Introduction to Data Science

Data Science Essentials

- Your name
- The place you call home
- Something people are usually surprised to discover about you

Classroom rules

- Ask lots of questions
- Help each other; learn from each other
- Coding tasks are a guide
  - You **don't** have to get them **all** done
  - You **can** form your own ideas and do your own exploration beyond what has been suggested

NASHVILLE SOFTWARE SCHOOL

Class format

- Review of previous coding tasks
- Concepts/Code Lecture
- Coding tasks
- Interactive with instruction team!

NASHVILLE SOFTWARE SCHOOL

Goals for the class

- Get hands-on experience of what it might be like to work as a data scientist
- Get an idea of whether or not this might be a good fit for a career
- Make discoveries and have fun

NASHVILLE SOFTWARE SCHOOL

Goals for today

- Pull new materials from the class repo
- DM us your github account name on Slack
- Define Data Science and the Data Science Process
- Understand the project questions
- Learn a little *pandas*
- Work on the coding tasks for this week

NASHVILLE SOFTWARE SCHOOL

Class Repository on GitHub

Internet

Your Computer

NASHVILLE SOFTWARE SCHOOL

Class Repository on GitHub

Fork

Personal Repository on GitHub

Internet

Your Computer

NASHVILLE SOFTWARE SCHOOL

Class Repository on GitHub

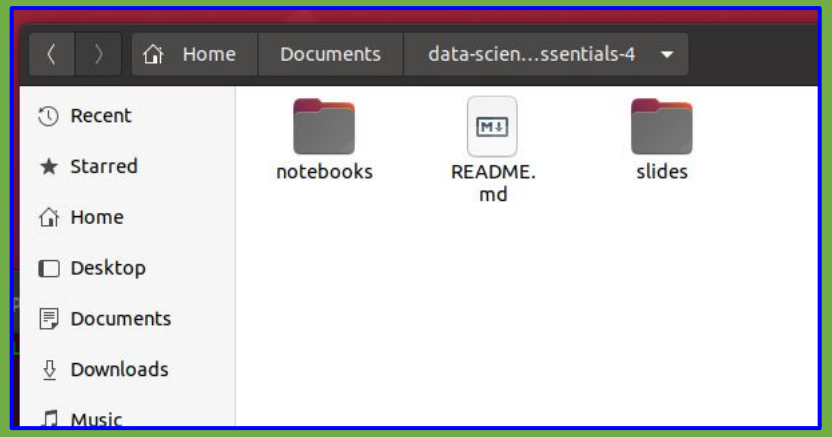
Fork

Personal Repository on GitHub

Internet

Clone

Your Computer



Local Copy of Personal Repository

