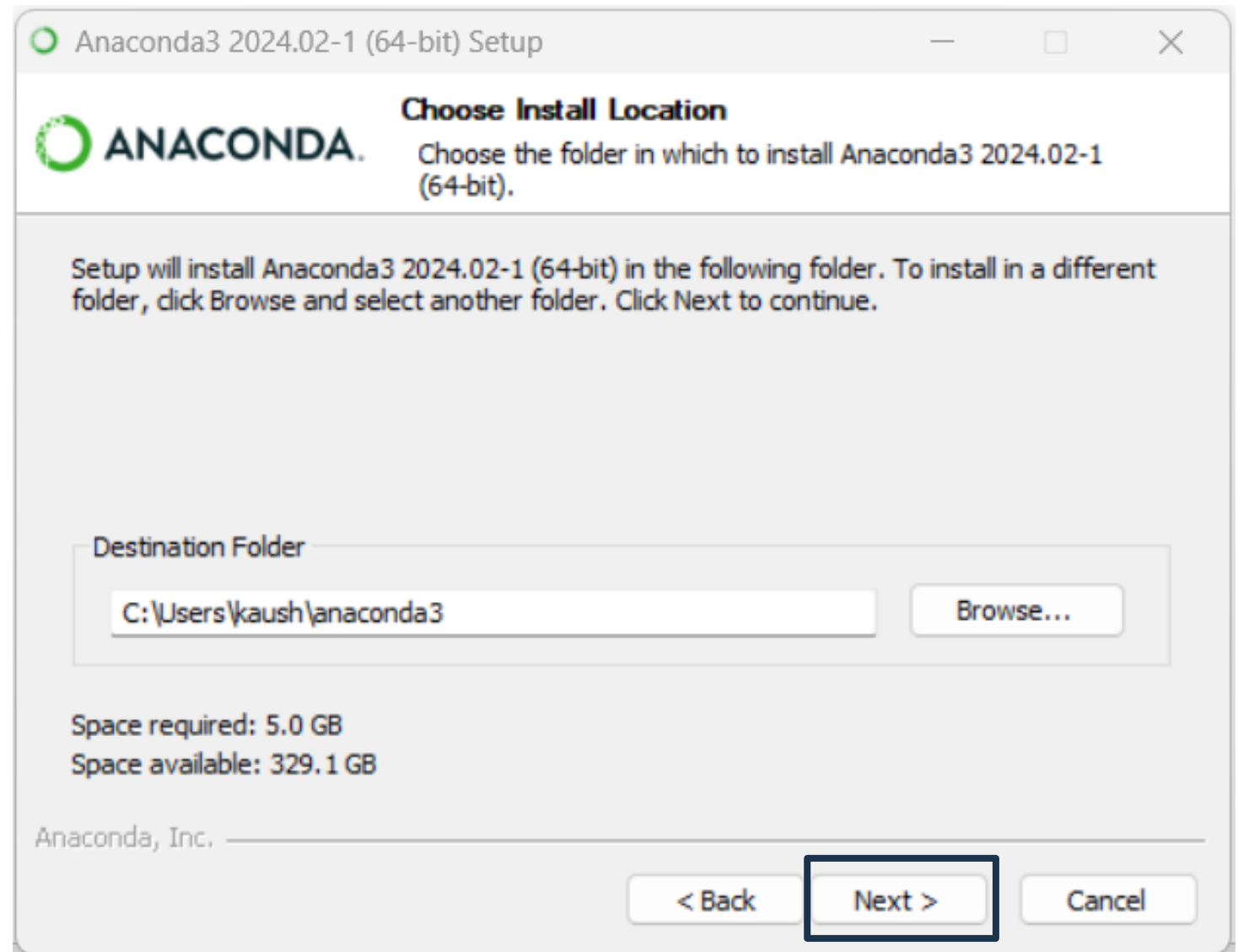


# Installation of Anaconda Navigator

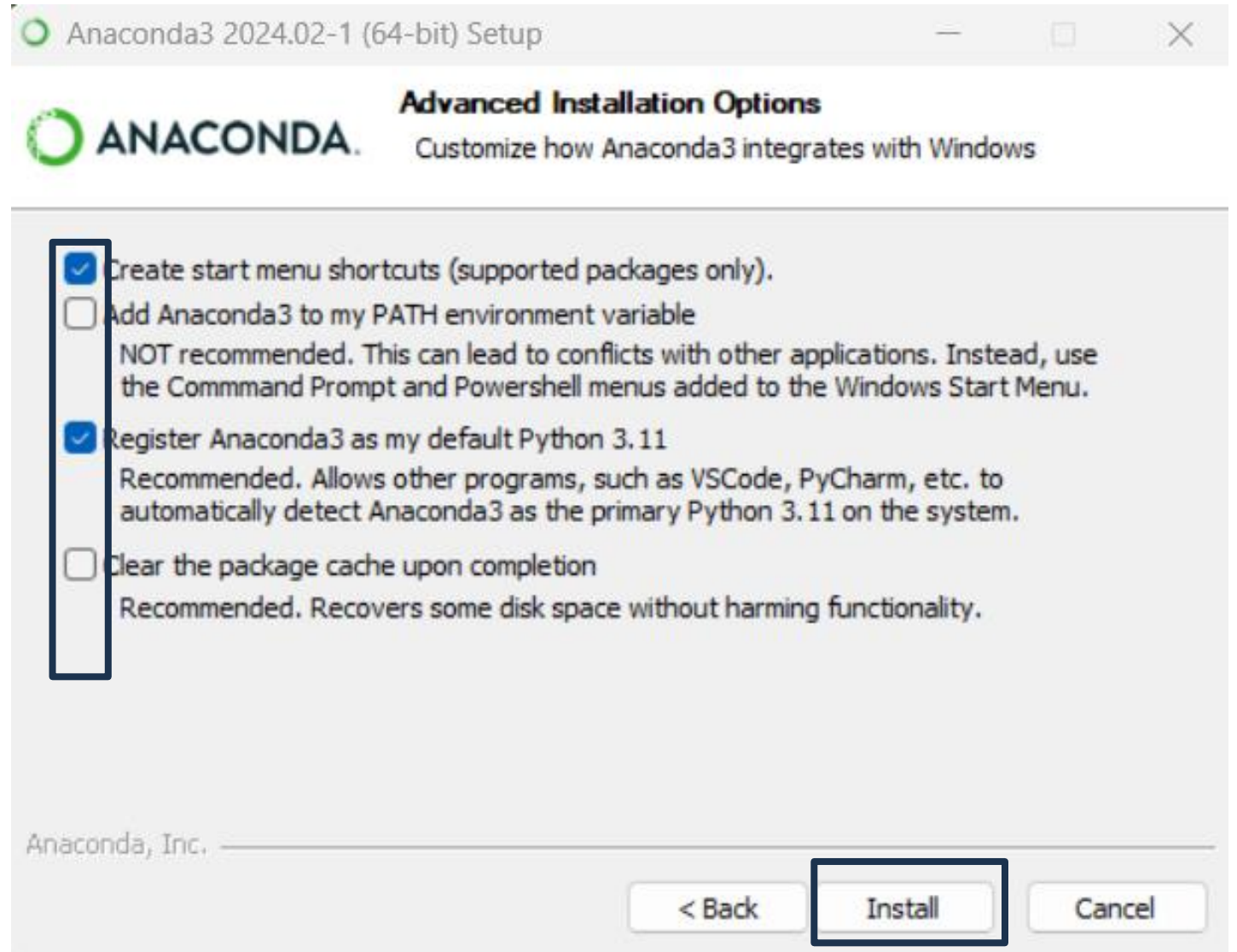
Step 5: Click on 'Next'.



