CS M148 –

# Data Science Fundamentals

## Lecture #1: Introduction to CS M148

**Baharan Mirzasoleiman**

**UCLA Computer Science**

(modified from Harvard CS109A)

**Instructor**: Baharan Mirzasoleiman
http://web.cs.ucla.edu/~baharan
baharan@cs.ucla.edu
Lectures: Monday-Wednesday 8-10am
Office hours: Monday 10-10:30am (until 11am if there are more students), Zoom (same for the lectures)

**Lectures will be recorded**
**You can find all the info in the Syllabus (Canvas)**

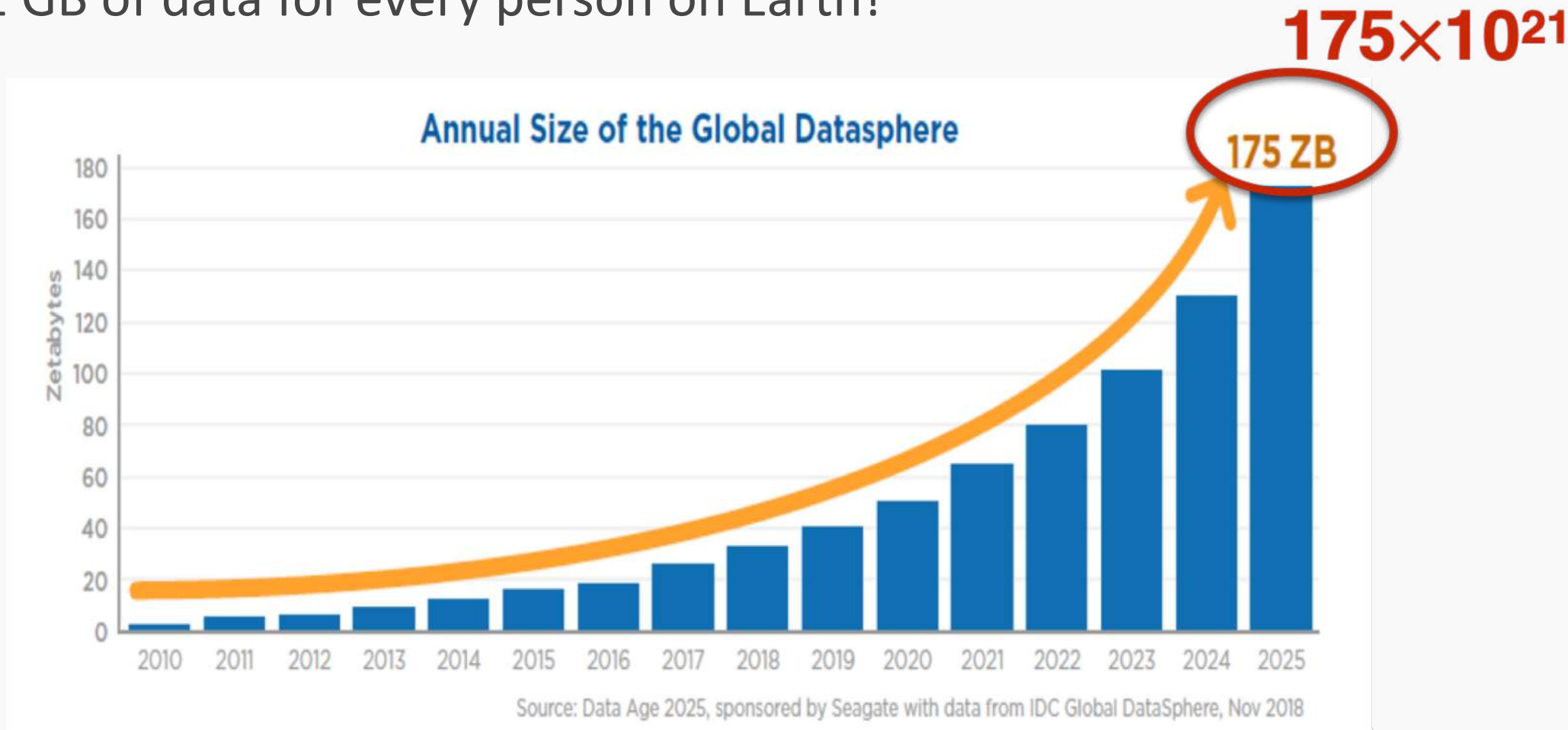
| TA | Email | Office Hours (**from week 2**) | Location |
|---|---|---|---|
| Lionel Levine | lionel@cs.ucla.edu | Thursday / 1-2pm | Zoom |
| Siddharth Joshi | sjoshi804@gmail.com | Wednesday / 2:30-3:30pm | Zoom |
| Wenhan Yang | hangeryang18@g.ucla.edu | Thursday / 2-3pm | Zoom |
| Yihe Deng | yihedeng@ucla.edu | Tuesday / 1-2pm | Zoom |
| Yu Yang | yuyang@cs.ucla.edu | Monday / 1-2pm | Zoom |

# Lecture Outline

- Why data science? Why taking (or not taking!) CS146?
- What is data science?
- What is this class and what it is not?
- The data science process
- Example

# How much data do we have?

22,571 GB of data for every person on Earth!