NASHVILLE SOFTWARE SCHOOL

Class Repository on GitHub

Fork

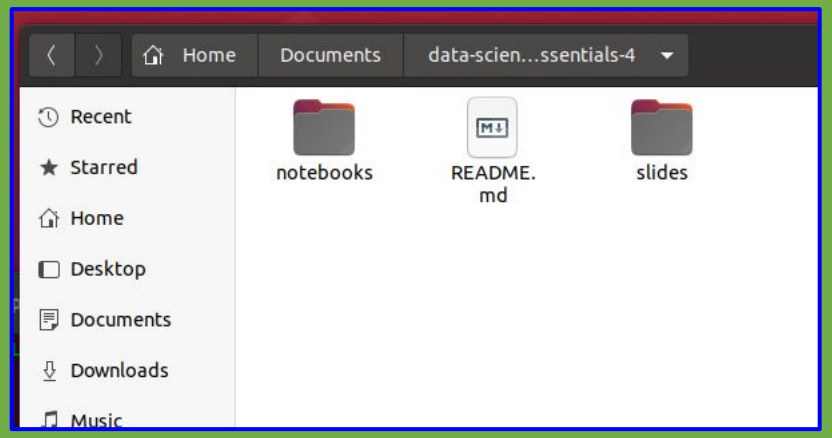
Personal Repository on GitHub

Internet

Clone

Push

Your Computer



Local Copy of Personal Repository

NASHVILLE SOFTWARE SCHOOL

Class Repository on GitHub

Fork

Personal Repository on GitHub

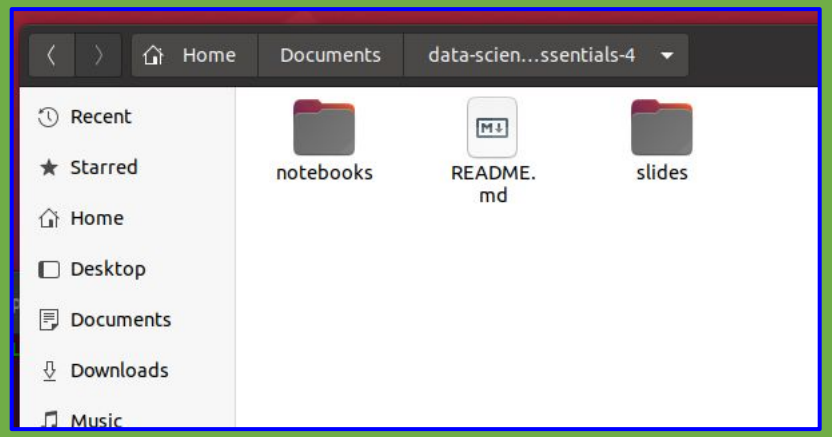
Internet

Pull

Clone

Push

Your Computer



Local Copy of Personal Repository

NASHVILLE SOFTWARE SCHOOL

Adding the class repo as a tracked repository

1. Add the class repository

```
git remote add upstream
https://github.com/Vanderbilt-Aspire-Data-Sci
ence/data-science-essentials-5.git
```

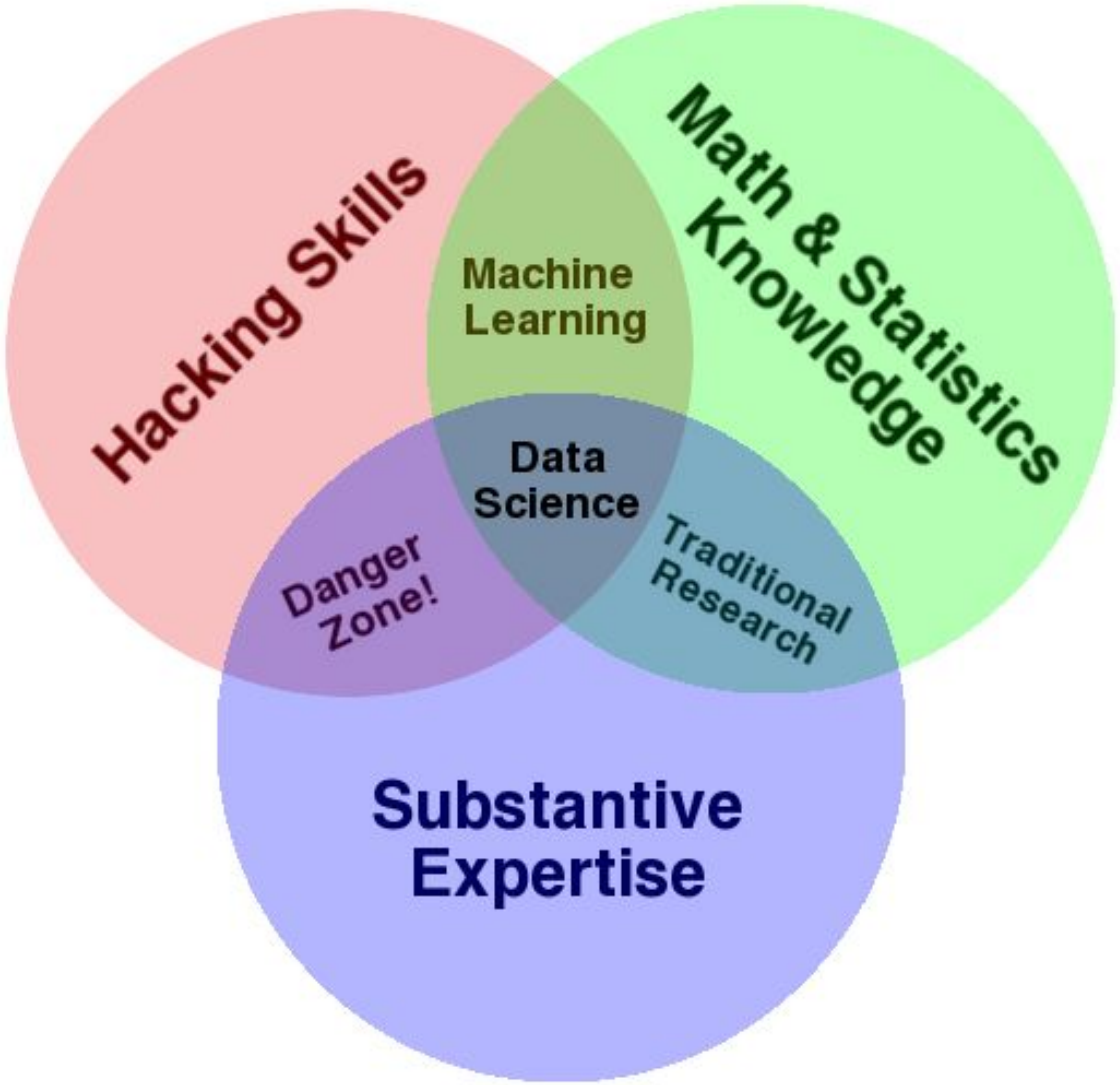
2. Pull changes to your local repository

```
git pull upstream main
```