# Installation of Anaconda Navigator

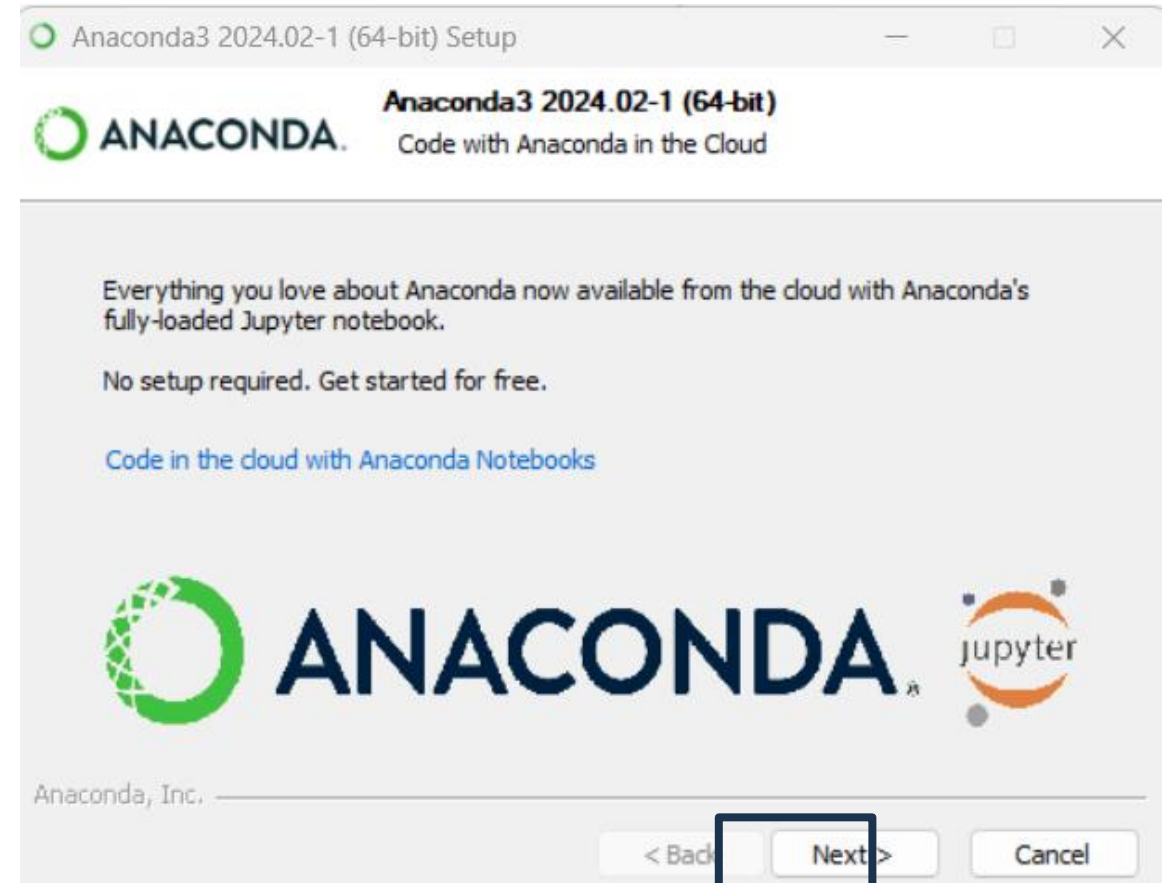
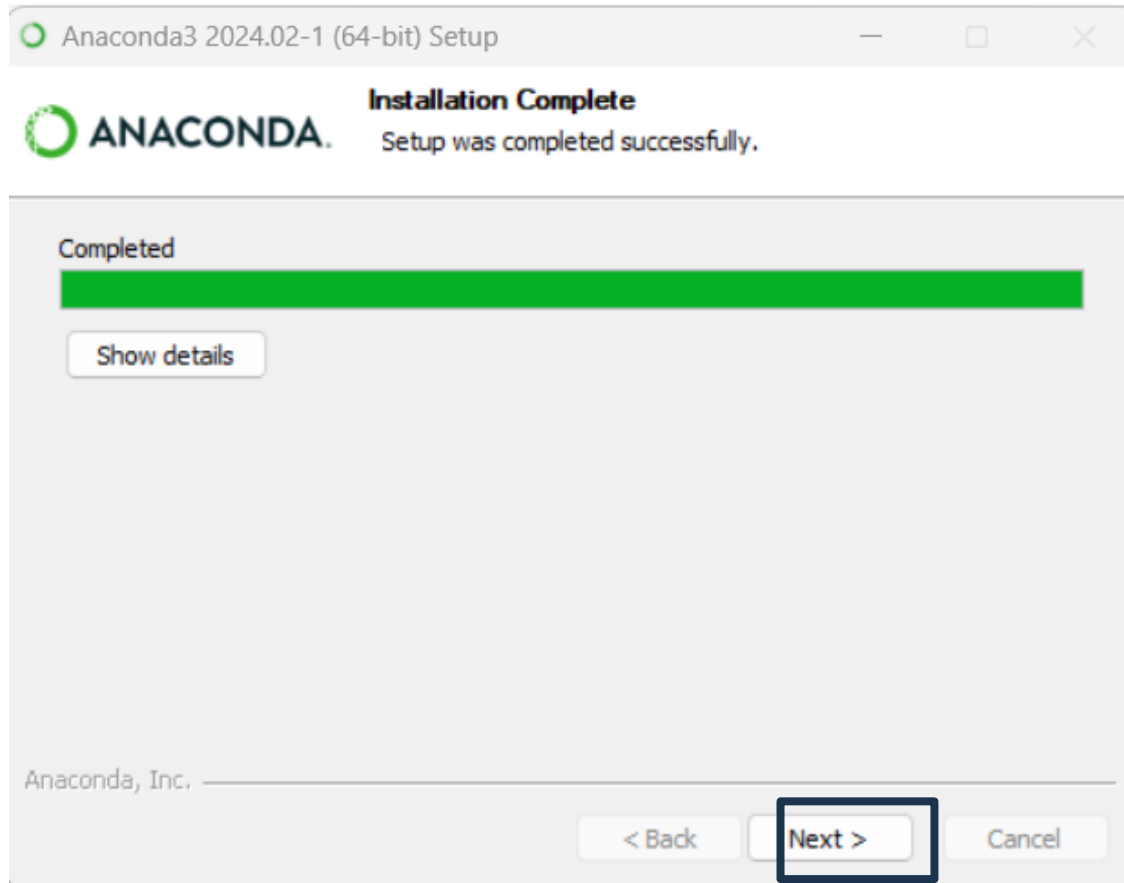


# Installation of Anaconda Navigator



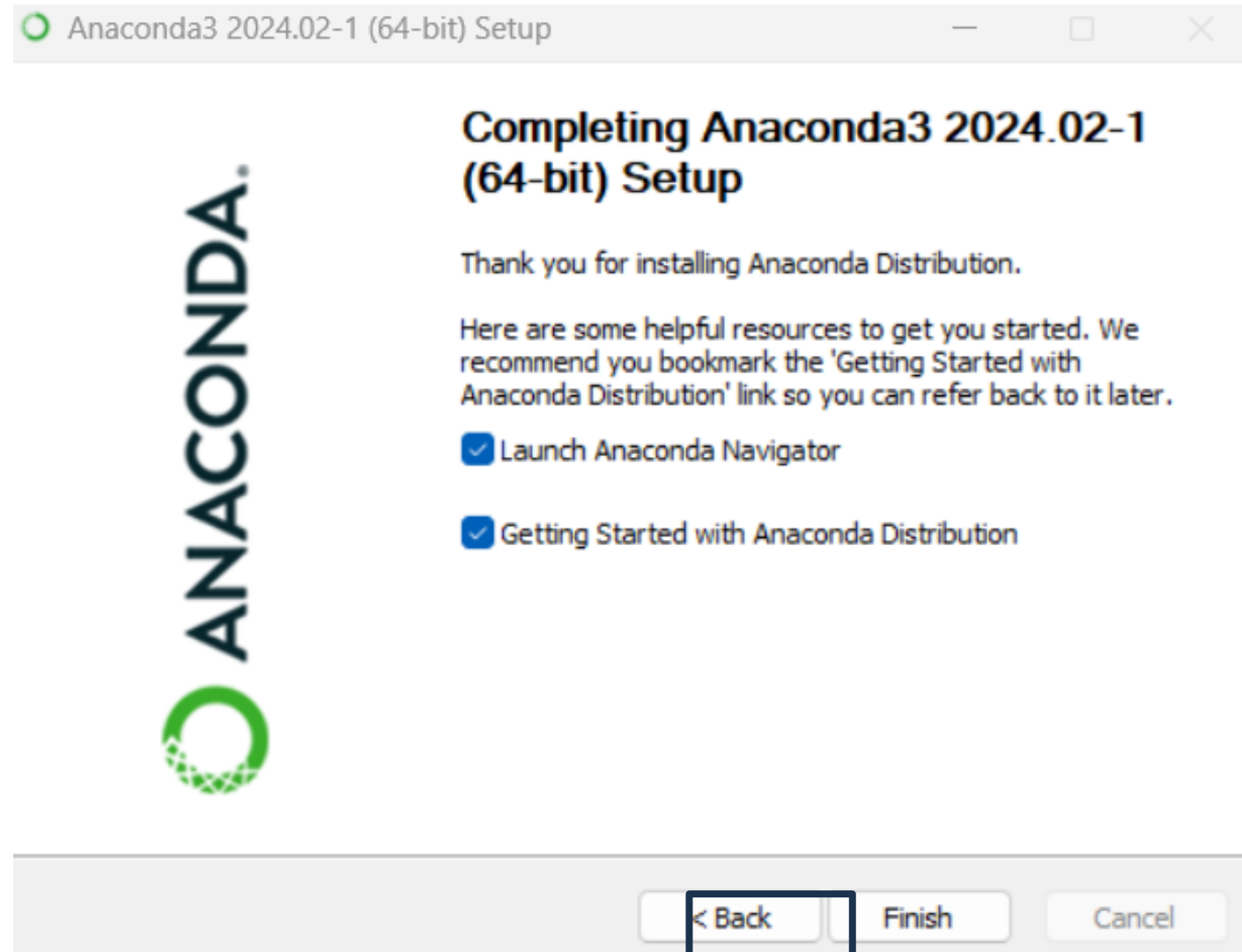
# Installation of Anaconda Navigator

Step 8: Click on 'Next'.