# 50 Best Jobs in America for 2020

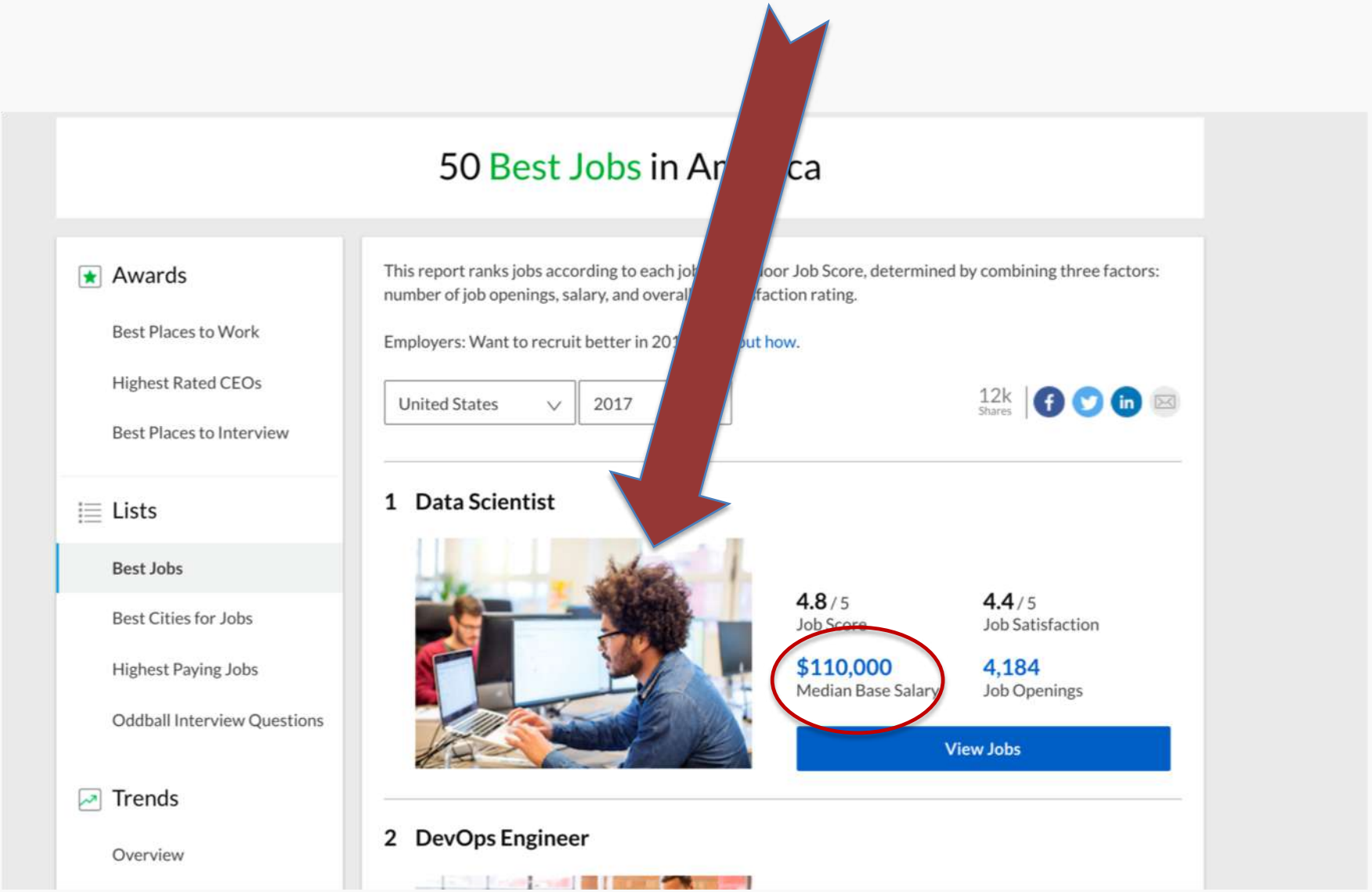
Best Jobs | 2020 | United States

Share

| | Job Title | Median Base Salary | Job Satisfaction | Job Openings | |
|---|---|---|---|---|---|
| #1 | Front End Engineer | $105,240 | 3.9/5 | 13,122 | View Jobs |
| #2 | Java Developer | $83,589 | 3.9/5 | 16,136 | View Jobs |
| #3 | Data Scientist | $107,801 | 4.0/5 | 6,542 | View Jobs |
| #4 | Product Manager | $117,713 | 3.8/5 | 12,173 | View Jobs |
| #5 | DevOps Engineer | $107,310 | 3.9/5 | 6,603 | View Jobs |
| #6 | Data Engineer | $102,472 | 3.9/5 | 6,941 | View Jobs |
| #7 | Software Engineer | $105,563 | 3.6/5 | 50,438 | View Jobs |

# Why?

Jobs!


50 Best Jobs in Ar ca
Awards
Best Places to Work
Highest Rated CEOs
Best Places to Interview
Lists
Best Jobs
Best Cities for Jobs
Highest Paying Jobs
Oddball Interview Questions
Trends
Overview
This report ranks jobs according to each job oor Job Score, determined by combining three factors:
number of job openings, salary, and overall faction rating.
Employers: Want to recruit better in 201 ut how.
United States
2017
12k
Shares
1 Data Scientist
4.8 / 5
Job Score
4.4 / 5
Job Satisfaction
$110,000
Median Base Salary
4,184
Job Openings
View Jobs
2 DevOps Engineer

# Why?

Why are you here?

# Memes !

## Deep Learning

What society thinks I do

What my friends think I do

What other computer scientists think I do

What mathematicians think I do

What I think I do

```
In [1]:
import ker
Using Tens
```

What I


Data Science!