NASHVILLE SOFTWARE SCHOOL

What is Data Science?

March 26, 2013



http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

NASHVILLE SOFTWARE SCHOOL

January 9, 2018

- **Data science** produces **insights**
- **Machine learning** produces **predictions**
- **Artificial intelligence** produces **actions**

VARIANCE EXPLAINED

David Robinson

Chief Data Scientist at DataCamp, works in R and Python.

Data science produces insights

Data science is distinguished from the other two fields because its goal is an especially human one: to gain insight and understanding. Jeff Leek has an excellent definition of the types of insights that data science can achieve, including descriptive ("the average client has a 70% chance of renewing") exploratory ("different salespeople have different rates of renewal") and causal ("a randomized experiment shows that customers assigned to Alice are more likely to renew than those assigned to Bob").

Again, not everything that produces insights qualifies as data science (the classic definition of data science is that it involves a combination of statistics, software engineering, and domain expertise). But we can use this definition to distinguish it from ML and AI. The main distinction is that in data science there's always a human in the loop: someone is understanding the insight, seeing the figure, or benefitting from the conclusion. It would make no sense to say "Our chess-playing algorithm uses data science to choose its next move," or "Google Maps uses data science to recommend driving directions".

This definition of data science thus emphasizes:

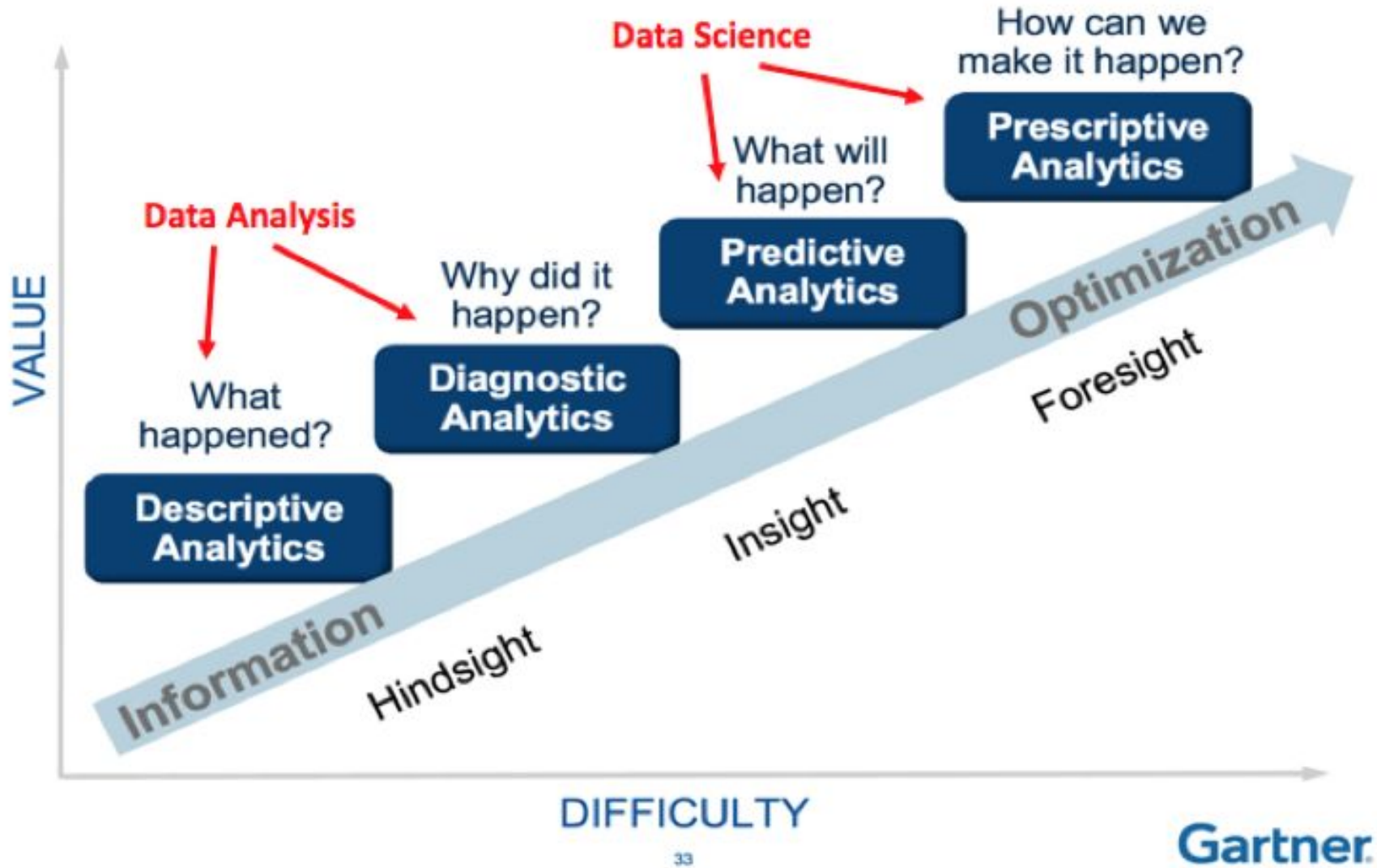
- Statistical inference
- Data visualization
- Experiment design
- Domain knowledge
- Communication

Data scientists might use simple tools: they could report percentages and make line graphs based on SQL queries. They could also use very complex methods: they might work with distributed data stores to analyze trillions of records, develop cutting-edge statistical techniques, and build interactive visualizations. Whatever they use, the goal is to gain a better understanding of their data.