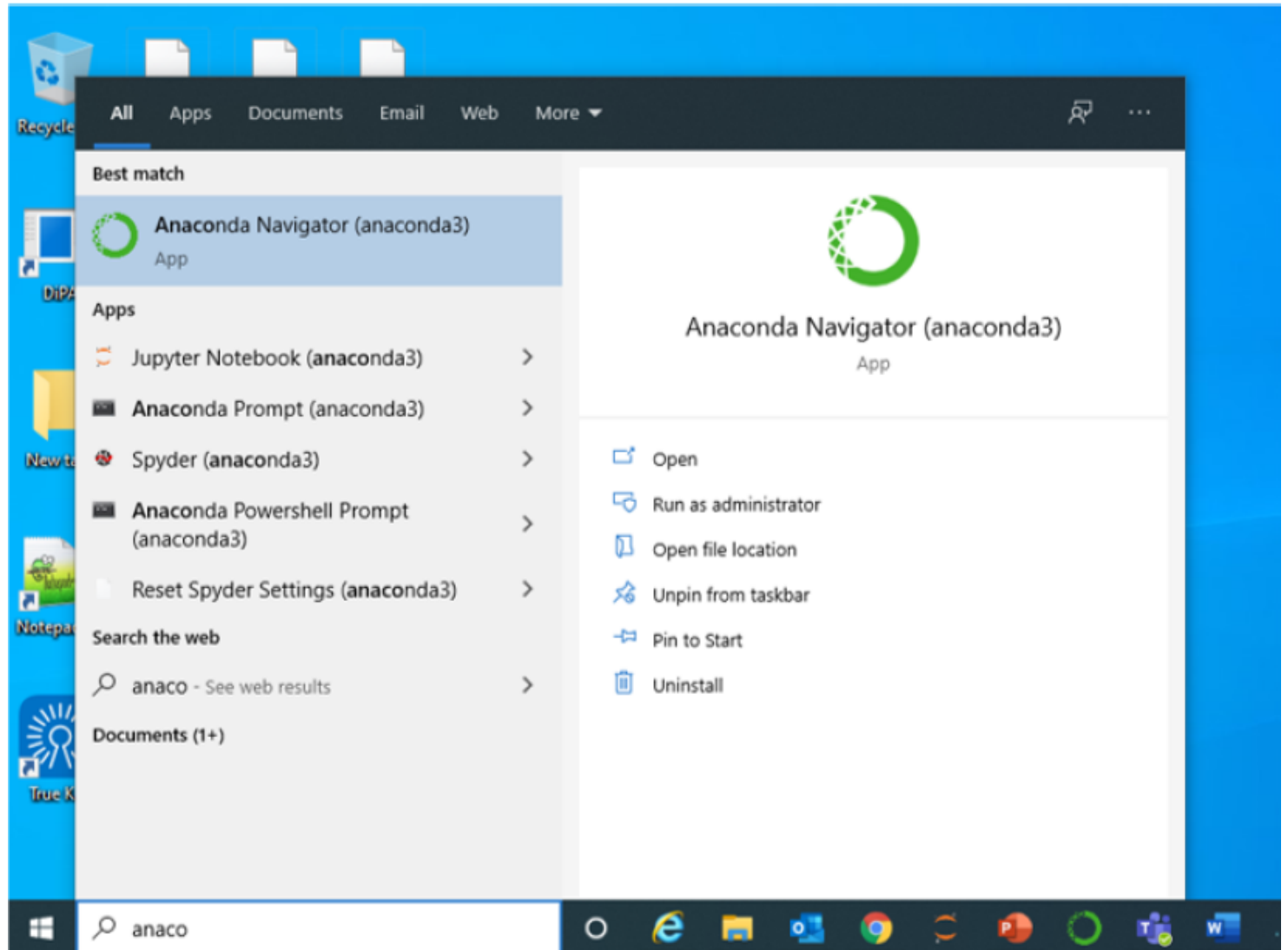


# Installation of Anaconda Navigator

Step 9: Click on 'Finish'.

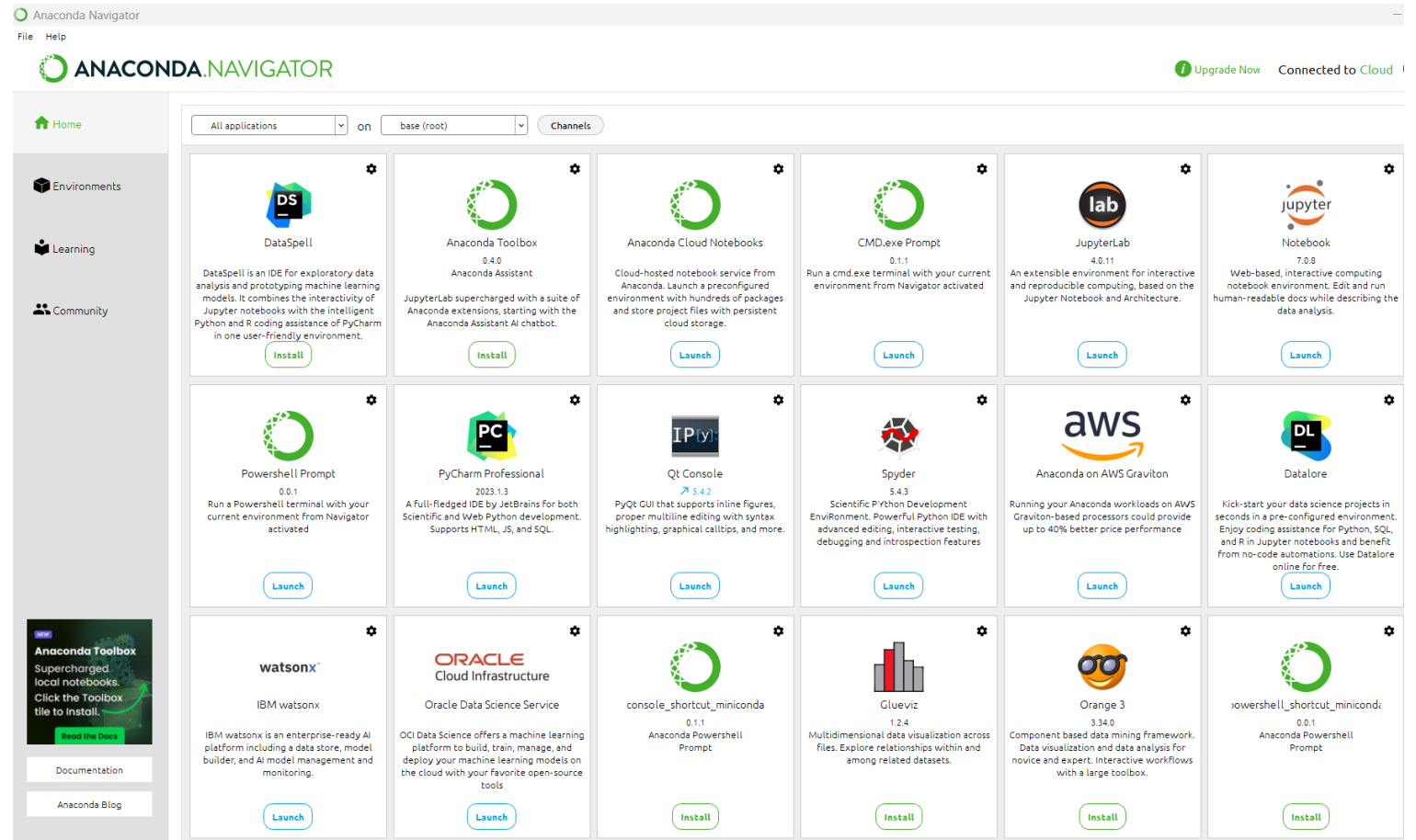


# Installation of Anaconda Navigator



# Jupyter Notebook

Step 11: Click Launch Jupyter Notebook/Spyder or If you have a terminal window open, you can launch Jupyter Notebook, simply by typing 'jupyter notebook' and pressing enter.



