



# Pandas: Risk and Returns Over Time

FinTech  
Lesson 3.3



# Class Objectives



By the end of this unit, you will be able to:



Group data in a DataFrame to perform calculations on the grouped data.



Manipulate datetime data in different formats: single variables, DataFrame columns, and series.



Identify the calculations that can be done with datetime data & declare and use a DateTimeIndex.



Calculate mean, median, and standard deviation using Pandas & apply standard deviation to risk analysis use cases.



Determine risk by identifying how stocks deviate from the mean.



Describe Sharpe ratios and calculate them using Pandas DataFrames.



Welcome/Refresher

# Returns Over Time

---

Returns over time require investment close prices, i.e., stock data. Stock close prices can be acquired from Google Sheets via the Google Finance function.

Returns over time can be calculated using the `pct_change()` function.





# Instructor Demonstration

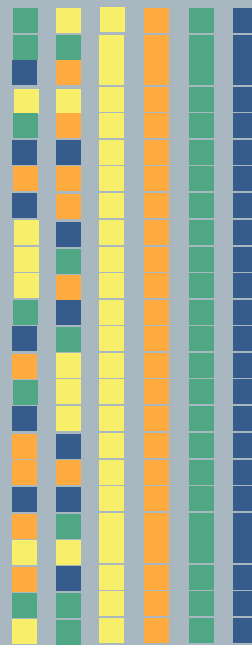
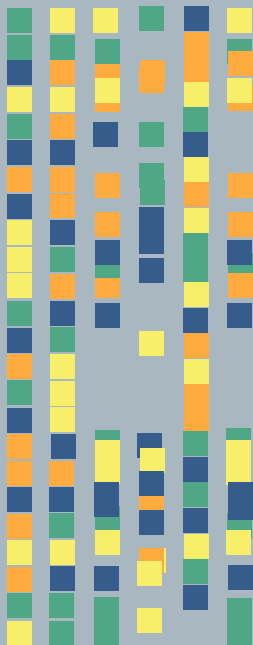
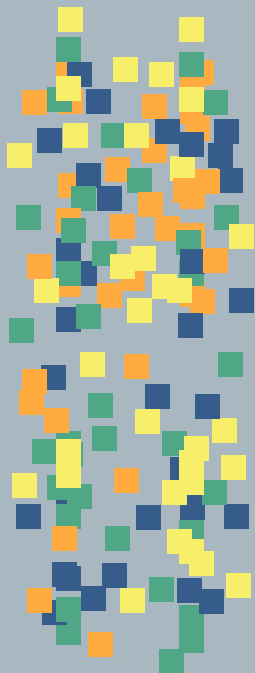
## Google Finance Function

# Sorting

# Sorting

---

Data is not always organized in the best way for analysis. Sometimes, data needs to be cleaned and sorted.



# Sorting

---

The `sort_values` function in Pandas can be used to sort a DataFrame. Sorting data helps improve visual representation of data.

Data can be sorted in either ascending or descending order.

```
sort_values(ascending=True)
```



**Consider dates:** would you rather see dates sorted or randomly listed?





# Instructor Demonstration

## Sorting DataFrames



## Activity: Out of Sorts

In this activity, you will extract data for a single ticker from Google Sheets via the in-built Google Finance function and calculate daily returns for the year 2019. The data will then be sorted in descending order to identify the top 5 performing days for returns.

(Instructions sent via Slack.)

**Suggested Time:**  
15 Minutes





**Time's Up!** Let's Review.

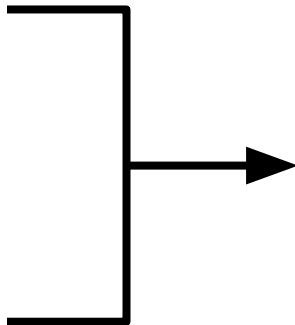
# Grouping

# Grouping

A key component of data analysis is grouping data. **Grouping** allows for similar data to be aggregated or manipulated as groups.

