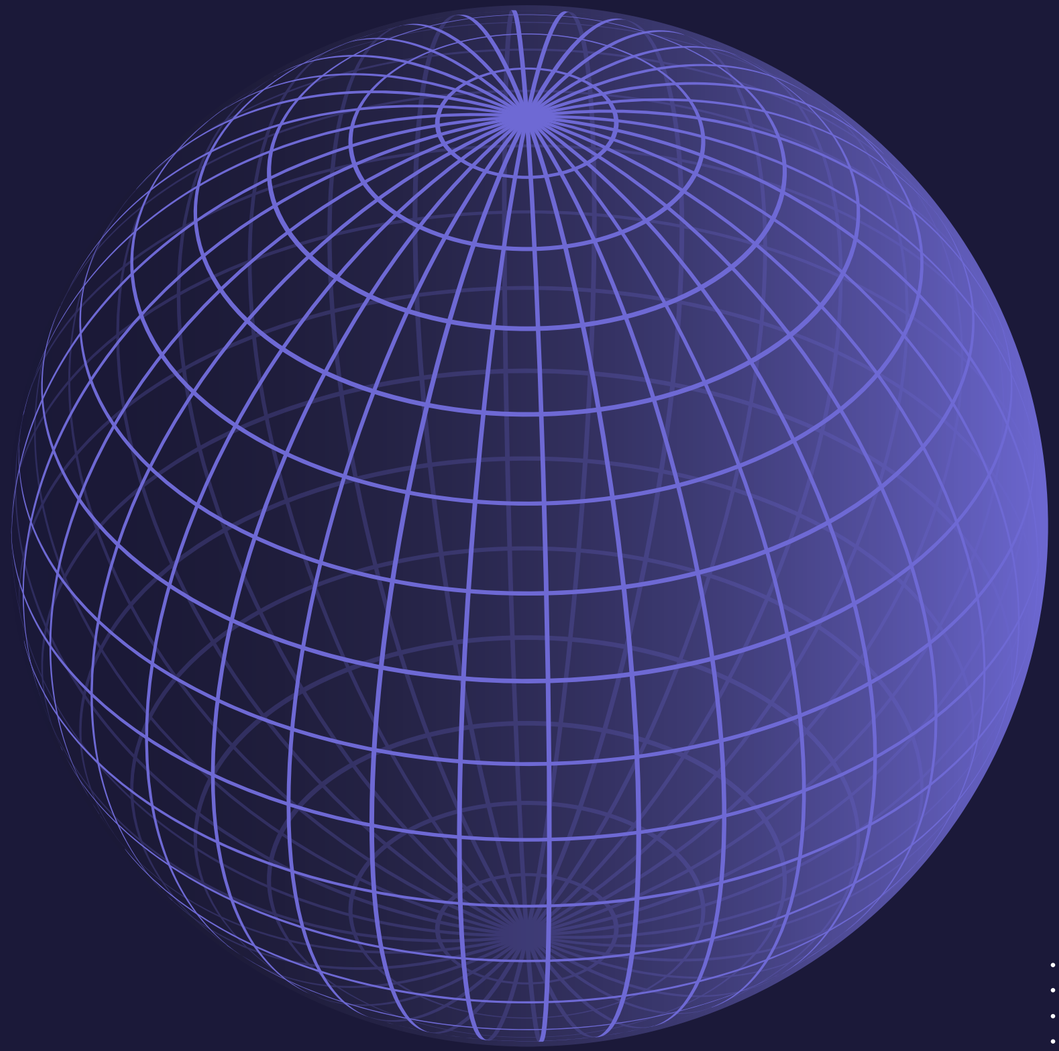
001

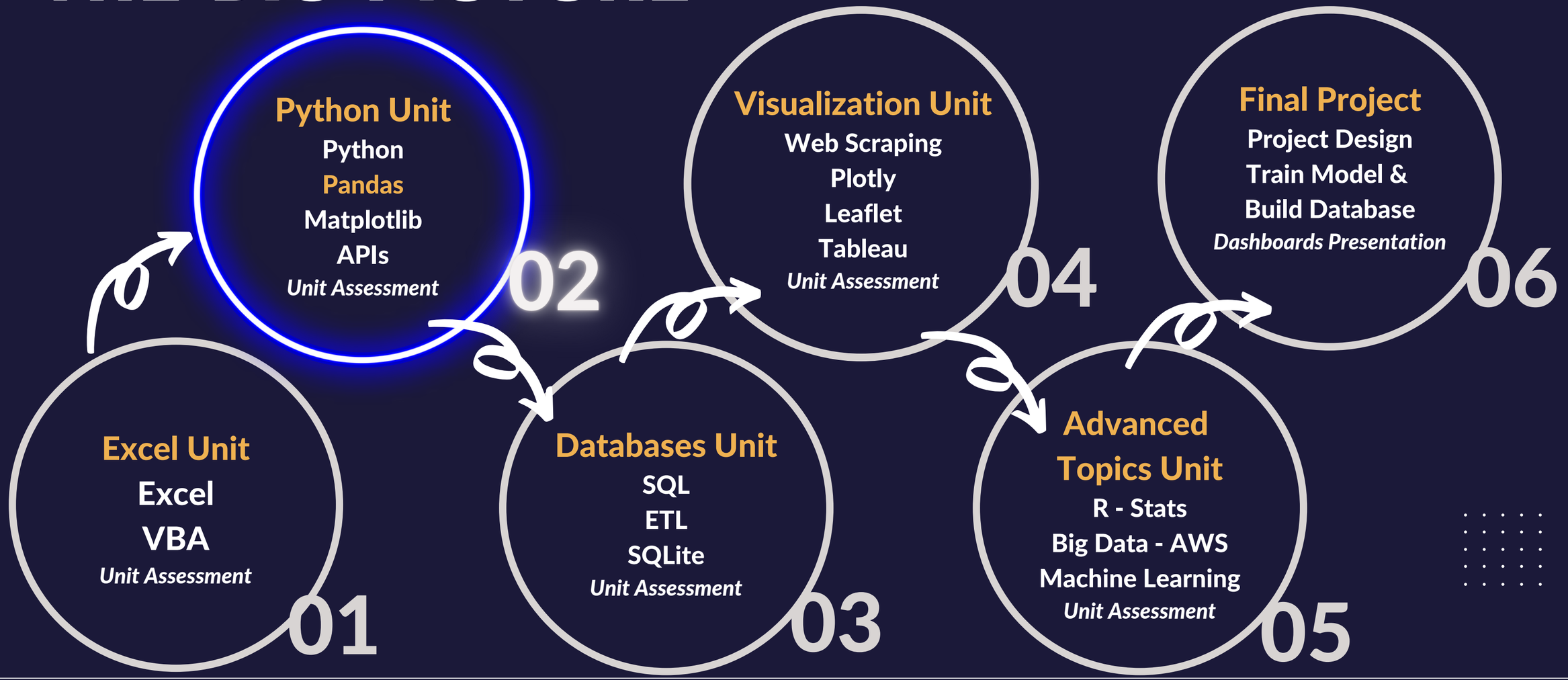
MODULE 4: PANDAS



DATA BOOTCAMP
UNIVERSITY OF KANSAS

UNIVERISTY OF KANSAS

THE BIG PICTURE



BOOT CAMP POINTERS

As you work through this module, remember the following:

01 **Learning Assistants**

02 **Office Hours**

03 **Tutoring Services**

THIS WEEK: PYTHON

By the end of this week, you'll know how to:

- Read an external CSV file into a DataFrame.
- Determine data types of row values in a DataFrame.
- Format and retrieve data from columns of a DataFrame.
- Merge, filter, slice, and sort a DataFrame..
- Apply the groupby() function to a DataFrame..
- Use multiple methods to perform a function on a DataFrame.
- Perform mathematical calculations on columns of a DataFrame or Series.

THIS WEEK'S CHALLENGE

PyCity Schools Challenge

Use Python and the Pandas library to analyze school district data and showcase trends in school performance.

Using the skills learned throughout the week, help a mock school board with their investigation by adjusting specific data.

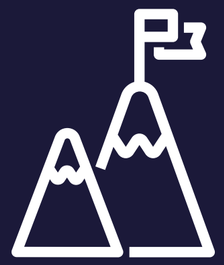
- **Deliverable 1**: Replace ninth-grade reading and math scores
- **Deliverable 2:** Repeat the school district analysis
- **Deliverable 3:** A written Analysis of the Election Audit (README.md)

WHAT IS PANDAS?