# **Best Python libraries for Machine Learning**

Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reasons is its vast collection of libraries. Python libraries that are used in Machine Learning are:

- Numpy
- Pandas
- Scipy
- Scikit-learn
- Theano
- TensorFlow
- Keras
- PyTorch
- Matplotlib



# NumPy

NumPy (Numerical Python) is a powerful library for numerical computing in Python. It provides support for large multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently

- ***ndarray***: A powerful n-dimensional array object which is faster and more efficient than Python's built-in lists.
- ***Mathematical functions***: Functions for performing element-wise operations, such as trigonometric, statistical, and algebraic functions.
- ***Linear algebra***: Tools for linear algebra operations, such as solving linear equations and matrix decompositions.
- ***Random sampling***: Utilities for generating random numbers and random sampling from distributions.

# Dimensions in Arrays

NumPy arrays can have multiple dimensions, allowing users to store data in multilayered structures.

Name	Example
0D (zero-dimensional)	Scalar – A single element
1D (one-dimensional)	Vector- A list of integers.
2D (two-dimensional)	Matrix- A spreadsheet of data
3D (three-dimensional)	Tensor- Storing a color image

# Pandas

Pandas is a data manipulation and analysis library that provides data structures and functions needed to work with structured data seamlessly. It is especially useful for data cleaning and preparation, data analysis, and data visualization. Key features include:

- **Series:** One-dimensional labeled array capable of holding any data type.
- **DataFrame:** Two-dimensional labeled data structure with columns of potentially different types, similar to a table in a database or a data frame in R.
- **Data manipulation:** Tools for reading/writing data in various formats (CSV, Excel, SQL, etc.), handling missing data, merging/joining datasets, and reshaping data.

# Pandas DataFrame

- Pandas DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns).
- A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns.
- Pandas DataFrame consists of three principal components, the data, rows, and columns.

The diagram illustrates a Pandas DataFrame with the following components and annotations:

- Column names:** Indicated by a blue arrow pointing to the header row.
- Columns axis=1:** Indicated by a grey arrow pointing to the column headers.
- Index label:** Indicated by a purple arrow pointing to the index values (0-6).
- Index axis=0:** Indicated by a grey arrow pointing to the index values.
- Missing value:** A pink box highlights the 'NaN' value in the 'Number' column for row index 3.
- Data:** An orange box highlights the numerical data values in the 'Age', 'Height', 'Weight', and 'Salary' columns for row index 5.

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston University	NaN
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0	6-8	235.0	LSU	1170960.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1824360.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN	6-9	260.0	Ohio State	2569260.0
6	Evan Turner	Boston Celtics	11.0	SG	27.0	6-7	220.0	Ohio State	3425510.0

# Matplotlib

Matplotlib is a plotting library for creating static, animated, and interactive visualizations in Python. It is highly customizable and integrates well with other libraries like NumPy and Pandas. Key features include:

- ***Plotting functions***: A variety of functions for creating line plots, scatter plots, bar charts, histograms, and more.
- ***Customization***: Extensive options for customizing plots, such as titles, labels, legends, and color schemes.
- ***Interactive plots***: Support for interactive plots in Jupyter Notebooks and other environments.

# SCIKIT-LEARN

Scikit-learn is a machine-learning library that provides simple and efficient tools for data mining and data analysis. It builds on NumPy, SciPy, and Matplotlib and is designed for creating machine learning models and pipelines. Key features include:

- ***Supervised learning***: Algorithms for classification and regression, such as linear regression, logistic regression, decision trees, and support vector machines.
- ***Unsupervised learning***: Algorithms for clustering and dimensionality reduction, such as k-means and principal component analysis (PCA).
- ***Model selection***: Tools for cross-validation, grid search, and metrics to evaluate model performance.
- ***Preprocessing***: Utilities for feature extraction, normalization, and transformation.

# Basic Python Syntax

- ***Variables and Data Types:*** Variables store data, and Python supports various data types like integers, floats, strings, and booleans.
- ***Comments:*** Use “#” for single-line comments and triple quotes ''' or """ for multi-line comments.
- ***Indentation:*** Python uses indentation to define blocks of code. Each block of code (like functions, loops) must be indented by the same amount of space.
- ***Basic Input/Output:***

# Data Structures

Data Structures are containers that organize and store data. Common data structures in Python include lists, dictionaries, and tuples.

- ***Lists***: Ordered, mutable collections of items. Defined using square brackets [ ].
- ***Dictionaries***: Unordered, mutable collections of key-value pairs. Defined using curly braces { }.
- ***Tuples***: Ordered, immutable collections of items. Defined using parentheses () .



# Functions and Control Flow

Functions are reusable blocks of code that perform a specific task. Control Flow manages the flow of execution based on conditions.

- ***Functions***: Defined using the def keyword.
- ***Conditional Statements***: Use if, elif, and else to make decisions.

# Loops:

*For Loop:* Iterates over a sequence.

*While Loop:* Repeats if a condition is true.

fillna

day	temperature	windspeed	event
1/1/2017	32	6	Rain
1/4/2017		9	Sunny
1/5/2017	28		Snow
1/6/2017		7	
1/7/2017	32		Rain
1/8/2017			Sunny
1/9/2017			
1/10/2017	34	8	Cloudy
1/11/2017	40	12	Sunny

```
new_df = df.fillna(method="ffill")
```



day	temperature	windspeed	event
1/1/2017	32	6	Rain
1/4/2017	32	9	Sunny
1/5/2017	28	9	Snow
1/6/2017	28	7	Snow
1/7/2017	32	7	Rain
1/8/2017	32	7	Sunny
1/9/2017	32	7	Sunny
1/10/2017	34	8	Cloudy

# fillna

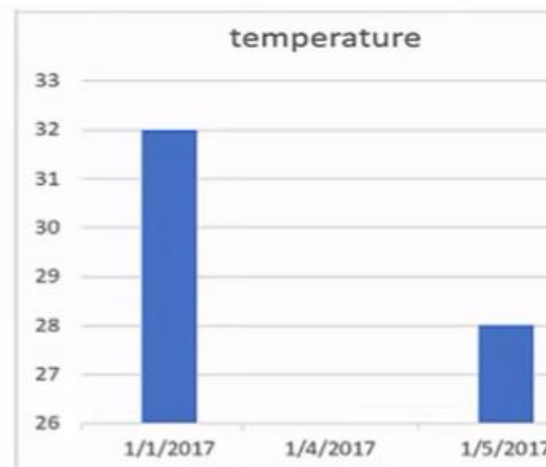
day	temperature	windspeed	event
1/1/2017	32	6	Rain
1/4/2017		9	Sunny
1/5/2017	28		Snow
1/6/2017		7	
1/7/2017	32		Rain
1/8/2017			Sunny
1/9/2017			
1/10/2017	34	8	Cloudy
1/11/2017	40	12	Sunny

new\_df = df.fillna(method='bfill')

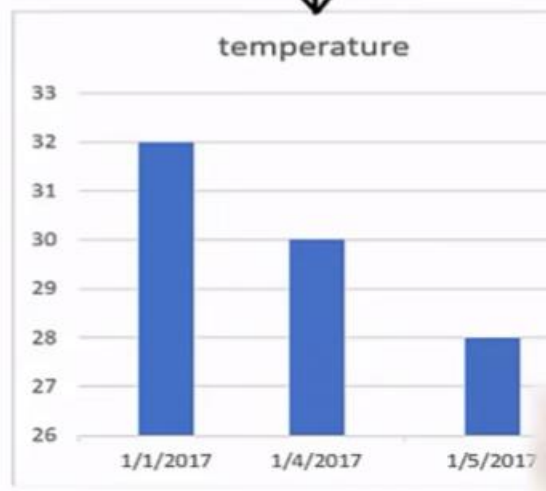


day	temperature	windspeed	event
1/1/2017	32	6	Rain
1/4/2017	28	9	Sunny
1/5/2017	28	7	Snow
1/6/2017	32	7	Rain
1/7/2017	32	8	Rain
1/8/2017	34	8	Sunny
1/9/2017	34	8	Cloudy
1/10/2017	34	8	Cloudy