# What is data science?

# A little bit of history

# History

Long time ago (thousands of years) science was only empirical and people counted stars

# History (cont)

Long time ago (thousands of years) science was only empirical and people counted stars or crops

# History (cont)

Long time ago (thousands of years) science was only

empirical and people counted stars or crops and used the data to create machines to describe the phenomena

# History (cont)

Few hundred years: theoretical approaches, try to derive equations to describe general phenomena.

1. $\nabla \cdot \mathbf{D} = \rho_V$
2. $\nabla \cdot \mathbf{B} = 0$
3. $\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$
4. $\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}$

$$T^2 = \frac{4\pi^2}{GM} a^3$$

can be expressed as simply

$$T^2 = a^3$$

If expressed in the following units:

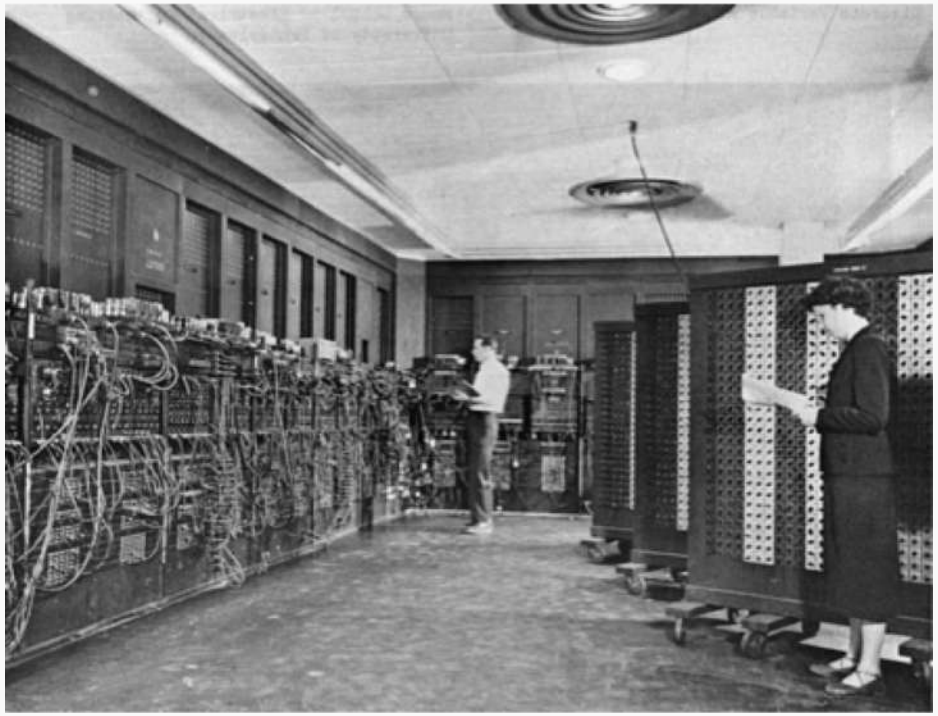
- $T$ Earth years
- $a$ Astronomical units AU (a = 1 AU for Earth)
- $M$ Solar masses $M_\odot$

Then $\frac{4\pi^2}{G} = 1$

$$H(t)|\psi(t)\rangle = i\hbar \frac{\partial}{\partial t}|\psi(t)\rangle$$

# History (cont)

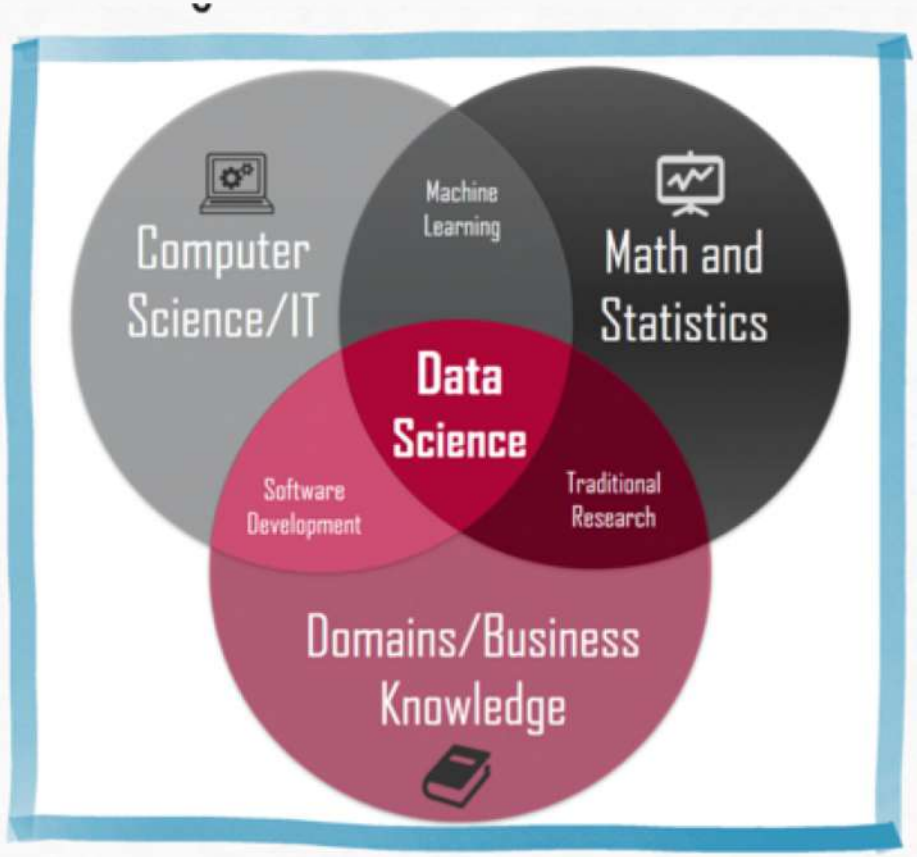
About a hundred years ago: computational approaches



# History (cont)

And then …. data science

In both data science and machine learning we extract pattern and insights from data.