- Open-source Python Library
- High-performance data manipulation and analysis tool
- Powerful data structures

Fun Fact: ***The name Pandas is derived from the word Panel Data – an Multidimensional data concept in Econometrics.***

WHY IS PANDAS IMPORTANT?



- **Easiest and cleanest way to *analyze* data in the Python programming language**

CAREER CONNECTION

UNIVERSITY OF KANSAS

008

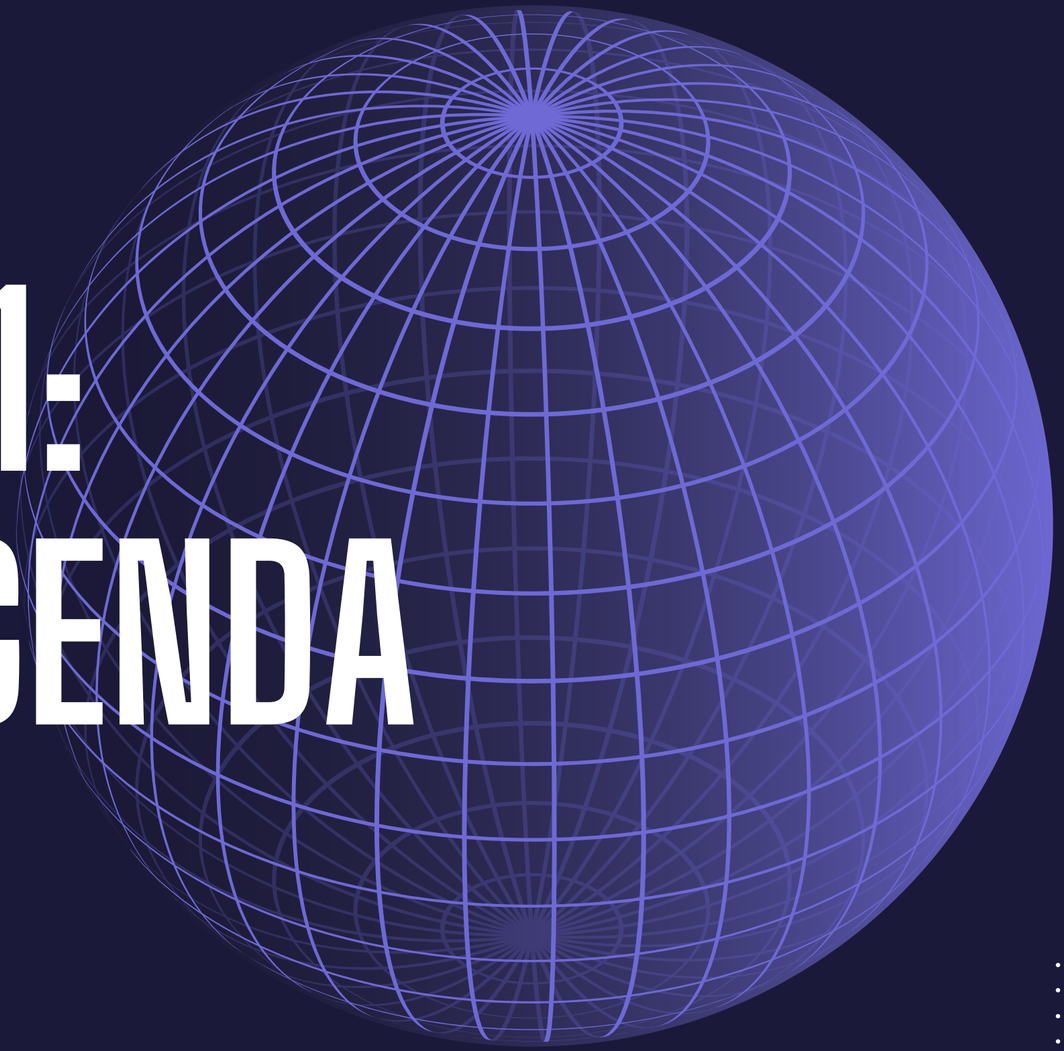
HOW TO SUCCEED THIS WEEK

UNIVERISTY OF KANSAS

QUICK TIP FOR SUCCESS

New syntax may not always be easy to remember, but don’t worry! The documentation you need is just a click away.

MODULE 4.1: TODAY'S AGENDA



DATA BOOTCAMP
UNIVERSITY OF KANSAS

TODAY'S AGENDA

By completing today's activities, you'll learn the following skills:

- **Anaconda Installation Check**
- **Series, Dataframes, & Dataframe Functions**
  - *Activity: Training Grounds*
- **Reading CSVs and Data Manipulation**
  - *Activity: Good Reads*
- **Merging Dataframes**
- **Formatting & Mapping**

  Group Activity: Formatting & Mapping

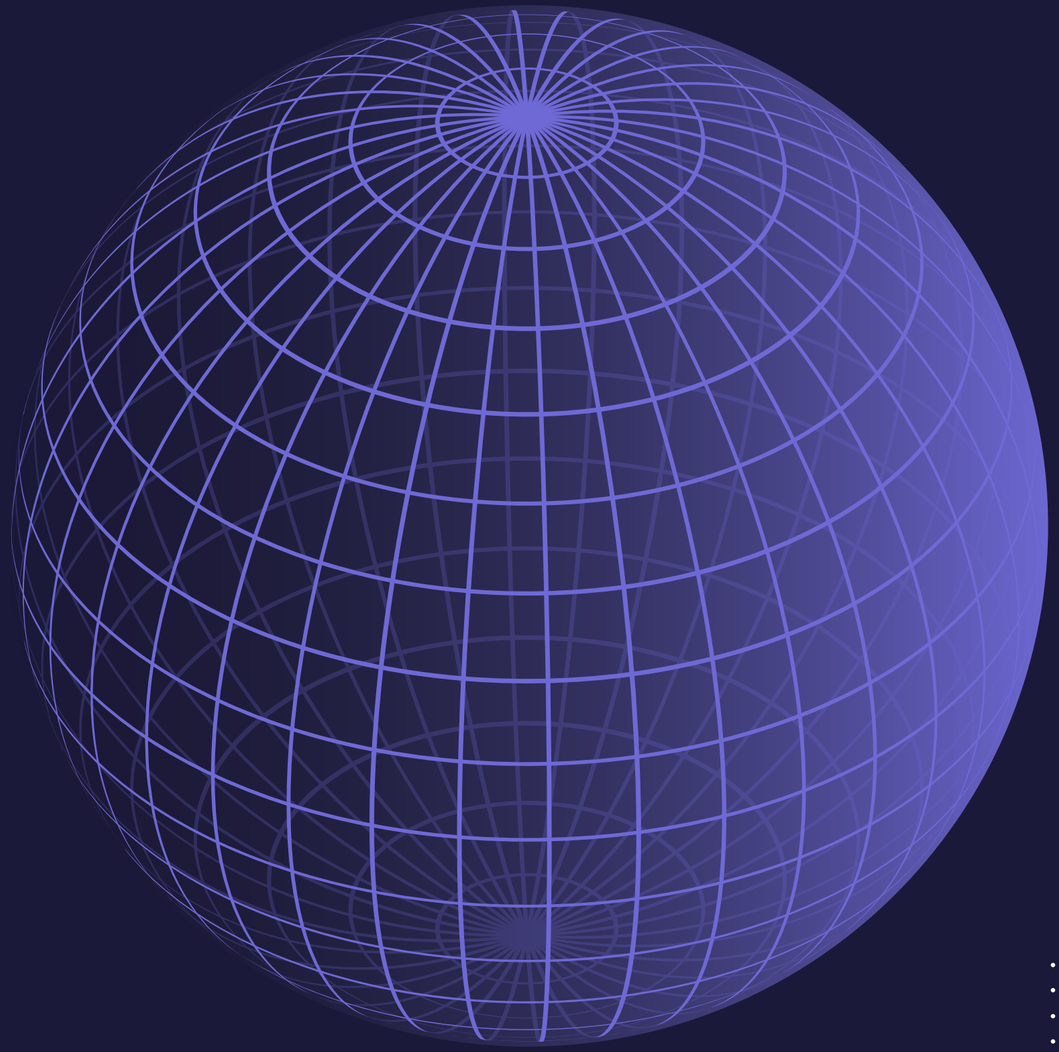
Make sure you've downloaded any relevant class files!