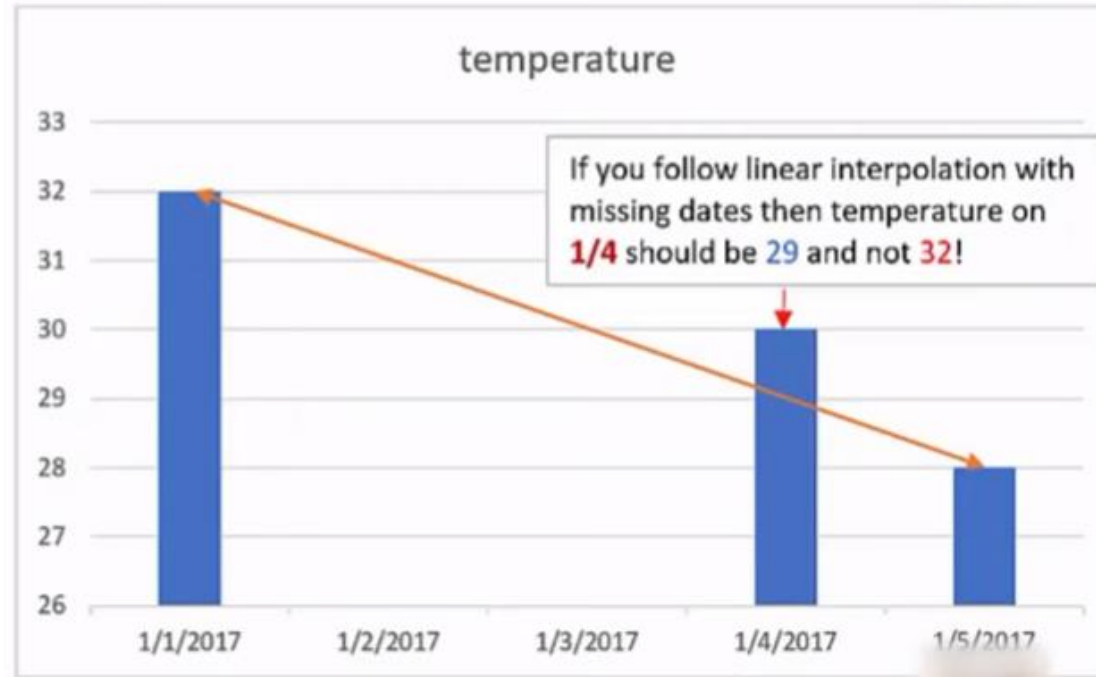
# fillna



`new_df = df.interpolate()`



fillna



india\_weather

	city	humidity	temperature
0	mumbai	80	32
1	delhi	60	45
2	banglore	78	30

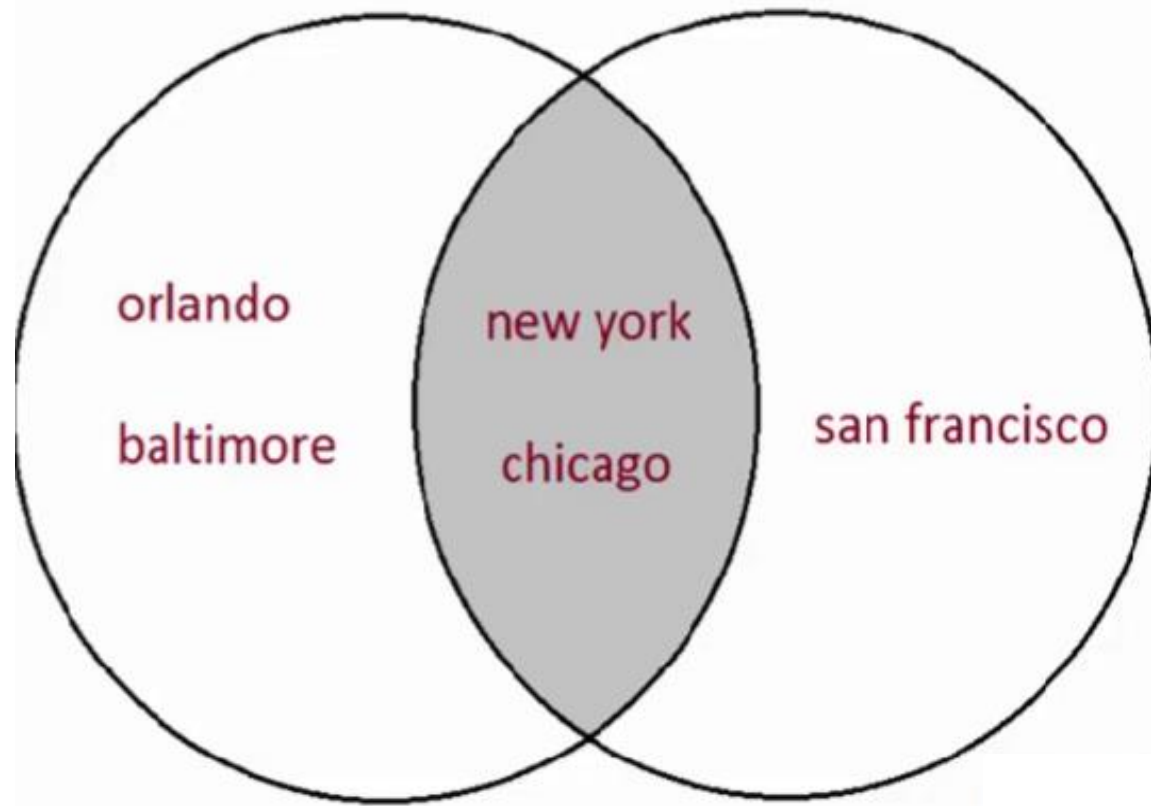
us\_weather

	city	humidity	temperature
0	new york	68	21
1	chicago	65	14
2	orlando	75	35

```
df = pd.concat([india_weather, us_weather])
```

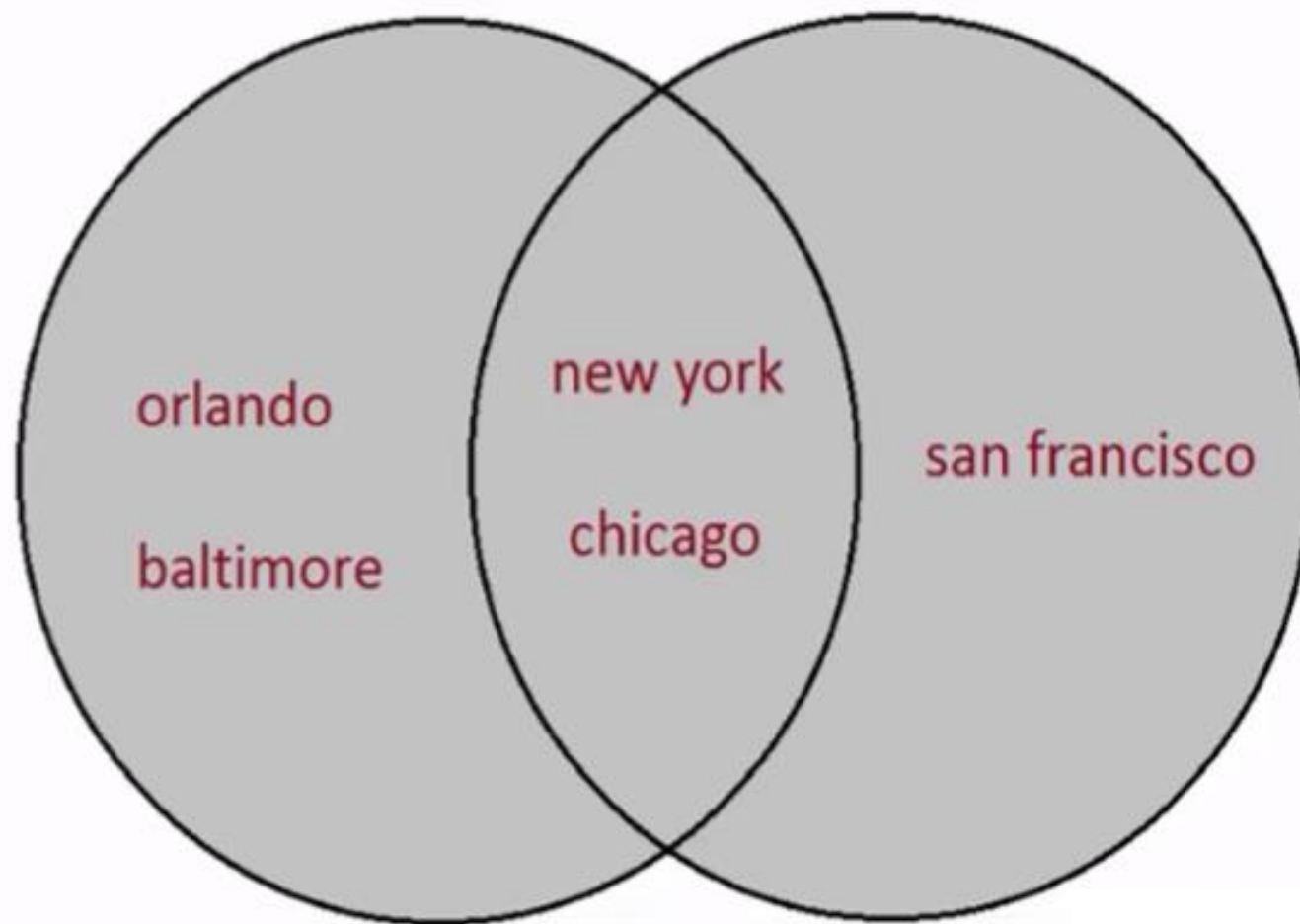
	city	humidity	temperature
0	mumbai	80	32
1	delhi	60	45
2	banglore	78	30
0	new york	68	21
1	chicago	65	14
2	orlando	75	35

