Analytics Jumpstart

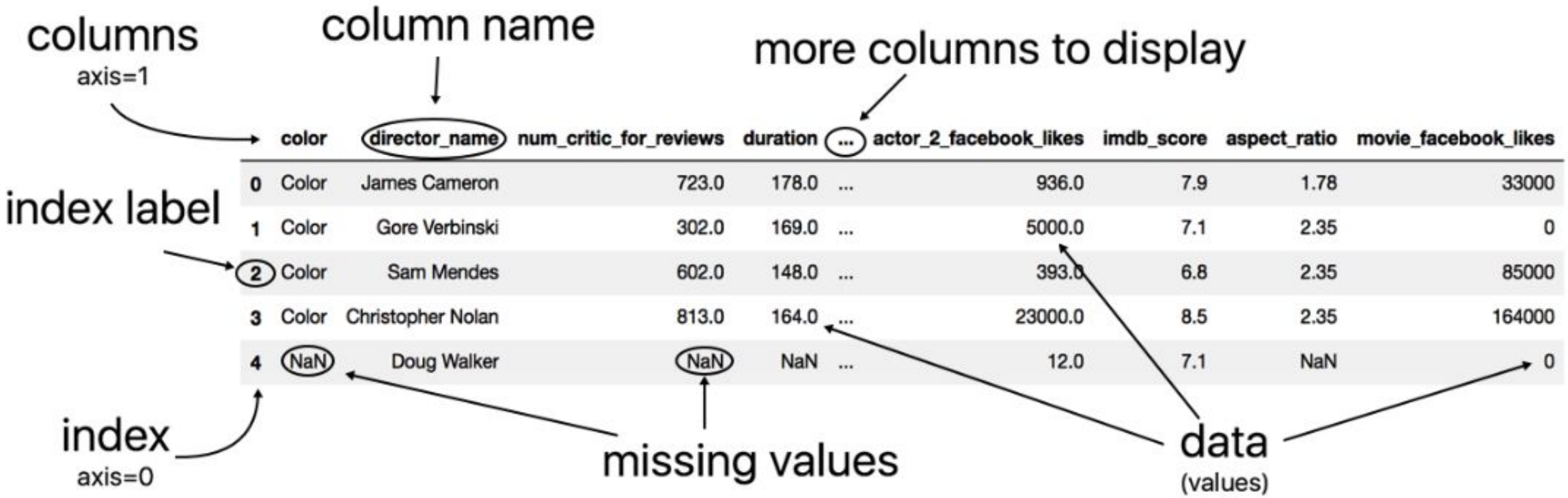
Intro to commonly used pandas methods

Nashville Software School

Goals for today

- Learn some pandas methods
- Work on coding tasks
- Use the pandas API to understand methods and their signatures

The Anatomy Of A Dataframe



	color	director_name	num_critic_for_reviews	duration	...	actor_2_facebook_likes	imdb_score	aspect_ratio	movie_facebook_likes
0	Color	James Cameron	723.0	178.0	...	936.0	7.9	1.78	33000
1	Color	Gore Verbinski	302.0	169.0	...	5000.0	7.1	2.35	0
2	Color	Sam Mendes	602.0	148.0	...	393.0	6.8	2.35	85000
3	Color	Christopher Nolan	813.0	164.0	...	23000.0	8.5	2.35	164000
4	NaN	Doug Walker	NaN	NaN	...	12.0	7.1	NaN	0

So, when thinking about *axes*..

- Axis = 0 --> Rows
- Axis = 1 --> Columns

You see this when
running dataframe.shape --> (n_rows,n_cols)

NASHVILLE SOFTWARE SCHOOL

Get Data Process + Clean Data Exploratory Data Analysis

CSV Workflow

Open the file in text editor to determine

- is there a header? **pandas default is *header = 'infer'***
- are there notes (non-data) at the top? **pandas default is *skiprows = None***
- are there footnotes or other (non-data) at the bottom? **pandas default is *nrows = None***

Read the file into a pandas DataFrame ***df = pd.read_csv()***

Check the top and the bottom to ensure all data was read ***df.head()*** and ***df.tail()***

Look at the dimensions ***df.shape***

NASHVILLE SOFTWARE SCHOOL

Get Data Process + Clean Data Exploratory Data Analysis

Workflow

- Clean column names

- Drop unnecessary columns (avoid using *inplace = True* since it will be deprecated next year, instead assign the df back to itself or use method chaining)

NASHVILLE SOFTWARE SCHOOL

pandas – https://pandas.pydata.org/pandas-docs/stable/api.html

- **pd.read_csv()** – read a comma delimited file; always look at the raw file in a text editor (not excel); additional arguments may be needed to handle extra rows at the top and extra data (footnotes) at the bottom.
- **df.head()** – looks at the top of the DataFrame; 5 rows by default
- **df.tail()** - looks at the bottomof the DataFrame; 5 rows by default
- **df.shape** – returns a tuple with number of rows and number of columns
- **df.drop()** – drop the specified labels (either rows or columns) from the DataFrame
- **df.columns** – column labels attribute
- **df.rename()** – rename values (can pass in a dictionary with existing columns as the key and new ones as the values)

- **df.loc[]** – pass in row name and column name to access data at that location
- **df.iloc[]** - pass in row index and column index to access data at that location (python is 0 indexed)
- **df.query()** – pass an expression to filter data in the DataFrame
- **df[[ ]]** - creates a slice (subset) of the DataFrame including just the columns passed in

Let’s open our first shared notebook - public_art1.ipynb