CHECK IN

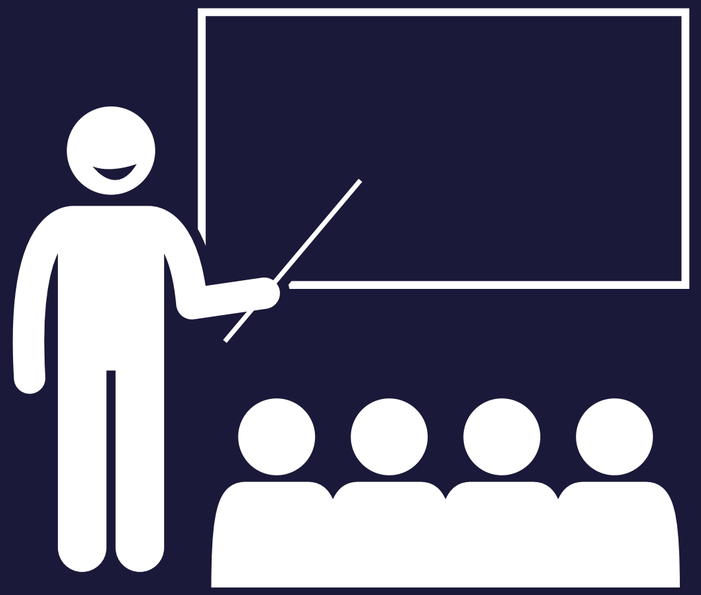
UNIVERSITY OF KANSAS

014

ANACONDA



UNIVERISTY OF KANSAS



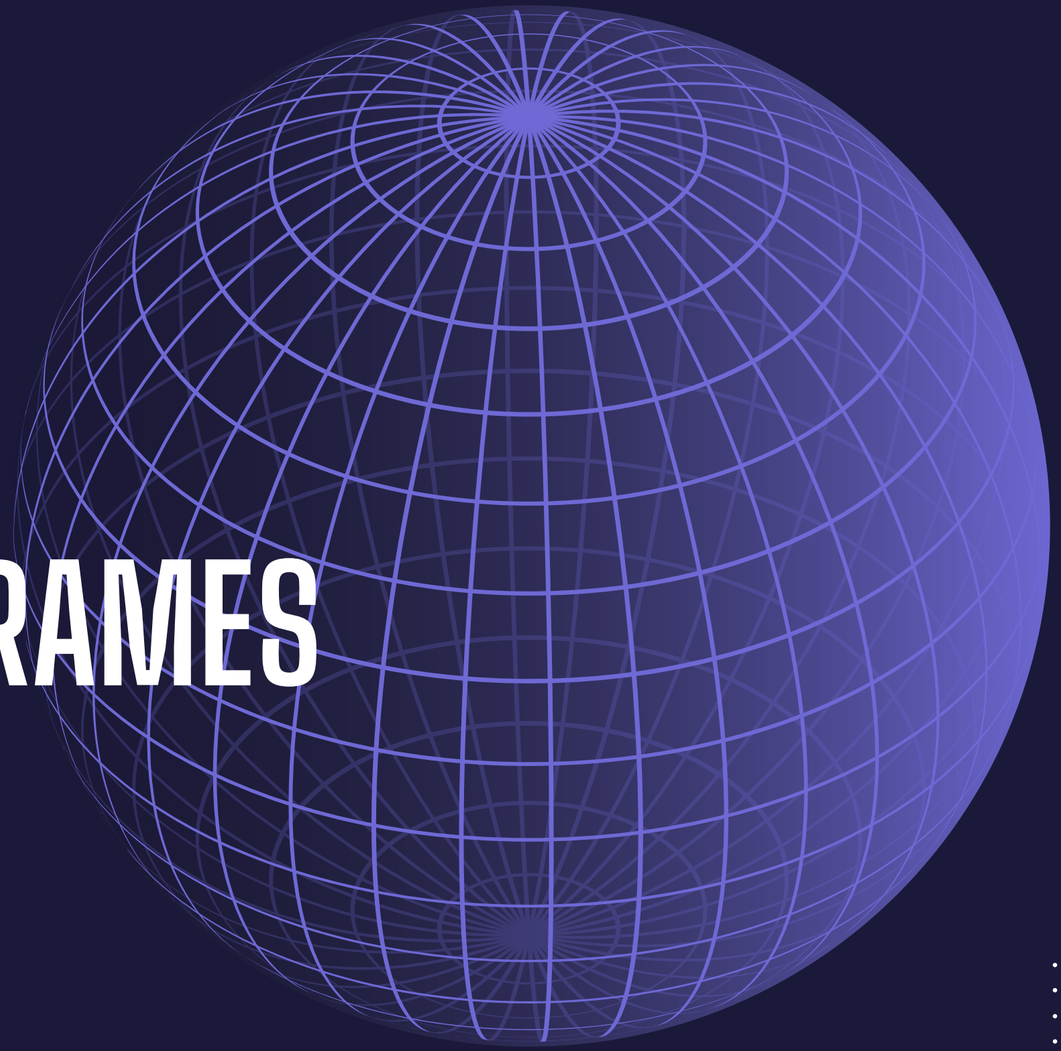
INSTRUCTOR DEMONSTRATION

Installation Check

UNIVERSITY OF KANSAS

014

PANDAS: SERIES & DATAFRAMES

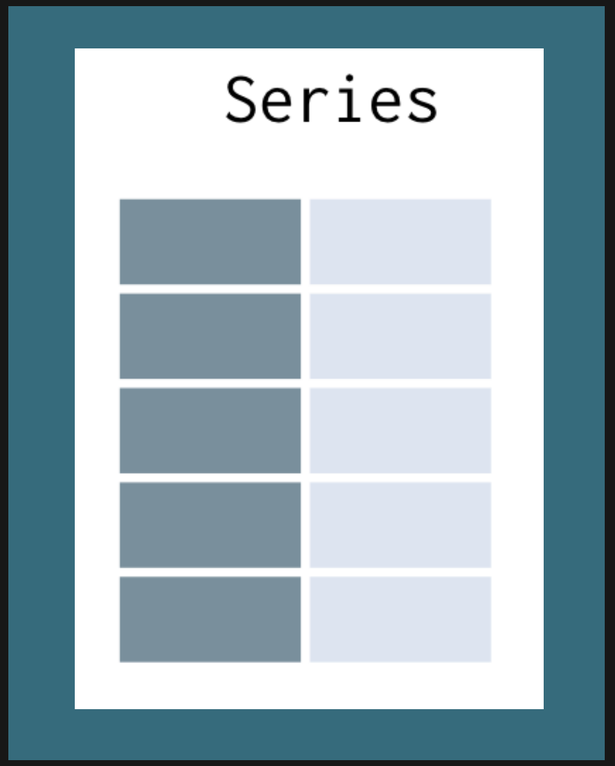


UNIVERISTY OF KANSAS

WHAT IS A SERIES IN THE PYTHON PANDAS LIBRARY?

A Series is a one-dimensional data structure.

It has a numeric index of the initial list, which acts as a key in a dictionary.



HOW TO CREATE PANDAS SERIES IN PYTHON

Import Pandas Library

First, import Pandas library running...

```
import pandas as pd
```

This method of import allows Pandas functions/methods to be called using the variable pd.

Create a Series

To create a Series, simply run...

```
pd.Series()
```

This is a function and you pass in a list within the parentheses.

Note that the index for the values within the Series will be the numeric index of the initial list.

```
0                          UCLA
1                    UC Berkeley
2                      UC Irvine
3    University of Central Florida
4                Rutgers University
```