inner join

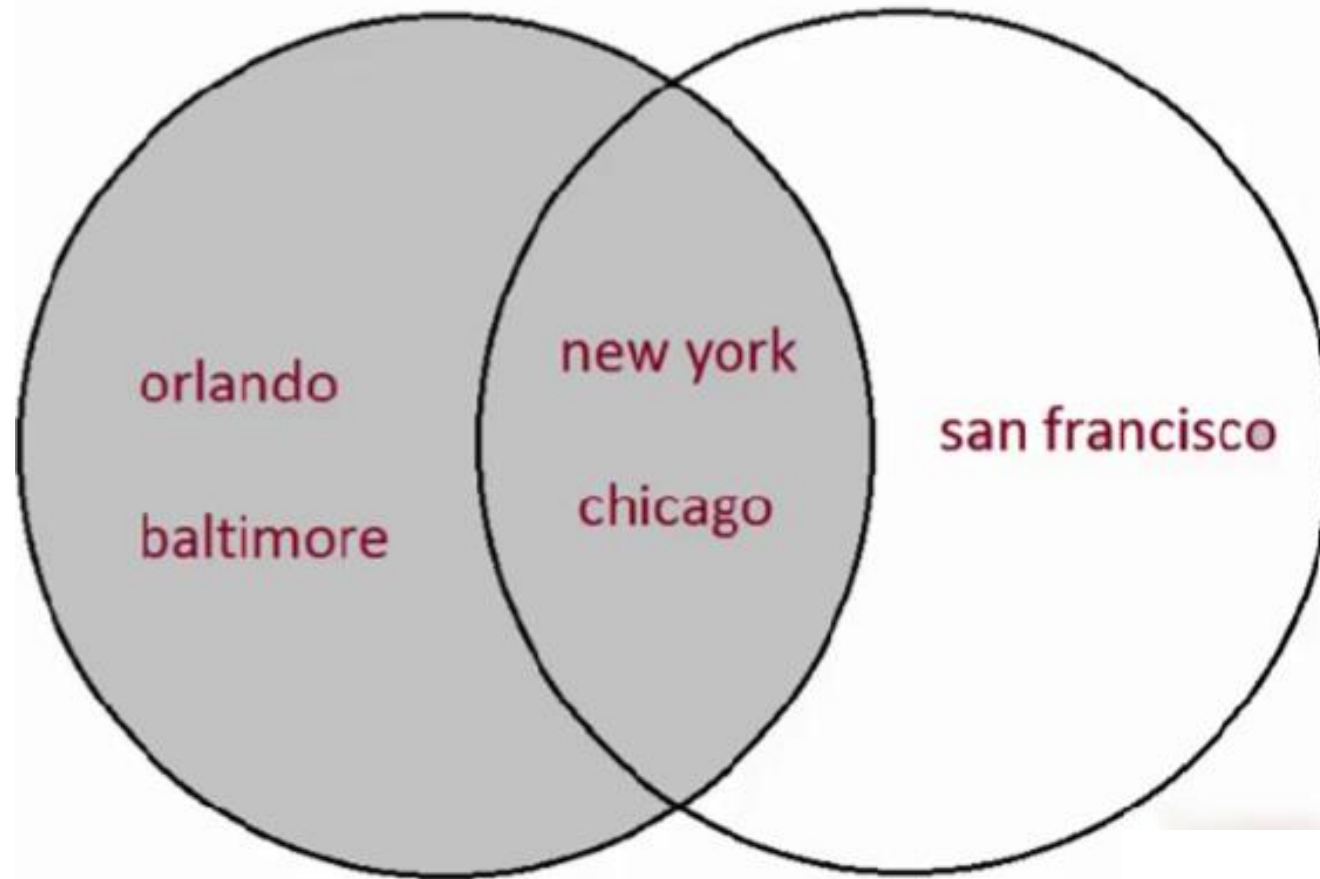




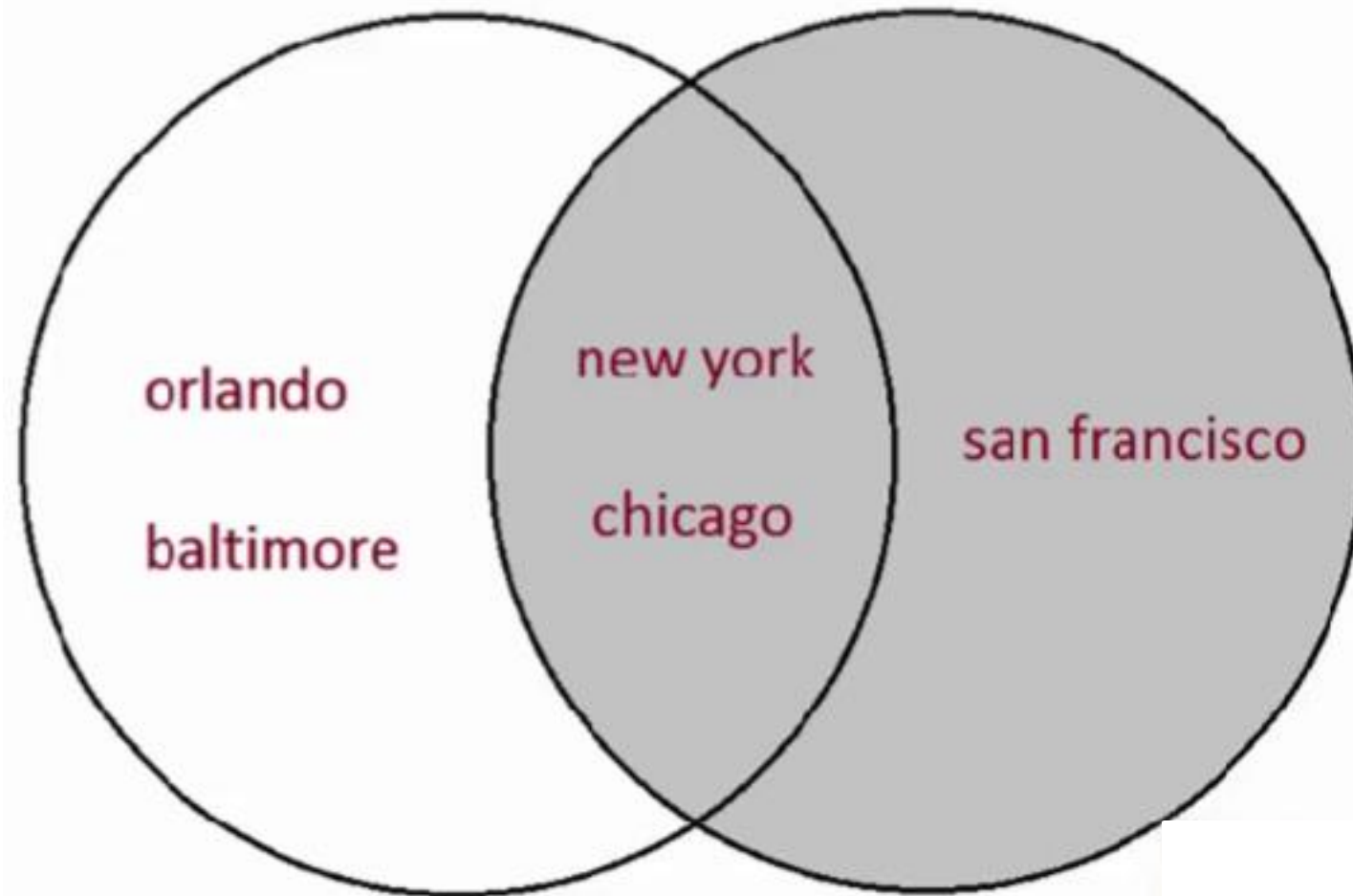
outer join



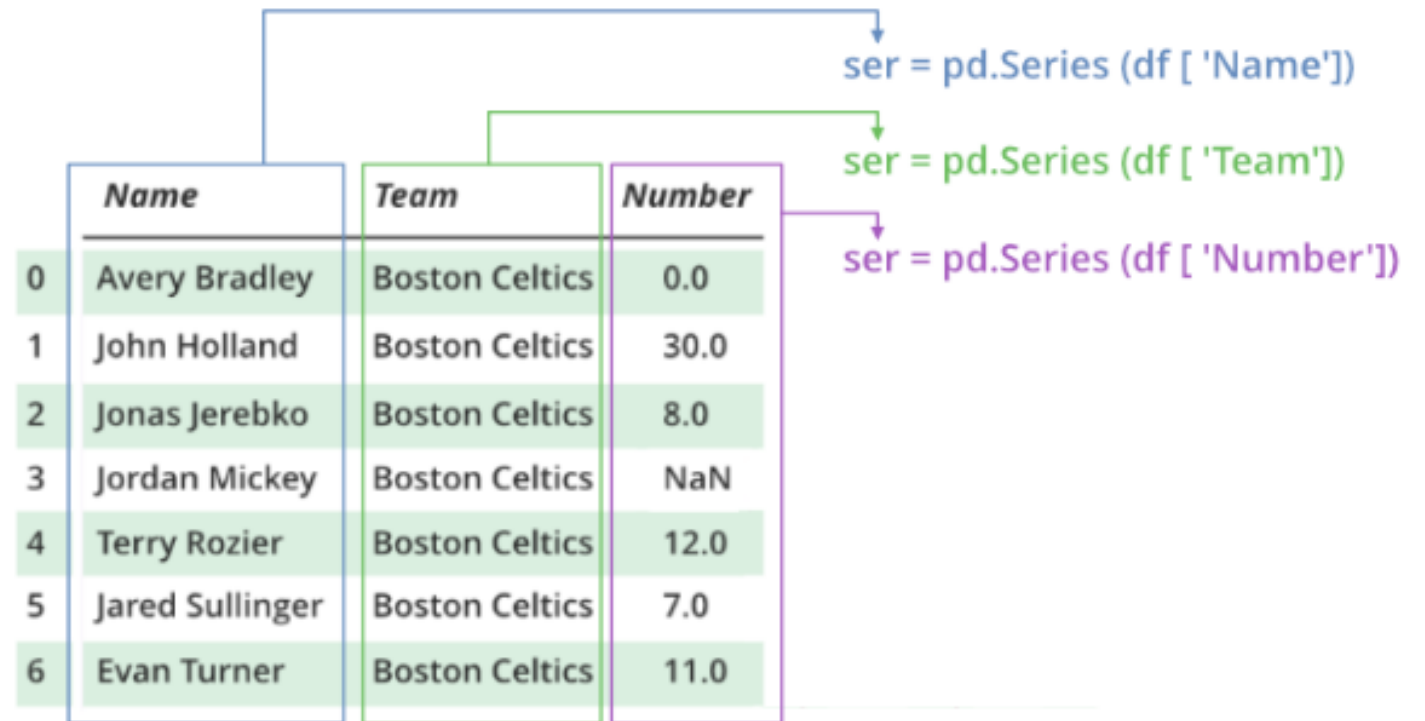
left join



right join



# Pandas Series



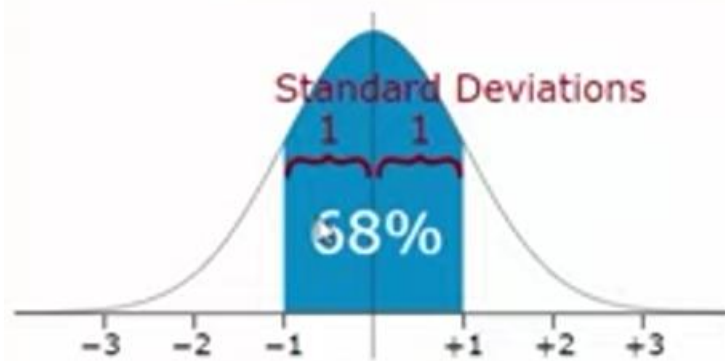
# Outliers

	test score (from 100)	percentage	percentile rank
mohan	69	69%	100.00%
maria	56	56%	50.00%
sakib	45	45%	25.00%
tao	32	32%	12.50%