- Inter-disciplinary
- Data and task focused
- Resource aware
- Adaptable to changes in the environment and needs

# Technology Trends

2020s ?

2010s **Data Industry**
- Collect and sell information

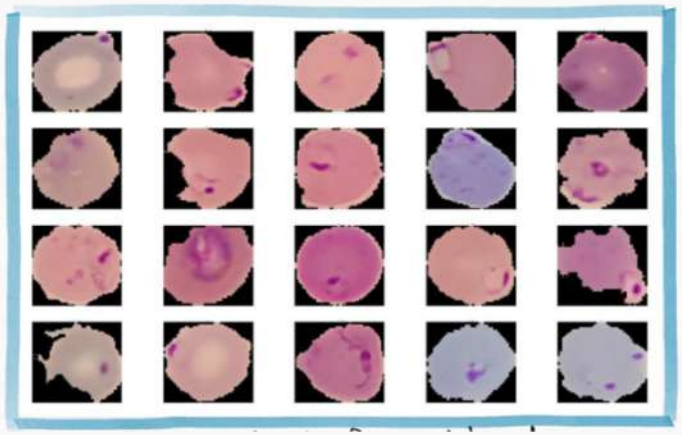
2000s **Internet Industry**
- Online retailers and services

1990s **Software Industry**
- Sold computer software

1980s **Hardware Industry**
- Sold computers


# The Potential of Data Science

## Disease Diagnosis

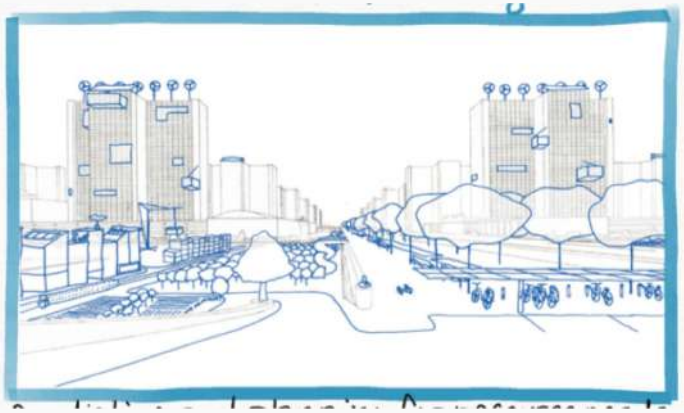


Detecting malaria from blood smears

## Drug Discovery

Quickly discovering new drugs for COVID

## Urban Planning



Predicting and planning for resource needs

## Agriculture

Precision agriculture

# The Potential of Data Science

## Gender Bias

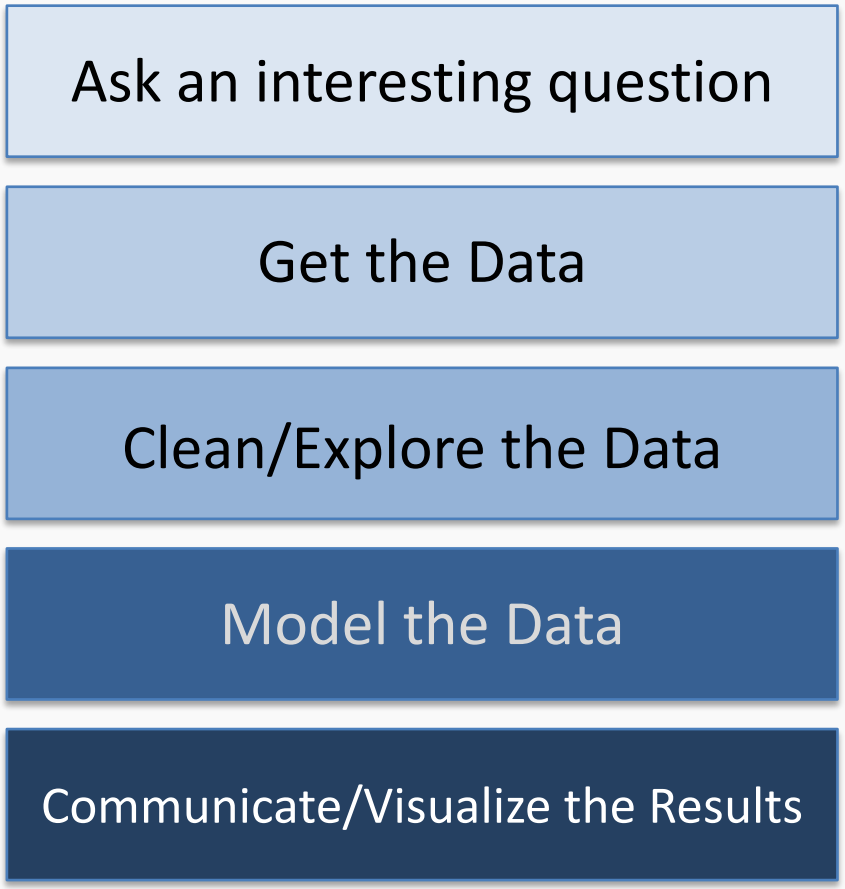
Some DS models for evaluating job applications show bias in favor of male candidate