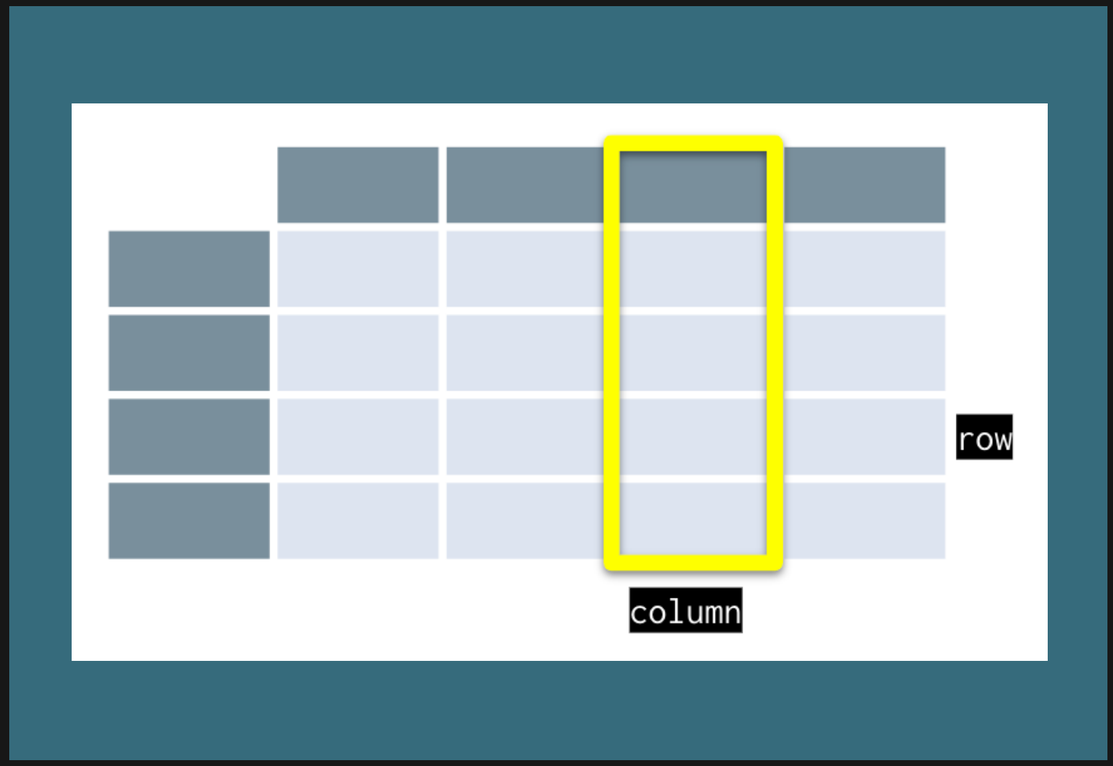
WHAT IS A DATAFRAME IN THE PYTHON PANDAS LIBRARY?

A DataFrame is a two-dimensional, labeled data structure.

It has rows and columns of potentially different data types such as strings, integers, and floats.

Data is aligned in a table, much like a spreadsheet.

Series is the data structure for a single column of a DataFrame



HOW TO CREATE PANDAS DATAFRAME IN PYTHON

*THERE ARE MULTIPLE WAYS TO CREATE DATAFRAMES

List of Dictionaries

Use the function...

pd.DataFrame()

and pass in a list of dictionaries.

Each dictionary will represent a new row where the keys become column headers, and the values will be placed inside the table.

Dictionary of Lists

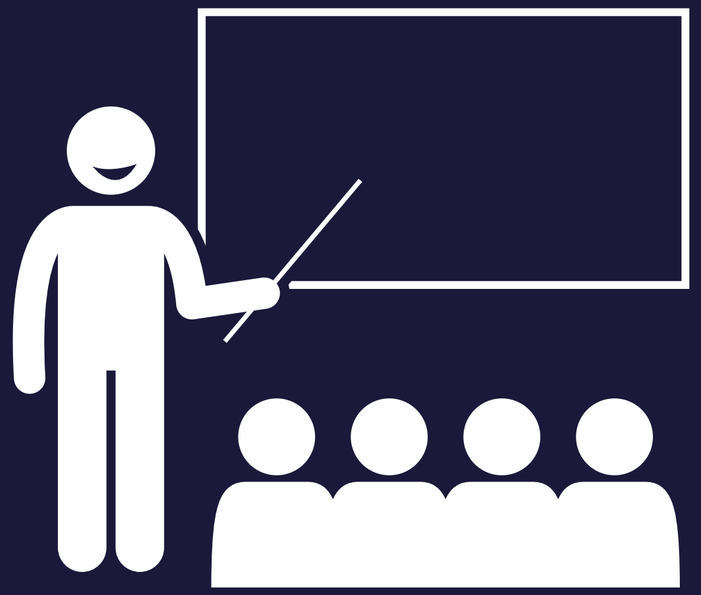
Use the function...

pd.DataFrame()

and pass in a dictionary of lists.

The keys of the dictionary will be the column headers, and the listed values will be placed into their respective rows.

	Title	Year Released
0	The Godfather	1972
1	The Shawshank Red emption	1994
2	Pulp Fiction	1994
3	The Lord of the Rings: The Return of the King	2003



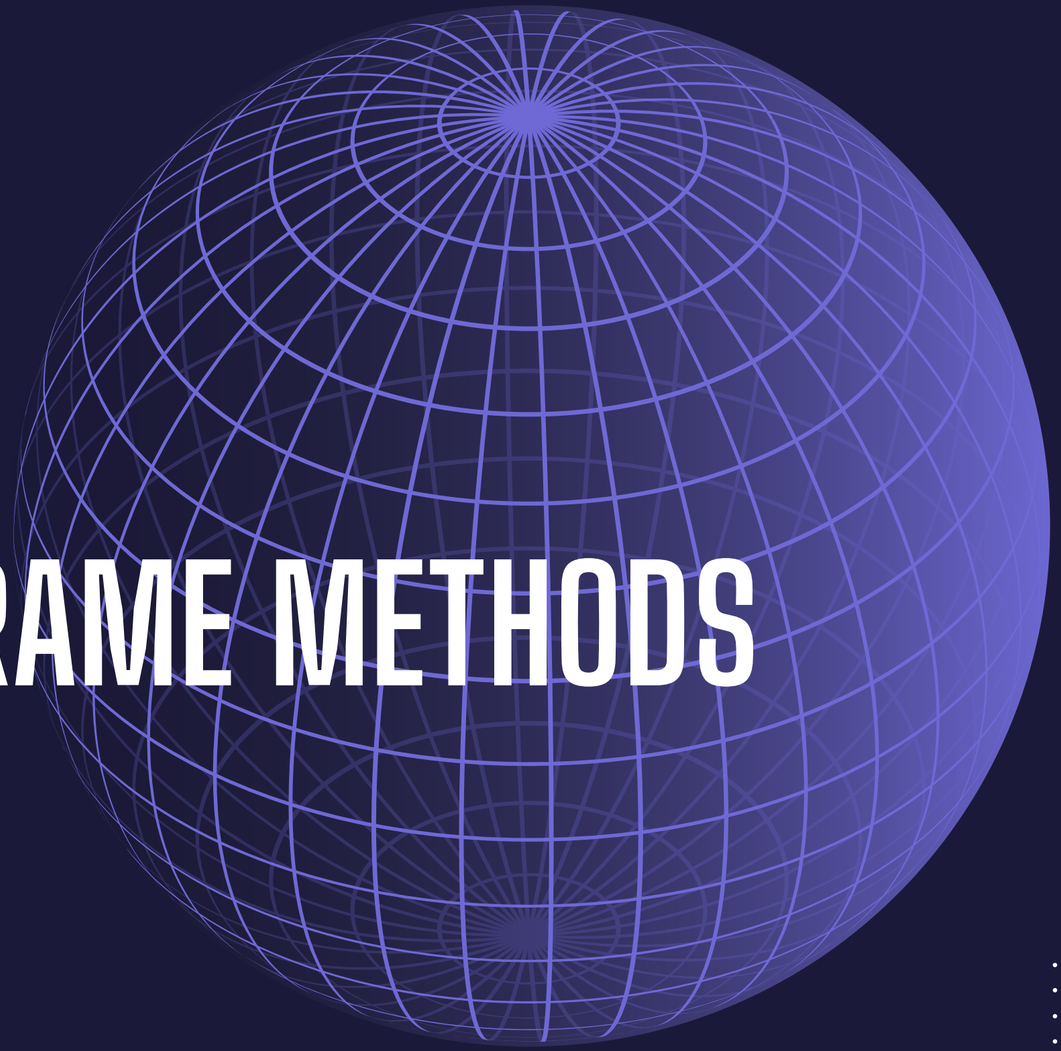
INSTRUCTOR DEMONSTRATION

Creating Series and Dataframes

UNIVERSITY OF KANSAS

014

PANDAS: BUILT-IN DATAFRAME METHODS



UNIVERISTY OF KANSAS

BUILT-IN PANDAS HEAD() METHOD

What does it do?

Shows only first 5 rows of a Pandas Dataframe. This number can be increased or decreased by placing an integer within the parentheses i.e. head(10) <- This will show the first 10 rows in the Dataframe

Why use it?

Allows you look at a small portion of a much larger table, thus you allowing you to make informed changes without having to search through the entire dataset.

```
In [3]: # Use Pandas to read data
        data_file_df = pd.read_csv(data_file)
        data_file_df.head()
```

Out[3]:

	id	First Name	Last Name	Gender	Amount
0	1	Todd	Lopez	M	8067.7
1	2	Joshua	White	M	7330.1
2	3	Mary	Lewis	F	16335.0
3	4	Emily	Burns	F	12460.8
4	5	Christina	Romero	F	15271.9

BUILT-IN PANDAS DESCRIBE() METHOD

What does it do?