virat	27	27%	0.00%
khusbu	65	65%	75.00%
dmitry	61	61%	62.50%
selena	66	66%	87.50%
john	45	45%	25.00%

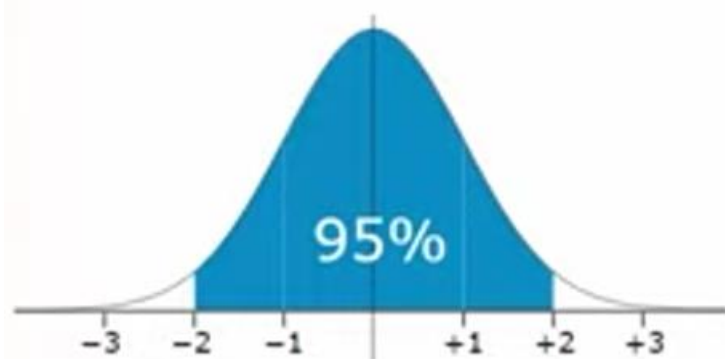
$$Z = \frac{X - \mu}{\sigma}$$

$\mu$  = mean

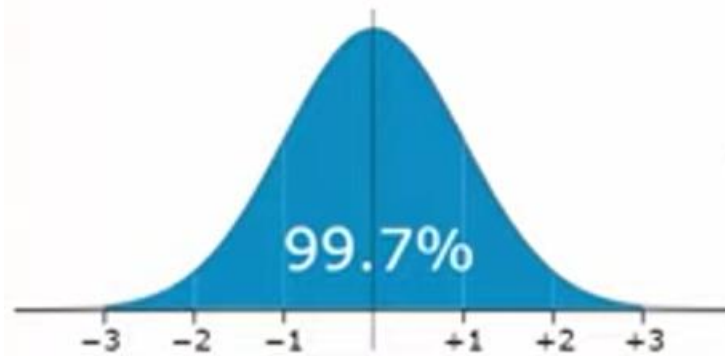
$\sigma$  = standard deviation



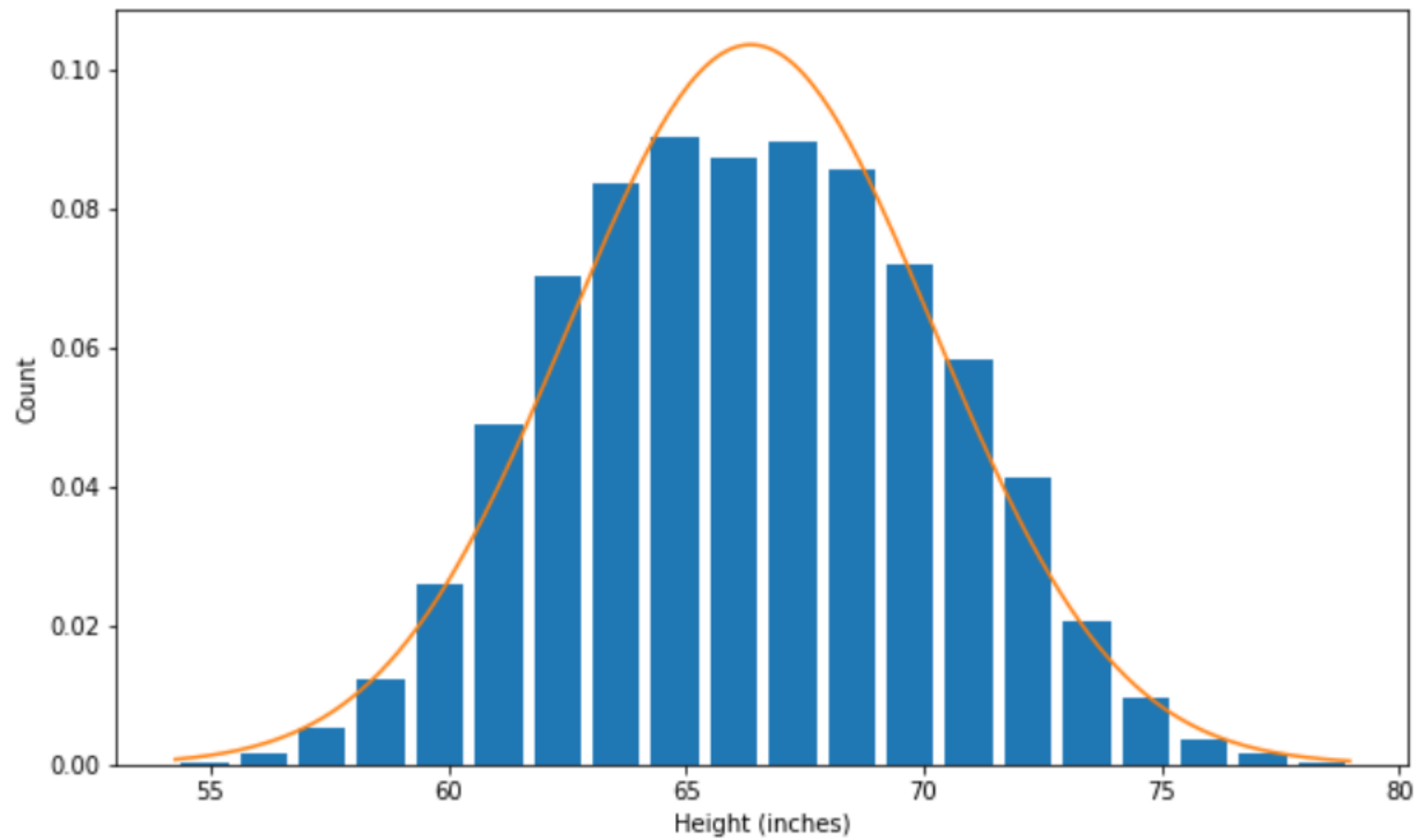
**68%** of values are within  
**1 standard deviation** of the mean