http://varianceexplained.org/r/ds-ml-ai/

NASHVILLE SOFTWARE SCHOOL

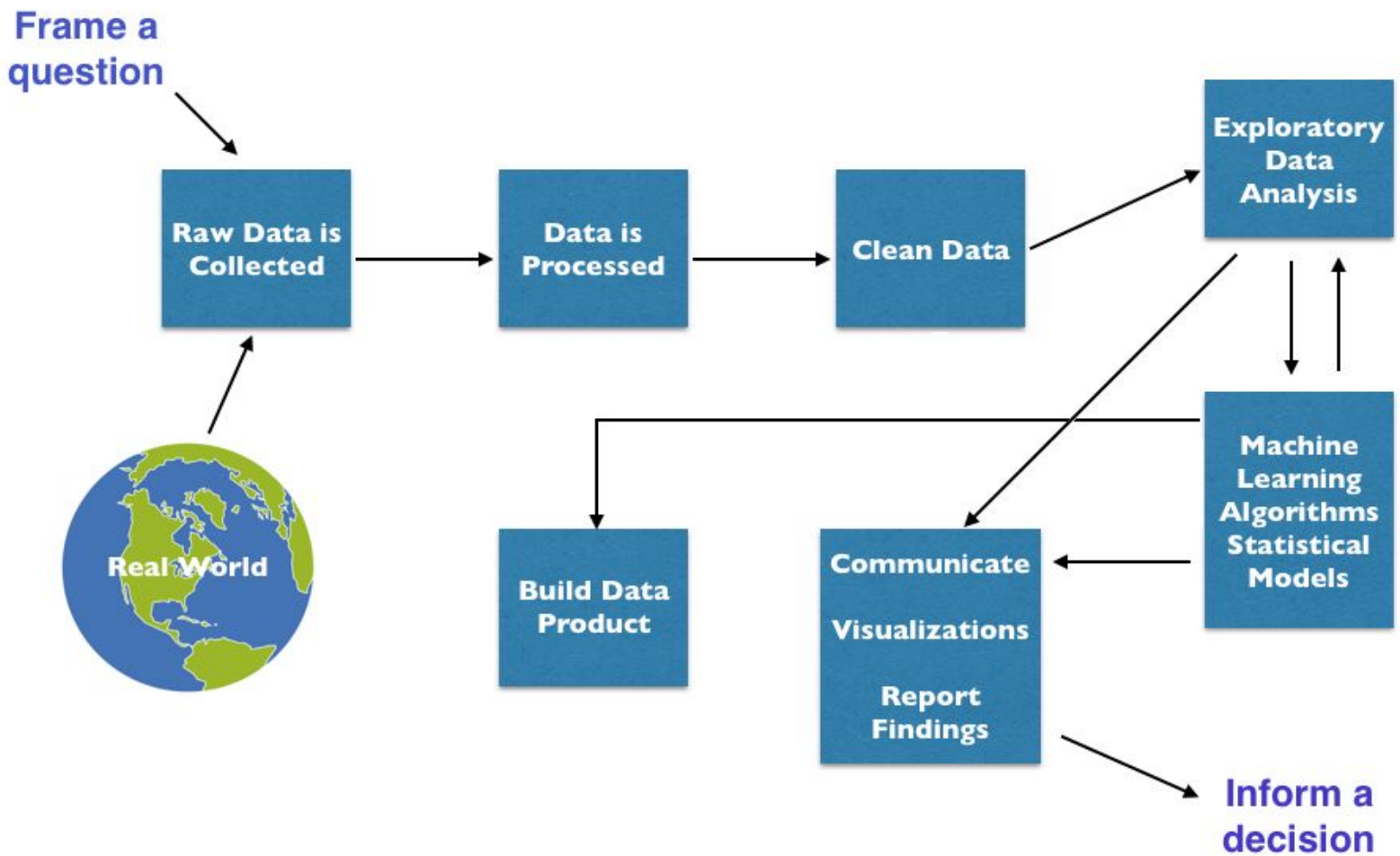
Gartner Analytic Ascendancy Model



Gartner

NASHVILLE SOFTWARE SCHOOL

Data Science Process



NASHVILLE SOFTWARE SCHOOL

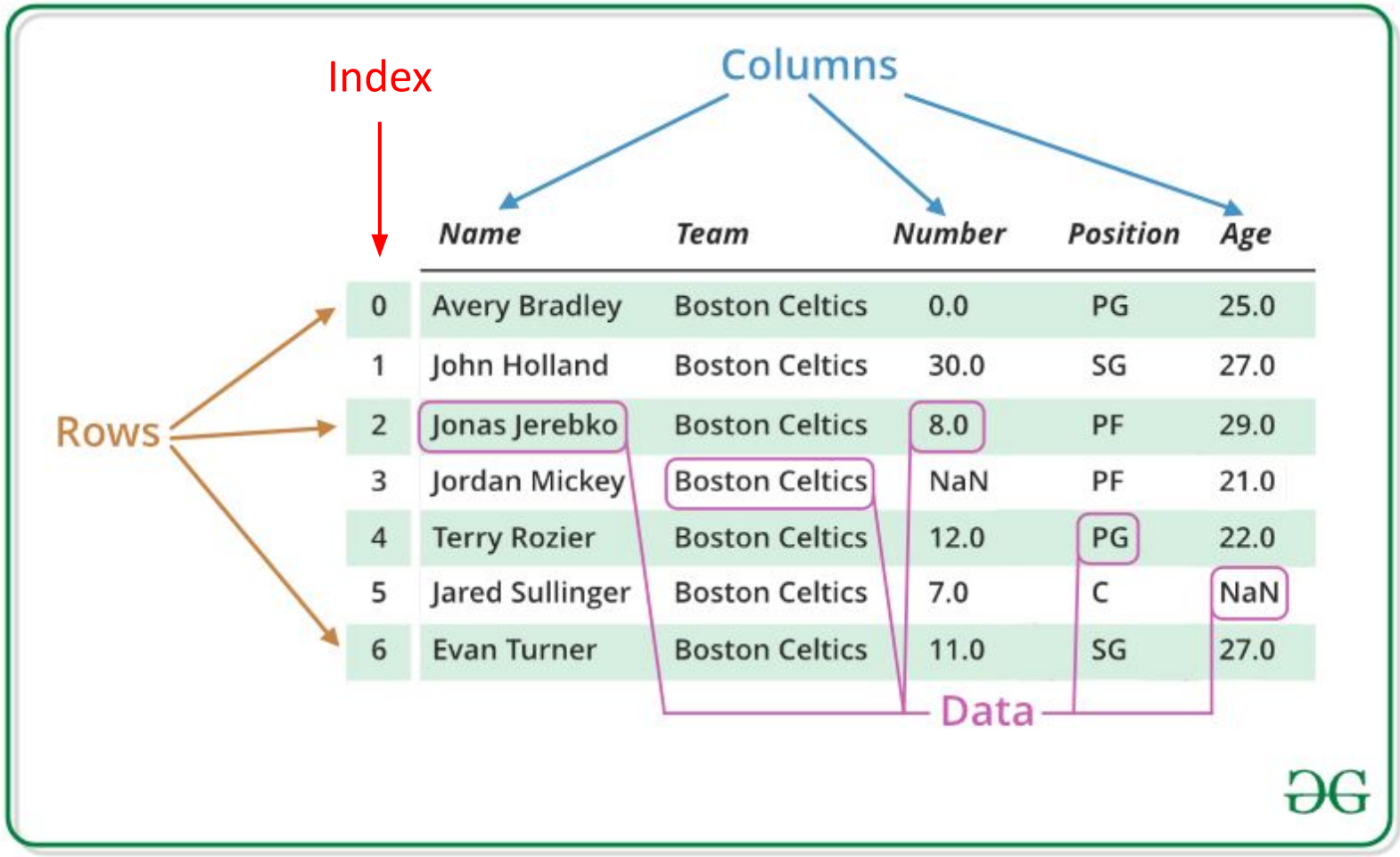
Project Goals

Introduction to *pandas*

We will work a lot with Python's *pandas* library, which provides methods for working with ***DataFrames*** and ***Series***.

A **DataFrame** is a two-dimensional (tabular) data structure.

A **Series** is a one-dimensional data structure – could be a row or a column of data, but usually when we work with a series it is a column of data.



	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

https://www.geeksforgeeks.org/python-pandas-dataframe/

NASHVILLE SOFTWARE SCHOOL

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

Importing Data

- **pd.read_csv()** – read a comma delimited file; good practice is to look at the raw file in a text editor (like Visual Studio Code, not Excel); additional arguments may be needed to handle extra rows at the top and extra data (footnotes) at the bottom.

Inspecting

- **df.info()** – method to get information about the DataFrame
- **df.dtypes** – datatypes attribute for the Data Frame
- **df.head()** – looks at the top of the DataFrame; 5 rows by default
- **df.tail()** - looks at the bottom of the DataFrame; 5 rows by default
- **df.shape** – returns a tuple with the number of rows and number of columns

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

Modifying

- **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.drop()** – drop the specified labels (either rows or columns) from the DataFrame

Summarizing

- **.unique()** – returns the unique values in a column
- **.nunique()** - returns the *number* of unique elements in a column
- **.value_counts()** - returns the unique elements in a column and the number of appearances of each

Slicing/Filtering

- **df.loc[]** – pass in row name and column name to access data at that location
- **df[[ ]]** - creates a slice (subset) of the DataFrame including just the columns passed in