It will print out a DataFrame containing summary statistics on the table and its columns

Why use it?

Shows what other data functions can be performed on a DataFrame or Series.

```
In [4]: # Display a statistical overview of the DataFrame
        data_file_df.describe()
```

Out[4]:

	id	Amount
count	1000.000000	1000.000000
mean	500.500000	10051.323600
std	288.819436	5831.230806
min	1.000000	3.400000
25%	250.750000	4854.875000
50%	500.500000	10318.050000
75%	750.250000	15117.425000
max	1000.000000	19987.400000

BUILT-IN PANDAS UNIQUE() METHOD

What does it do?

It looks into a Series and returns all distinct values

Why use it?

It will list out all of the unique values stored within a column

```
In [9]: # The unique method shows every element of the series that appears only once
        unique = data_file_df["Last Name"].unique()
        unique

Out[9]: array(['Lopez', 'White', 'Lewis', 'Burns', 'Romero', 'Andrews', 'Baker',
               'Diaz', 'Burke', 'Richards', 'Hansen', 'Tucker', 'Wheeler',
               'Turner', 'Reynolds', 'Carpenter', 'Scott', 'Ryan', 'Marshall',
               'Fernandez', 'Olson', 'Riley', 'Woods', 'Wells', 'Gutierrez',
               'Harvey', 'Ruiz', 'Lee', 'Welch', 'Cooper', 'Nichols', 'Murray',
               'Gomez', 'Green', 'Jacobs', 'Griffin', 'Perry', 'Dunn', 'Gardner',
               'Gray', 'Walker', 'Harris', 'Lawrence', 'Black', 'Simpson', 'Sims',
               'Weaver', 'Carr', 'Owens', 'Stephens', 'Butler', 'Matthews', 'Cox',
               'Brooks', 'Austin', 'Moore', 'Hunter', 'Cunningham', 'Lane',
               'Montgomery', 'Vasquez', 'Freeman', 'Hernandez', 'Alexander',
               'Pierce', 'Mcdonald', 'Kelly', 'Foster', 'Bell', 'Johnson',
               'Bowman', 'Porter', 'Wood', 'Reid', 'Willis', 'Bishop',
               'Washington', 'Gonzales', 'Davis', 'Martinez', 'Martin', 'Long',
               'Howell', 'Hawkins', 'Knight', 'Price', 'Day', 'Bailey', 'Flores',
               'Young', 'Evans', 'Cruz', 'Chavez', 'Barnes', 'Coleman', 'Burton',
               'Clark', 'Carter', 'Franklin', 'Ellis', 'Miller', 'Allen', 'Mason',
               'Patterson', 'Stevens', 'Kim', 'Kelley', 'Robinson', 'Hughes',
               'Morgan', 'Dean', 'Stewart', 'Murphy', 'Fox', 'Simmons',
               'Thompson', 'Fuller', 'Peterson', 'Hanson', 'Wright', 'Reed',
               'Graham', 'Parker', 'Boyd', 'Taylor', 'Greene', 'George', 'Mills',
               'Duncan', 'Hill', 'Jordan', 'Stanley', 'Hall', 'James', 'Stone',
               'Warren', 'Fowler', 'Williamson', 'Lynch', 'Harper', 'Little',
               'Nguyen', 'Morrison', 'Ramirez', 'Howard', 'Watkins', 'Robertson',
               'Powell', 'Sanchez', 'Sanders', 'Grant', 'Ross', 'Mitchell',
               'Henderson', 'Rose', 'Perez', 'Berry', 'Watson', 'Gordon',
               'Morales', 'Arnold', 'Morris', 'Crawford', 'Smith', 'Medina',
               'Alvarez', 'Collins', 'Rodriguez', 'Mccoy', 'Bennett',
               'Richardson', 'Chapman', 'Johnston', 'Gilbert', 'Ford', 'Russell',
               'Nelson', 'Castillo', 'Cole', 'Rice', 'Payne', 'Frazier', 'Webb',
               'Armstrong', 'Wilson', 'Garza', 'Garrett', 'Spencer', 'Peters',
               'Sullivan', 'Brown', 'Williams', 'Gonzalez', 'Palmer', 'Fields',
               'Snyder', 'Jackson', 'Edwards', 'Anderson', 'Cook', 'Ramos',
               'Harrison', 'Lawson', 'Banks', 'Wallace', 'Ortiz', 'Gibson',
               'Reyes', 'Shaw', 'Ward', 'Perkins', 'Bradley', 'Rivera', 'Jenkins',
               'Hart', 'Phillips', 'Garcia', 'Fisher', 'King', 'Larson', 'Hunt',
               'Jones', 'Hudson', 'Myers', 'Hayes', 'Dixon', 'Schmidt', 'Moreno',
               'Rogers', 'Thomas', 'Meyer', 'Daniels', 'Bryant', 'Henry',
               'Campbell', 'Ferguson', 'Oliver', 'Ray', 'Carroll', 'Wagner',
               'Kennedy', 'Holmes'], dtype=object)
```

BUILT-IN PANDAS VALUE_COUNTS() METHOD

What does it do?

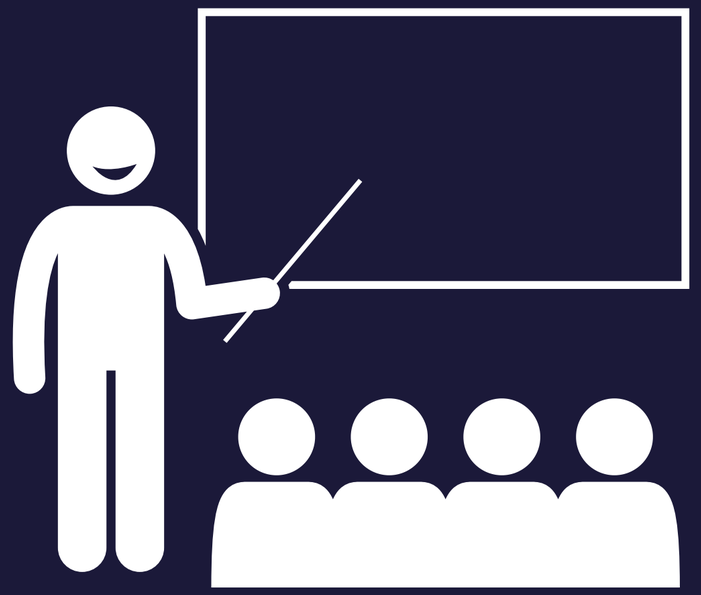
It returns a list of all unique values within a series AND also counts how many times a unique value appears

Why use it?

You can count the number of times a unique value appears in a column

```
In [10]: # The value_counts method counts unique values in a column
         count = data_file_df["Gender"].value_counts()
         count

Out[10]: M    515
         F    485
         Name: Gender, dtype: int64
```



INSTRUCTOR DEMONSTRATION

Built-in Dataframe Methods

UNIVERSITY OF KANSAS

ACTIVITY:
TRAINING GROUNDS

In this activity, you will take a large DataFrame consisting of 200 rows, analyze it using some data functions, and then add a new column.

Suggested Time:
15 minutes

UNIVERSITY OF KANSAS

INSTRUCTIONS: TRAINING GROUNDS

Using the DataFrame provided, perform all of the following actions...

