Exploratory Data Analysis in Python

BDAA

BIG DATA & ANALYTICS ASSOCIATION

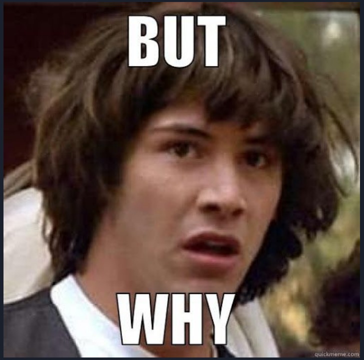
Outline

- Intro to EDA
- Intro to Dataset
- Doing EDA on our Dataset!
  - Importing Packages
  - Reading & Viewing Data
  - Plotting with Plotly Express
  - Manipulating our Data with Pandas
- Questions

What is Exploratory Data Analysis (EDA)?

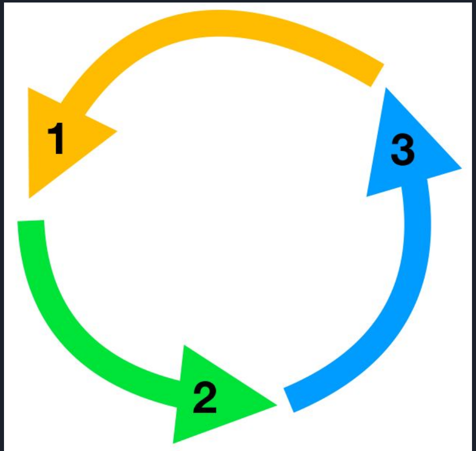
- A process for analyzing data that emphasizes looking at the data in various ways to detect patterns, spot anomalies, test hypotheses, and check assumptions
- Mainly uses visualization and summary statistics, sometimes light model building is included

Why perform EDA?



- Suggest hypotheses about the causes of observed phenomena
- Assess assumptions on which statistical inference will be based
- Support the selection of appropriate statistical tools and techniques
- Provide a basis for further data collection through surveys or experiments
  - Source: Behrens - Principles and Procedures of Exploratory Data Analysis - American Psychological Association - 1997

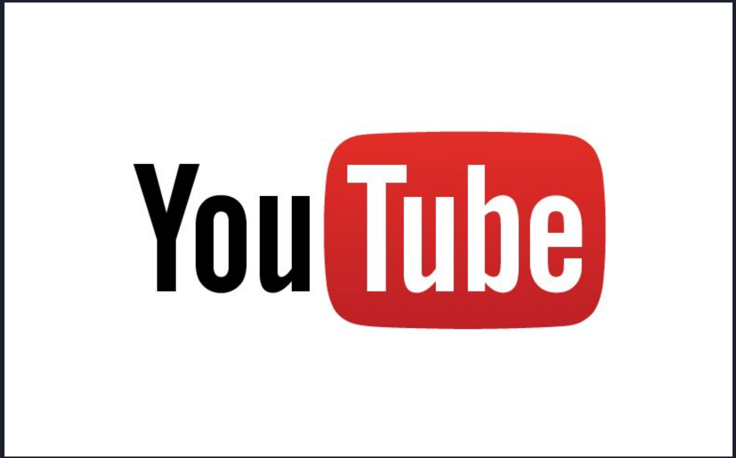
EDA Process



- No one process or set of rules
- Instead, EDA is an iterative cycle:
  1. Generate questions about your data
  2. Search for answers by visualizing, transforming, and modeling your data
  3. Use what you learn to refine your questions and/or generate new questions
- Explore every idea! Some will pan out, others will be dead ends
  - Source: Wickham & Grolemund - *R for Data Science*

Our Dataset: Trending Youtube Videos

- Info: https://www.kaggle.com/datasnaek/youtube-new#USvideos.csv



Python Packages & Modules

- A module is a single file (or files) that are imported under one import and used:

```
import my_module
```

- A package is a collection of modules:

```
import my_package
```

- All in all, collection of useful classes and functions for common tasks
- Many common ones: pandas, matplotlib, numpy, scikit-learn

Summary Statistics on Pandas Dataframes

Function	Returns
`df.mean()`	Mean of all columns
`df.corr()`	Correlation between columns
`df.count()`	Number of non-null values in each column
`df.max()`	Highest value in each column
`df.min()`	Lowest value of each column
`df.median()`	Median of each column
`df.std()`	Standard deviation of each column

Further Resources

- Classes at OSU:
  - CSE 4256
- BDAA Workshops! Next Thursday - Machine Learning with Python!
- Online Courses:
  - Coursera: Python for Everybody
- Books:
  - Beginner: *Python Crash Course: A Hands-On, Project-Based Introduction to Programming* - Eric Matthes
  - Beginner: *Head-First Python: A Brain-Friendly Guide* - Paul Barry
  - Intermediate: *Fluent Python* - Luciano Ramalho
- Internet
  - Codecademy - Interactive Tutorials
- People: BDAA Slack, Mentors, CSE Professors
- Projects
  - Kaggle

Questions?

- Any Questions?

- My Contact Info:
  - Leo Glowacki
  - Message me on Slack!
  - www.leoglowacki.com