## Racial Bias

Risk models used in US courts have shown to be biased against non-white defendants

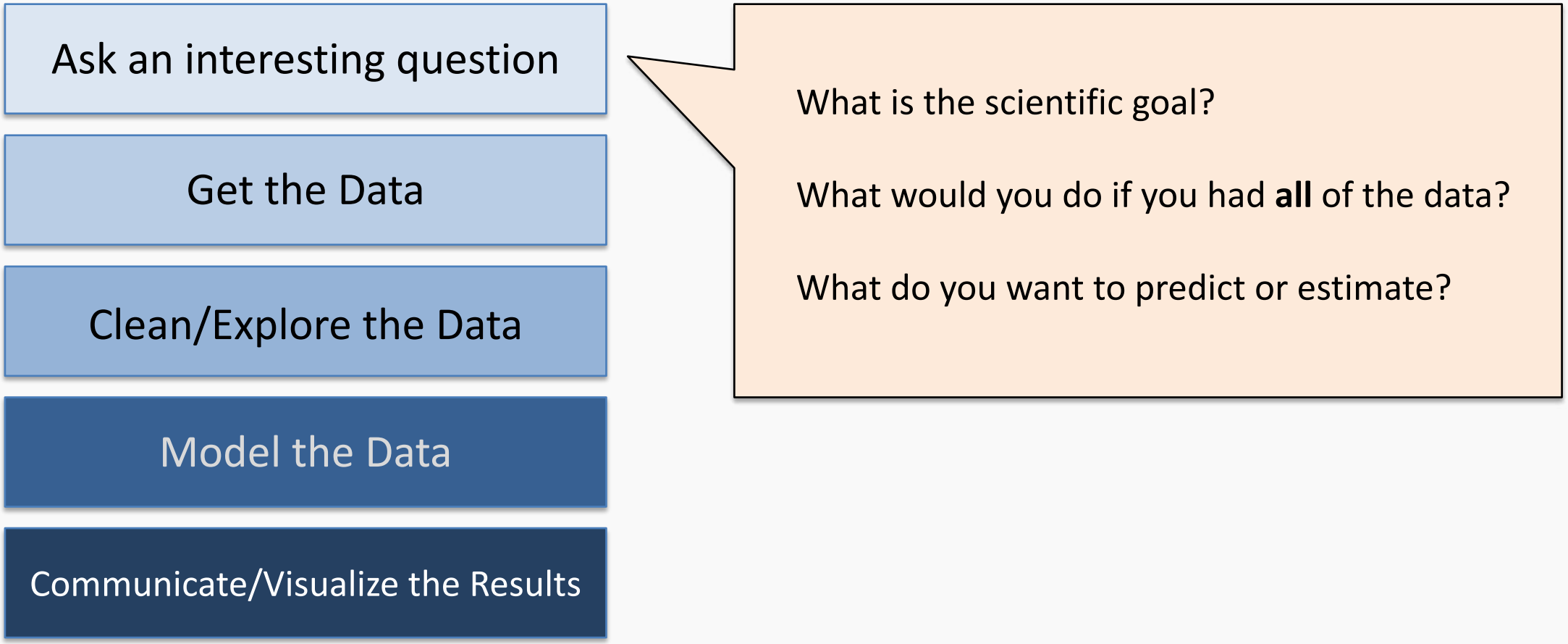
# What?

## The Data Science Process



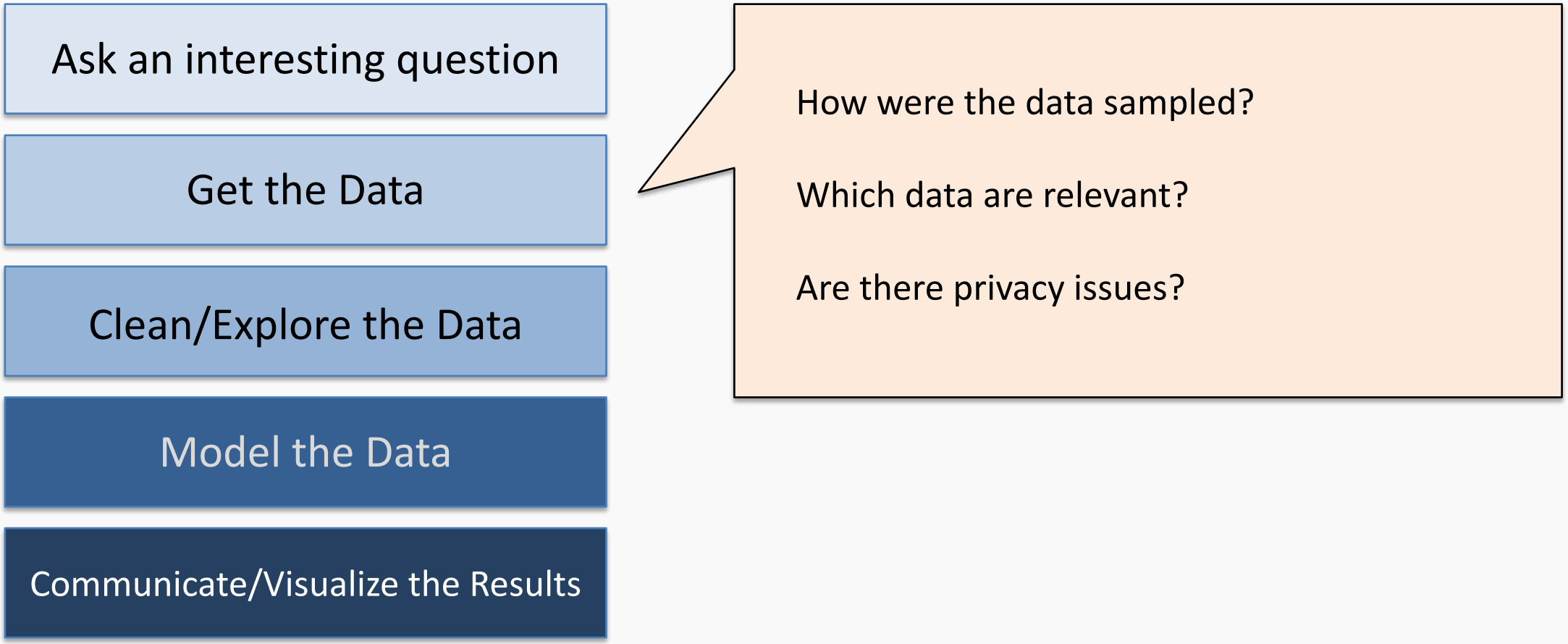
# What?