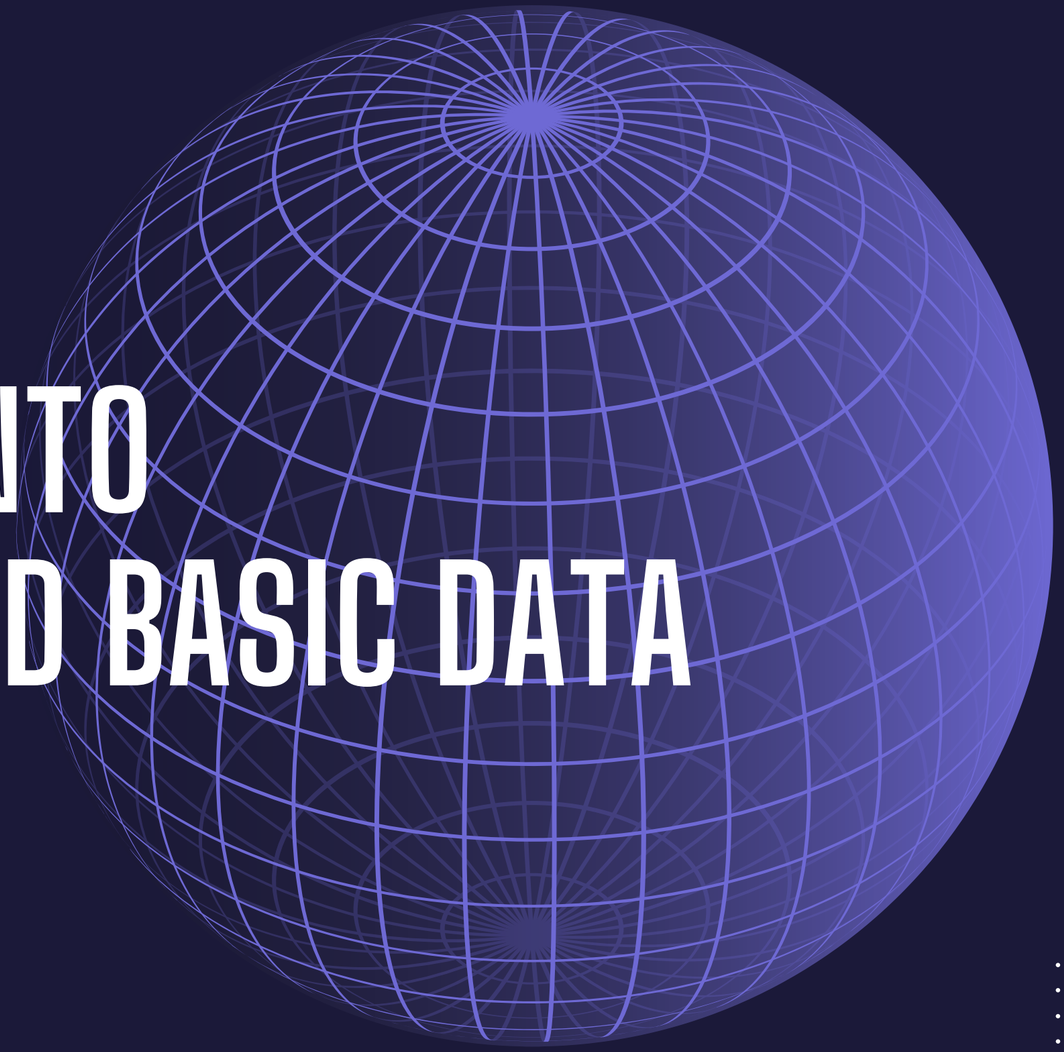
- Provide a simple, analytical overview of the dataset's numeric columns
- Collect all of the names of the trainers within the dataset
- Figure out how many students each trainer has
- Find the average weight of the students at the gym
- Find the combined weight of all of the students at the gym
- Convert the "Membership (Days)" column into weeks and then add this new series into the DataFrame

CHECK IN

UNIVERSITY OF KANSAS

014

PANDAS: READING FILES INTO DATAFRAMES AND BASIC DATA CLEANING

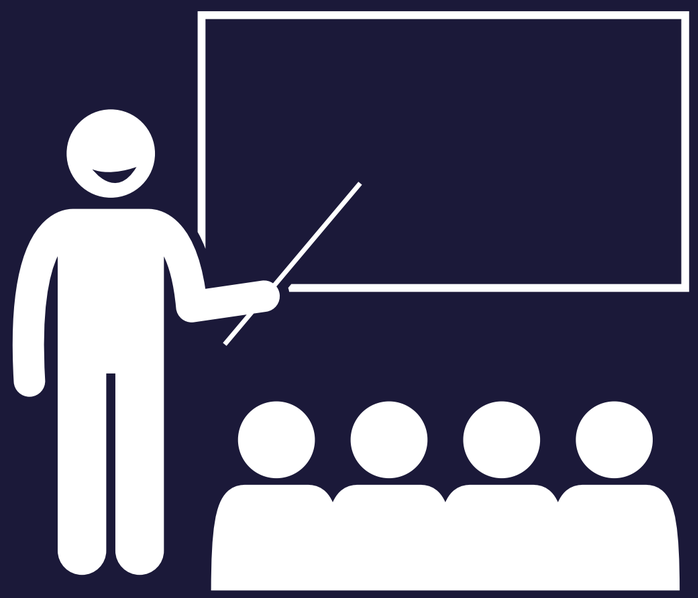


UNIVERISTY OF KANSAS

READING DATA IN PANDAS

If you are given data that is in an .xlsx or .csv format, how do you create a DataFrame?

- Import the xlsx file using using pd.read_excel()
- Import the csv file using pd.read_csv()



INSTRUCTOR DEMONSTRATION

Reading CSVs

UNIVERSITY OF KANSAS

ACTIVITY:
GOOD READS CSV

In this exercise, students will take a large CSV of books, read it into Jupyter Notebook using Pandas, and clean up the columns.

Suggested Time:
15 minutes

UNIVERSITY OF KANSAS

INSTRUCTIONS: GOOD READS CSV

- Read in the GoodReads CSV using Pandas with utf-8 encoding.
- Get the date types of each column.
- Get a list of all columns within the DataFrame.
- Remove unnecessary columns from the DataFrame so that only the following columns remain: isbn, original_publication_year, original_title, authors, ratings_1, ratings_2, ratings_3, ratings_4, and ratings_5
- Rename the columns to the following: ISBN, Publication Year, Original Title, Authors, One Star Reviews, Two Star Reviews, Three Star Reviews, Four Star Reviews, and Five Star Reviews

ACTIVITY: GOOD READS CSV

The initial CSV file is encoded using UTF-8...

...so it should be read using this encoding as well to ensure there are no strange characters hidden within the dataset.

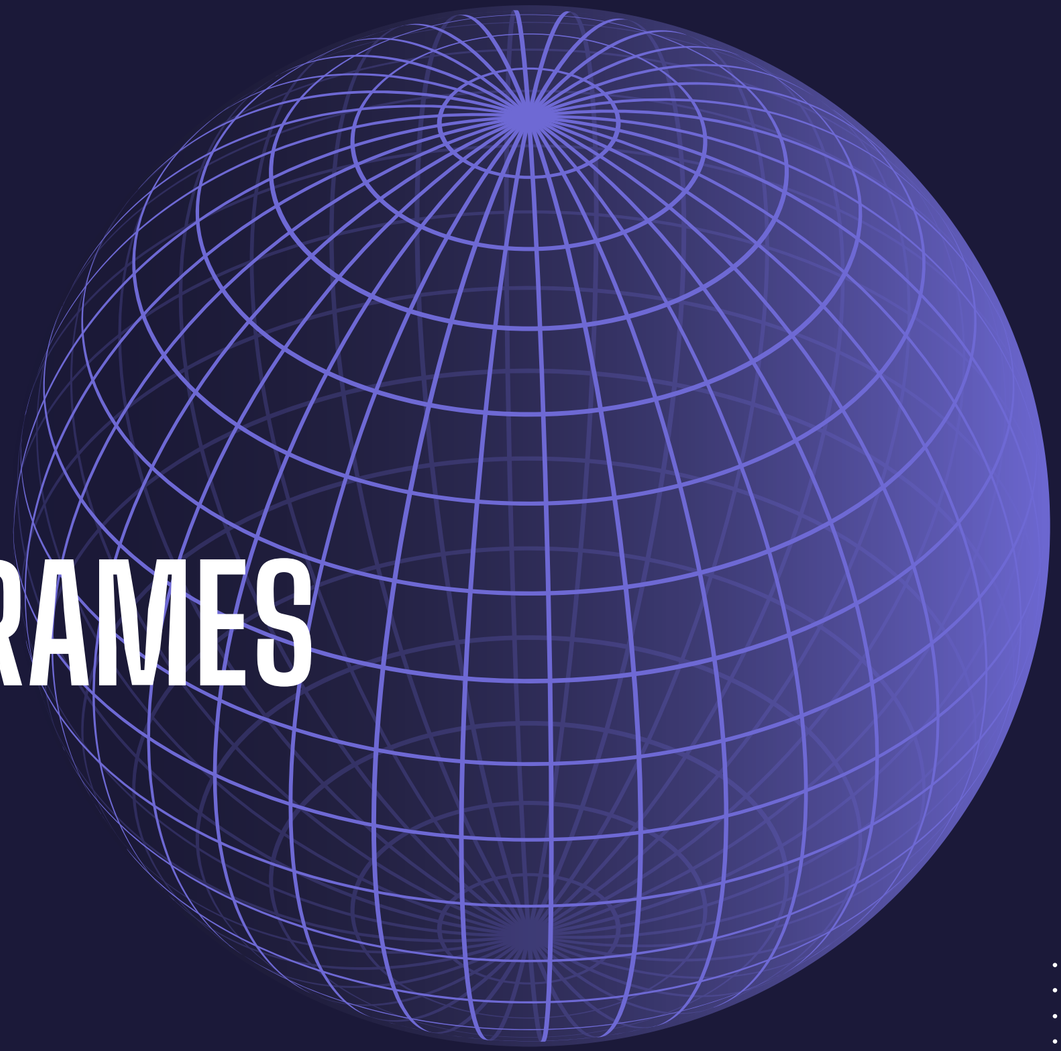
There are a lot of columns that are being modified within this code...