Example aggregations that can be done on groups are adding, summing, determining min and max, etc.

| Category | Sales |
|----------|-------|
| a        | 1     |
| a        | 2     |
| b        | 10    |
| b        | 9     |



| Category | Sales |
|----------|-------|
| a        | 3     |
| b        | 19    |

# Grouping

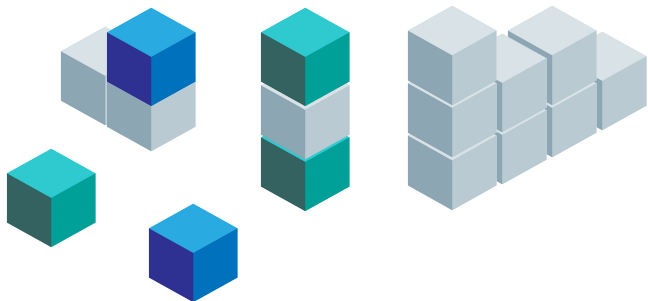
Behind the scenes, the Pandas `groupby` function does the following:



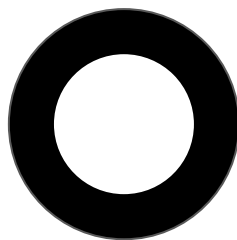
**Splits** the data into groups based on certain criteria.



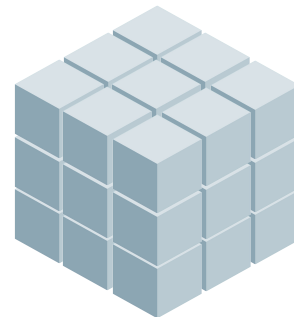
**Applies** a function to each group independently.



Splitting Data



Applying a Function



Combining Results



# Instructor Demonstration

## Grouping DataFrames



## Activity: Group Dynamics

In this activity, you will work with historical cryptocurrency data. You will load in cryptocurrency data, group data by each crypto, and then perform aggregations to analyze price trends. You will then plot the results.

(Instructions sent via Slack.)

**Suggested Time:**  
15 Minutes

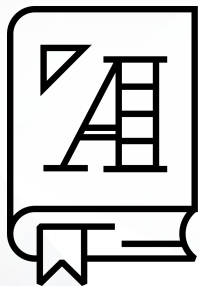






**Time's Up!** Let's Review.

# Multi-Indexing



**Multi-indexing** is the process of creating more than one index for a DataFrame.

# Multi-Indexing

---

Sometimes, one index is not enough for performing data lookups; more than one index is needed. For example, it is common to use multi-indexing when working with dates. This allows data to be accessed by year, month, and/or day.

|   | one       |           |           | two       |           |           |
|---|-----------|-----------|-----------|-----------|-----------|-----------|
|   | a         | b         | c         | a         | b         | c         |
| 0 | -1.401530 | 0.626666  | -0.165782 | -0.361173 | -1.139887 | -0.027251 |
| 1 | 1.201998  | -0.665546 | -0.554207 | 0.644199  | 0.572868  | -1.552404 |
| 2 | -1.201190 | -1.428929 | 1.226697  | 0.162548  | 1.481702  | 0.721526  |
| 3 | -1.622470 | 0.541475  | -0.482980 | -1.970389 | 1.974586  | 0.165966  |



# Instructor Demonstration

## Multi-indexing DataFrames



## Activity: Indexing Fever

In this activity, you will use hierarchical indexes to gain access to historical stock data. You will leverage Google Sheets to extract Google Finance data to perform data segmentation for a single ticker over multiple months in a year.

(Instructions sent via Slack.)

**Suggested Time:**  
15 Minutes





**Time's Up!** Let's Review.



Countdown timer

**15:00**

(with alarm)



# Concatenation



**Concatenation** is the process of joining one dataset with another.

# Concatenation

---

Pandas has a `concat` function that can be used to combine DataFrames.