## The Data Science Process



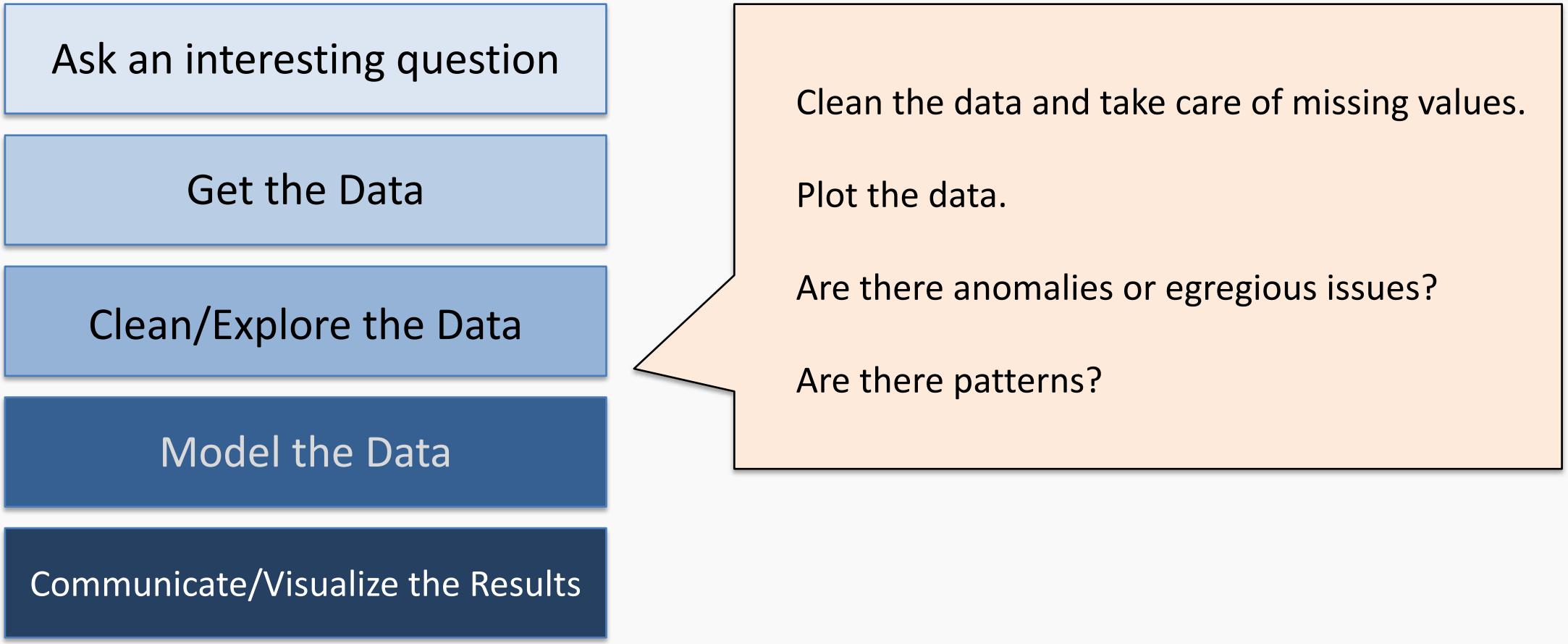
# What?