...so it is useful to get all the columns in an array using the .columns attribute.

This helps to make sure that all references are made accurately so as to avoid any potential errors.

014

PANDAS:
MERGING DATAFRAMES



UNIVERISTY OF KANSAS

WHAT IS MERGING?

Merging is the process of combining two tables based on shared data.

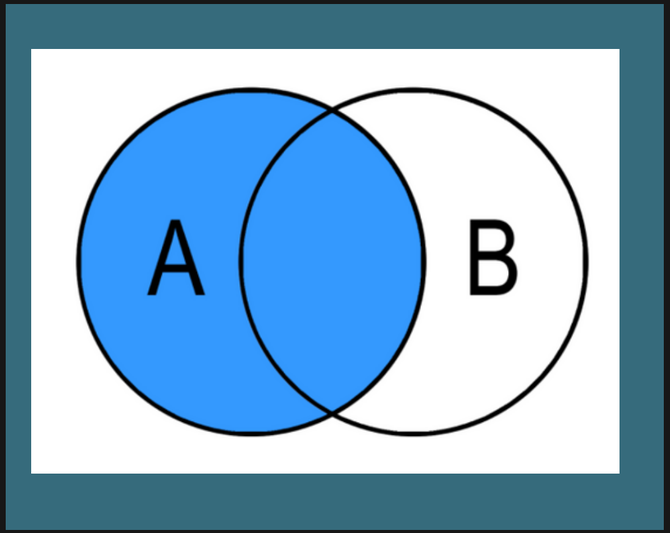
WHY MIGHT YOU NEED TO MERGE DATAFRAMES?

- Sometimes an analyst will receive data split across multiple tables and sources
- Working across multiple tables is error prone and confusing
- Shared data can be an identical column in both tables or a shared index
- In Pandas, we can merge separate DataFrames using the pd.merge() method

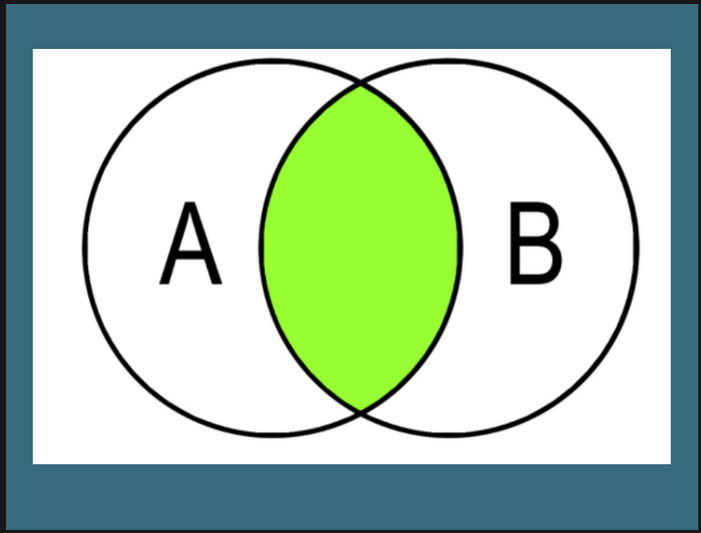
JOINS ARE VERY IMPORTANT IN DATA SCIENCE!

When merging data tables, joins tell the program what data to keep.

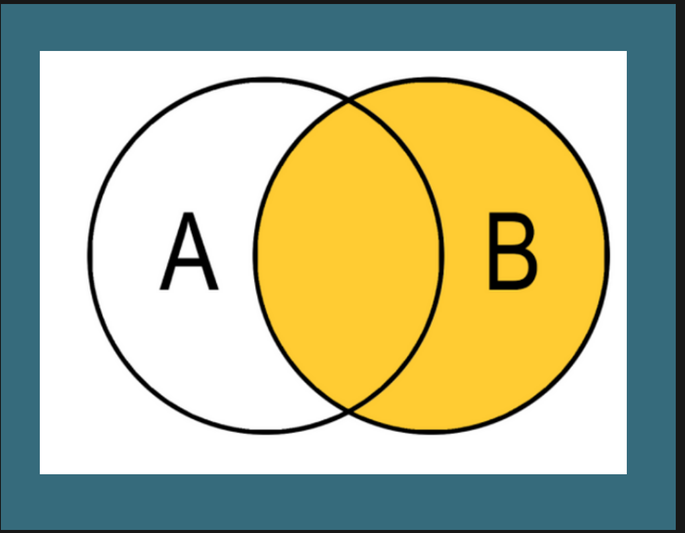
Besides outer joins, there are three other common joins:



Left Outer Join:
All rows from table A, even if they do not exist in table B



Inner Join:
Fetch the results that exist in both tables



Right Outer Join:
All rows from table B, even if they do not exist in table A

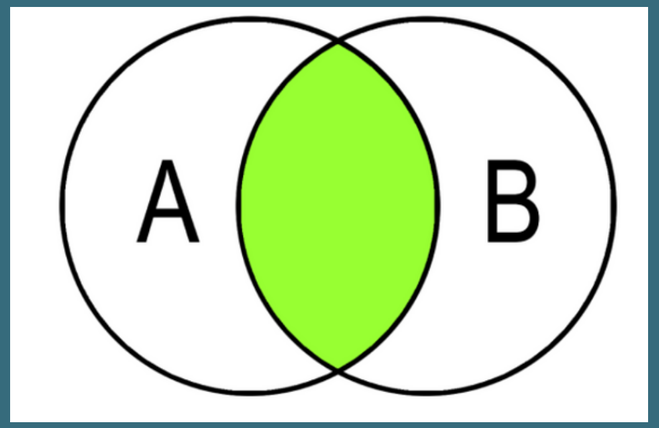
MERGING DATAFRAMES: INNER JOIN

Inner joins are the default means through which DataFrames are combined using the pd.merge() method and will only return data whose values match. Rows that do not include matching data will be dropped from the combined DataFrame.

```
In [6]: # Merge two dataframes using an inner join (default).
        merge_df = pd.merge(info_df, items_df, on="customer_id")
        merge_df
```

Out[6]:

	customer_id	name	email	item	cost
0	112	John	jman@gmail	chips	4.5
1	403	Kelly	kelly@aol.com	soda	3.0
2	999	Sam	sports@school.edu	Laptop	900.0
3	543	April	April@yahoo.com	TV	600.0



Inner Join:
Fetch the results that exist in both tables

MERGING DATAFRAMES: OUTER JOIN

Outer joins will combine the DataFrames regardless of whether any of the rows match and must be declared as a parameter within the pd.merge() method using the syntax how="outer".

```
In [5]: # Merge two dataframes using an outer join
        merge_df = pd.merge(info_df, items_df, on="customer_id", how="outer")
        merge_df
```

Out[5]:

	customer_id	name	email	item	cost
0	112	John	jman@gmail	chips	4.5
1	403	Kelly	kelly@aol.com	soda	3.0
2	999	Sam	sports@school.edu	Laptop	900.0
3	543	April	April@yahoo.com	TV	600.0
4	123	Bobbo	HeyImBobbo@msn.com	NaN	NaN
5	654	NaN	NaN	Cooler	150.0

ANY ROWS THAT DO NOT INCLUDE MATCHING DATA WILL HAVE THE VALUES WITHIN REPLACED WITH **NAN** INSTEAD.