DataFrames can be concatenated so that the records from two DataFrames are combined.

DataFrames can be combined by column so that the columns from one DataFrame are placed adjacent to columns from another DataFrame.





# Instructor Demonstration

## Concatenating DataFrames



## Activity:

# Mastering Concatenation

In this activity, you will combine multiple DataFrames using the `concat` function.

(Instructions sent via Slack.)

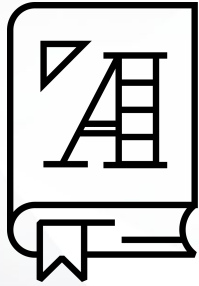
**Suggested Time:**  
15 Minutes





**Time's Up!** Let's Review.

# Standard Deviation and Risk

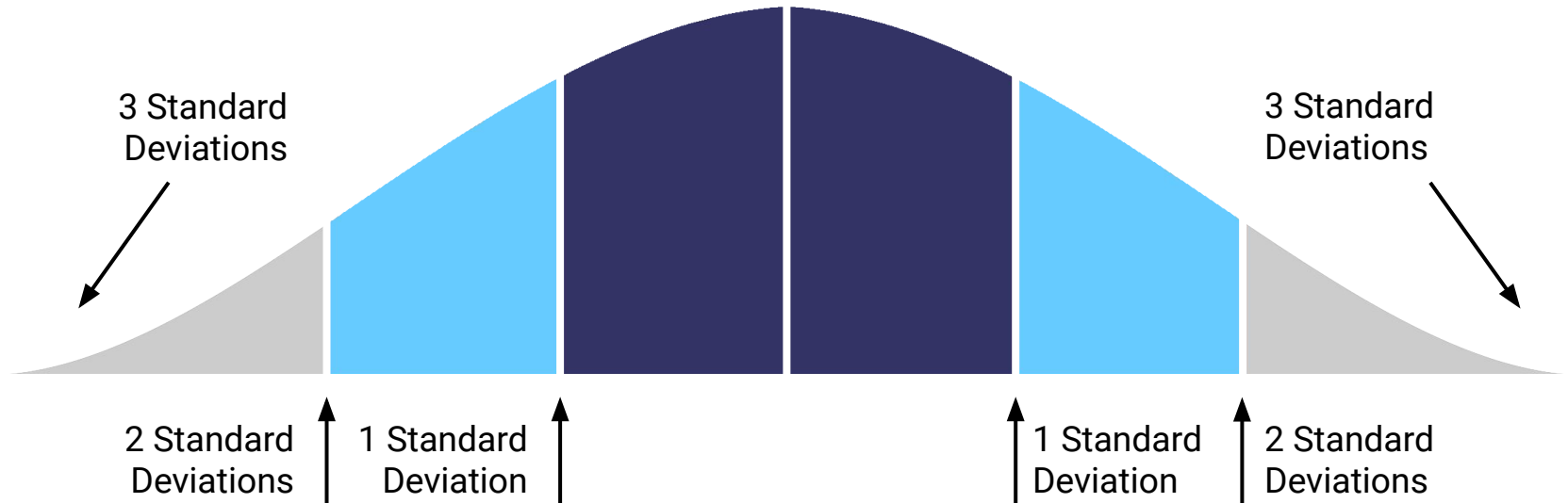


**Standard deviation** measures how dispersed a set of values are from their average.



# Standard Deviation

The `std` Pandas function is used to calculate standard deviation for a DataFrame. Standard deviation can be used to determine the risk associated with an investment. Standard deviation is also used to calculate how much returns have been distributed from the average.





The greater the standard deviation, the greater the risk and the potential for a greater payout.