## The Data Science Process



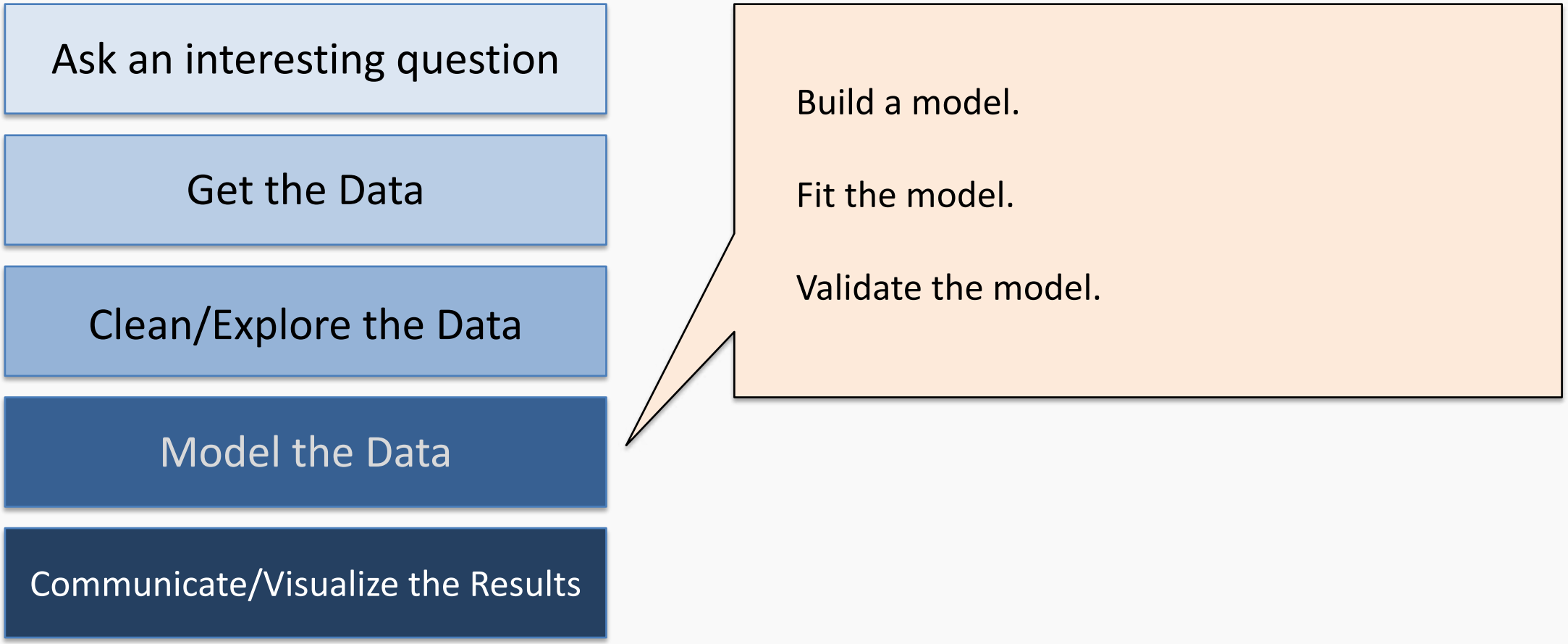
# What?

## The Data Science Process



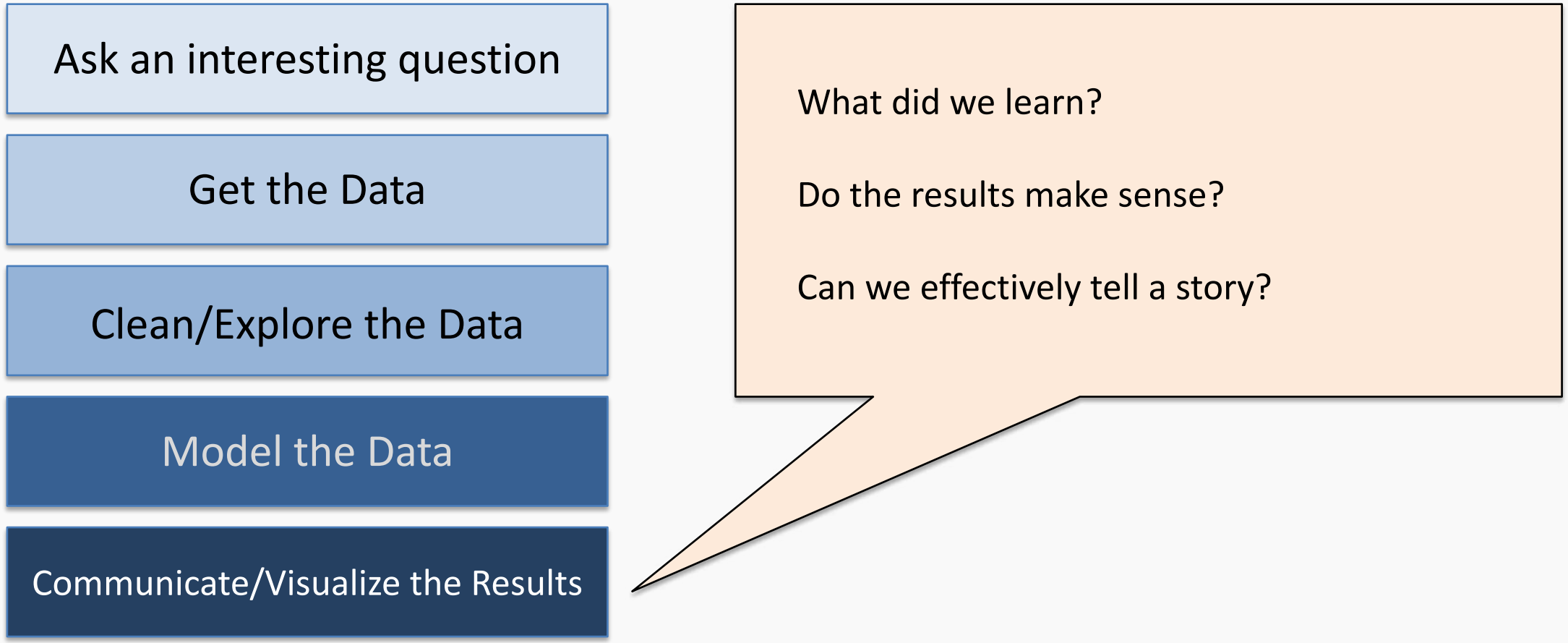
# What?