MERGING DATAFRAMES: RIGHT AND LEFT JOINS

These joins will protect the data contained within one DataFrame, like an outer join does, while also dropping the rows with null data from the other DataFrame.

```
In [6]: # Merge two dataframes using a left join
        merge_df = pd.merge(info_df, items_df, on="customer_id", how="left")
        merge_df
```

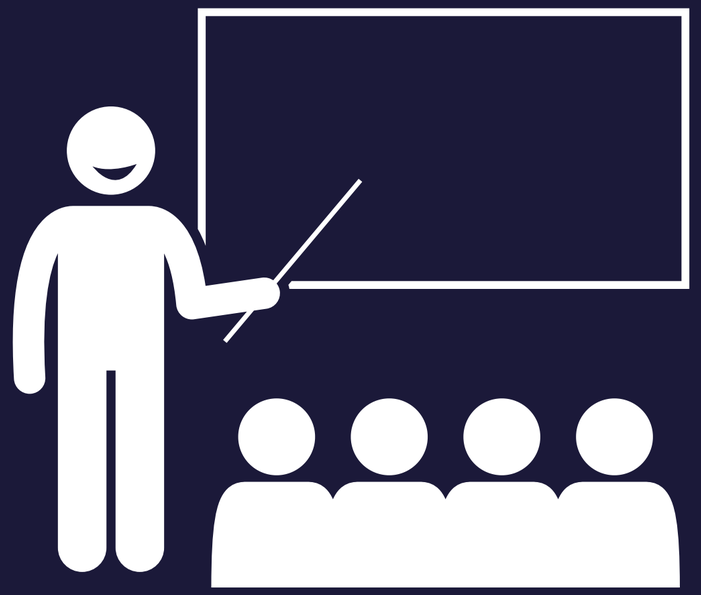
Out[6]:

	customer_id	name	email	item	cost
0	112	John	jman@gmail	chips	4.5
1	403	Kelly	kelly@aol.com	soda	3.0
2	999	Sam	sports@school.edu	Laptop	900.0
3	543	April	April@yahoo.com	TV	600.0
4	123	Bobbo	HeyImBobbo@msn.com	NaN	NaN

```
In [7]: # Merge two dataframes using a right join
        merge_df = pd.merge(info_df, items_df, on="customer_id", how="right")
        merge_df
```

Out[7]:

	customer_id	name	email	item	cost
0	112	John	jman@gmail	chips	4.5
1	403	Kelly	kelly@aol.com	soda	3.0
2	999	Sam	sports@school.edu	Laptop	900.0
3	543	April	April@yahoo.com	TV	600.0
4	654	NaN	NaN	Cooler	150.0



INSTRUCTOR DEMONSTRATION

Merging Data

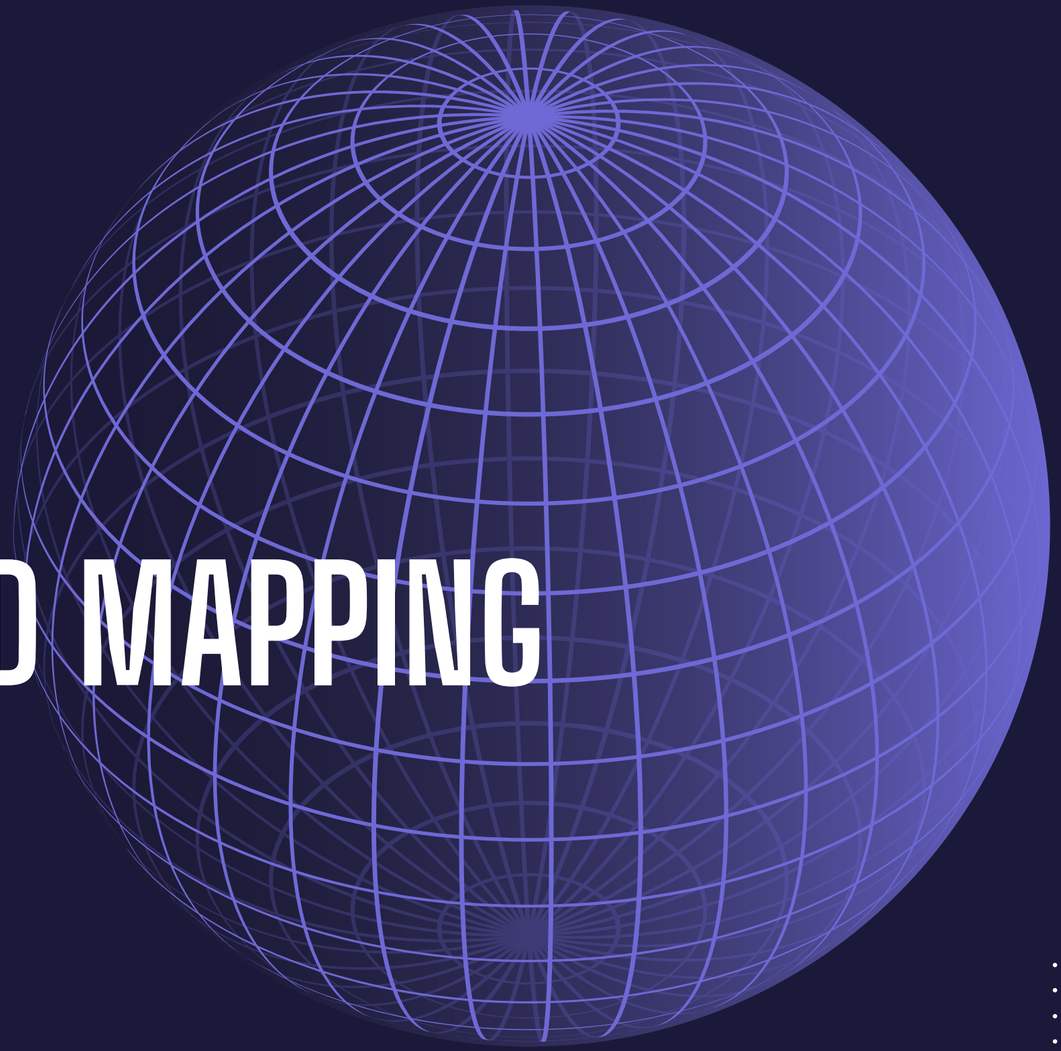
UNIVERSITY OF KANSAS

CHECK IN

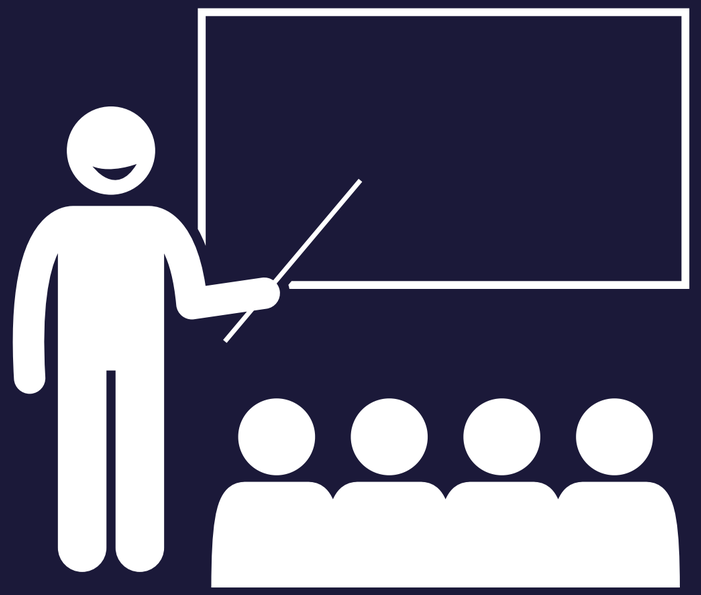
UNIVERSITY OF KANSAS

014

PANDAS: FORMATTING AND MAPPING



UNIVERISTY OF KANSAS



INSTRUCTOR DEMONSTRATION

Formatting & Mapping

UNIVERSITY OF KANSAS

GROUP ACTIVITY:
FORMATTING & MAPPING

In this exercise, you will read sales data into a DataFrame and are asked to format the columns that are int64 or float64 data types with comma notation,
a dollar sign, and to two decimal places.

Suggested Time:
15 minutes

UNIVERSITY OF KANSAS

INSTRUCTIONS: FORMATTING & MAPPING

- Read in the sales_data.csv using Pandas.
- Get the data types of each column.
- Get a list of all columns within the DataFrame.
- Use the map() and .format methods to format the following columns:
  - Units Sold with comma notation
  - Unit Price, Unit Cost, Total Revenue, Total Cost, and Total Profit with a dollar sign, comma notation, and to two decimal places.

CHECK IN

UNIVERSITY OF KANSAS

SUMMARY

- The pd.DataFrame(), read_csv(), and head() methods were covered in Lessons 4.4.3.
- The pd.merge() method was covered in Lesson 4.7.1.
- The unique() function was covered in Lesson 4.7.3.
- The .format and .map() methods were covered in Lesson 4.7.8.
- The value_counts() method was covered in Lesson 4.8.2.
- The describe() method was covered in Lesson 4.11.1.
- The rename() function and the .copy() method are covered in later modules.