# Instructor Demonstration

## Calculating Standard Deviation & Risk with Pandas

# Sharpe Ratios

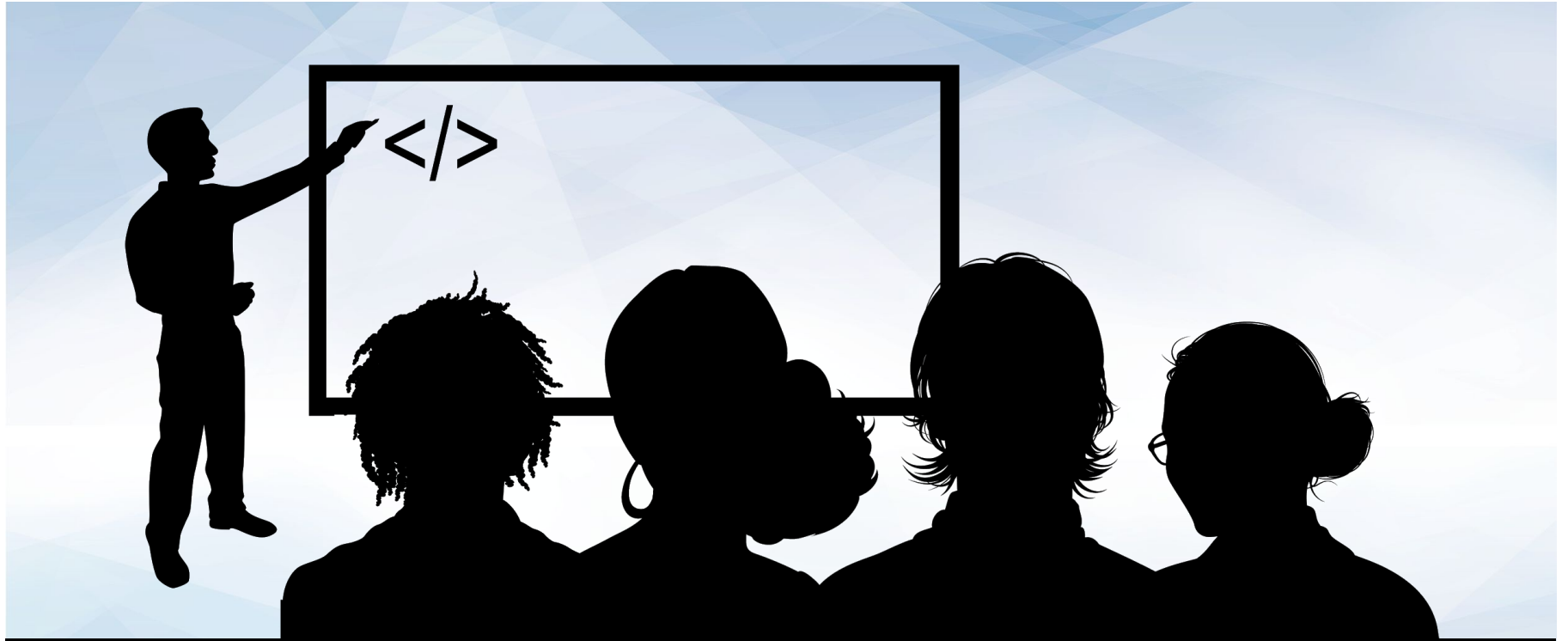
# Sharpe Ratios (Factoring risk into Returns)

---

Whereas standard deviation calculates how dispersed a set of values are, **Sharpe ratios** identify how much excess reward is associated with an investment after risk has been accounted for.

**Sharpe ratios** are calculated by dividing adjusted annualized average returns by annualized standard deviation.

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{O_p}$$



# Instructor Demonstration

## Calculating Sharpe Ratios



## Activity: Risky Business

It's time to put it all together. In this activity, you will prep data and use standard deviation and Sharpe ratios to analyze cryptocurrency portfolio performance.

(Instructions sent via Slack.)

**Suggested Time:**  
15 Minutes





**Time's Up!** Let's Review.



# Congratulations!

---

You just leveled up and acquired the following skills:



Data consolidation



Data segmentation



Data profiling and quality



Investment and portfolio risk analysis





Questions?

*The  
End*