**95%** of values are within  
**2 standard deviations** of the mean



**99.7%** of values are within  
**3 standard deviations** of the mean



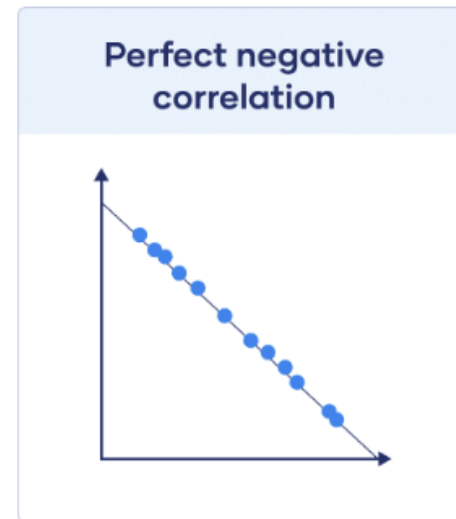
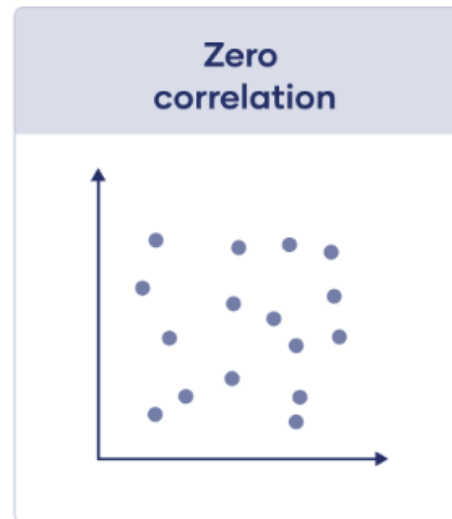
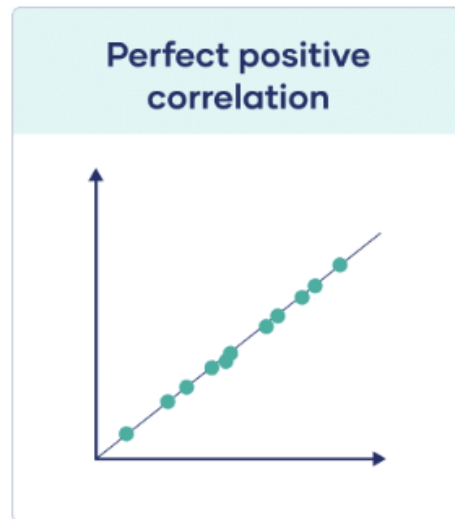




# Correlation Coefficient

- A correlation coefficient is a descriptive statistic.
- A correlation coefficient is a number between -1 and 1 that tells you the strength and direction of a relationship between variables.
- In other words, it reflects how similar the measurements of two or more variables are across a dataset.

Correlation coefficient value	Correlation type	Meaning
1	Perfect positive correlation	When one variable changes, the other variables change in the same direction.
0	Zero correlation	There is no relationship between the variables.
-1	Perfect negative correlation	When one variable changes, the other variables change in the opposite direction.

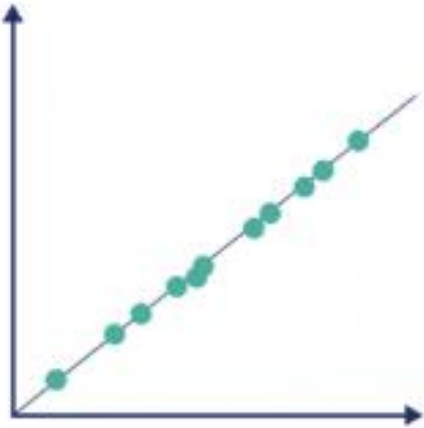


Correlation coefficient	Correlation strength	Correlation type
-.7 to -1	Very strong	Negative
-.5 to -.7	Strong	Negative
-.3 to -.5	Moderate	Negative
0 to -.3	Weak	Negative
0	None	Zero
0 to .3	Weak	Positive
.3 to .5	Moderate	Positive
.5 to .7	Strong	Positive

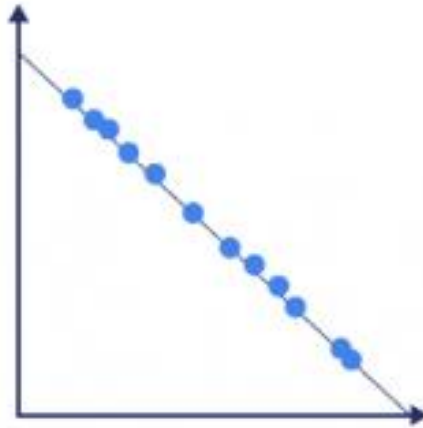
**Interpreting a correlation coefficient**

# Visualizing linear correlations

Perfect positive correlation

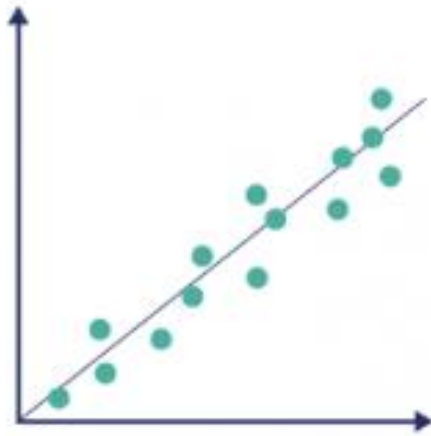


Perfect negative correlation

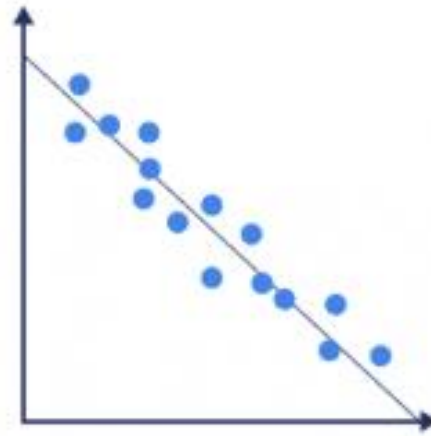


# Visualizing linear correlations

High positive  
correlation

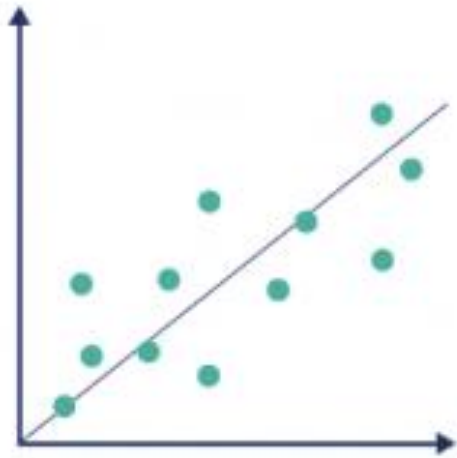


High negative  
correlation

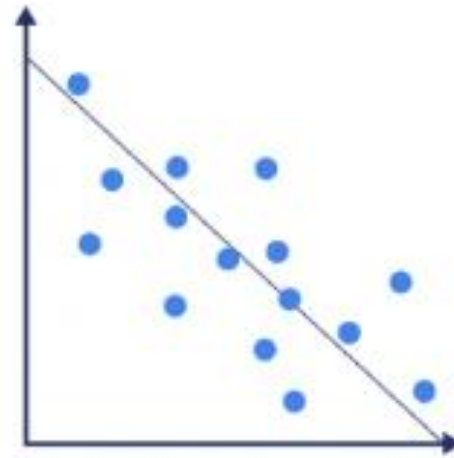


# Visualizing linear correlations

Low positive  
correlation



Low negative  
correlation



# Types of correlation coefficients

Correlation coefficient	Type of relationship	Levels of measurement	Data distribution
Pearson's r	Linear	Two quantitative (interval or ratio) variables	Normal distribution
Spearman's rho	Non-linear	Two ordinal, interval or ratio variables	Any distribution
Point-biserial	Linear	One dichotomous (binary) variable and one quantitative (interval or ratio) variable	Normal distribution
Cramér's V (Cramér's $\phi$ )	Non-linear	Two nominal variables	Any distribution
Kendall's tau	Non-linear	Two ordinal, interval or ratio variables	Any distribution