## The Data Science Process



# What?

## The Data Science Process



# What Is Data Science?


https://www.youtube.com/watch?v=X3paOmcrTjQ

# Get the data

**Will not be covered in this course**

- Can be from the internet or external/internal databases).
- Must be extracted into a **usable format** (.csv, json, xml, etc..)

**Skills Required**:

- **Database Management**: MySQL, PostgresSQL, MongoDB
- **Querying Relational Databases**
- **Retrieving Unstructured Data**: text, videos, audio files, documents
- **Distributed Storage**: Hadoops, Apache Spark/Flink

**We will talk about data collection process and the bias it can cause in the data**

# Scrubbing / Cleaning Your Data

**Covered by lectures & projects**

**Requires the most time and effort**

- The results and output of your machine learning model is only as good as what you put into it

**Objective:**

- **Examine the data**: understand every feature you're working with, identify errors, missing values, and corrupt records
- **Clean the data:** throw away, replace, and/or fill missing values/errors

**Skills Required:**

- **Scripting language**: Python, R, SAS
- **Data Wrangling Tools**: Python Pandas, R
- **Distributed Processing**: Hadoop, Map Reduce / Spark

# Exploring (Exploratory Data Analysis)

**Covered by lectures & projects**

Trying to **understand** what patterns and values our data has

- Different types of **visualizations** and **statistical testings** to back up the findings.
- Derive hidden meanings behind data through various graphs and analysis.

**Objective:**

- Find patterns in your data through visualizations and charts
- Extract features by using statistics to identify and test significant variables

**Skills Required:**

- **Python**: Numpy, Matplotlib, Pandas, Scipy, **R**: GGplot2, Dplyr
- **Inferential statistics, Experimental Design, Data Visualization**

# Modeling (Machine Learning)

**Covered by lectures & projects**

After cleaning your data and finding what features are most important, using your model as a predictive tool will enhance your business **decision making**

**Objective:**

- **In-depth Analytics:** create predictive models/algorithms
- **Evaluate and refine the model**

**Skills Required:**

- **Machine Learning**: Supervised/Unsupervised algorithms
- **Evaluation methods**
- **Machine Learning Libraries**: Python (Sci-kit Learn) / R (CARET)
- **Linear algebra & Multivariate Calculus**

# Interpreting (Data Storytelling)

**Practice in final project's report**

Understand and learn how to explain your findings through communication

**Objective**:

- **Identify business insights**: return back to business problem
- **Visualize your findings accordingly**: keep it simple and priority driven
- **Tell a clear and actionable story**: effectively communicate to non-technical audience

**Skills Required:**

- **Business Domain Knowledge**
- **Data Visualization Tools**: Tablaeu, D3.JS, Matplotlib, GGplot, Seaborn
- **Communication**: Presenting/Speaking & Reporting/Writing

# Updating Your Model

**Covered by lectures & projects**

The more data you receive the more frequent the update.

- If not, your model will degrade over time and won't perform as good

# Goal of the course

## Theory

1. Key Machine Learning concept
2. Important metrics for evaluation
3. Handling different kinds of data
4. Extracting insights from analysis of the models

## Practice

1. Implement ML and deep learning models using python libraries
2. Using free online tools and resources for data science

## Impact

1. Solving real-life problems using DS
2. Evaluating the social impact of DS

**We wont get deep in math**

**We will focus on insights & interpretation**

## Project: Data

Data Formats , Pandas
Data cleaning & exploration

## Weeks 1-4: Regression

Data collection & bias
kNN Regression
Linear Regression
Multi and Poly Regression
Model Selection and Cross Validations
Hypothesis testing
Ridge and Lasso Regularization

## Weeks 4-5: Classification

kNN Classification
Logistic Regression
Multi-class Classification

## Weeks 5-6: Trees

Decision Trees
Bagging
Random Forest
**Week 6: Midterm Exam**

## Weeks 7: Data

Data Imputation

## Weeks 8-9: Neural Networks

Multi-Layer Perceptron
Architecture of NN
Fitting NN, backprop and SGD
Regularization of NN

## Weeks 10

Clustering
Model Interpretation
Extra: learning from large datasets

**We may make a few changes based on how it goes**

# Class statistics

Let’s fill out a survey:

shorturl.at/hwBC5

# Do you need CS148?

## If you took CS146 or CS145 before

CS146 and CS 145 are more advanced than CS148

CS148 is more focused on data, and interpretation

You may see the same topics in CS146, CS145.
CS148 covers them from a more applied perspective.

# Grading

Find the schedule in the Syllabus uploaded to Canvas

- Homework: 20% — 3 homework assignments
- Midterm: 20% — 6th week of the class — **In class**
- Projects: 20% — 3 projects
- Final Exam: 40%
- **Total: 100%**

Ungraded labs are for you to practice

# Logistics

**Lectures:** Monday/Wednesday 8:00 am - 9:50 am, Kaplan 150, Zoom **recorded**

**Discussions:**

- Lionel Levine Sec (1A) F 10:00 pm - 11:50 pm, Royce Hall 156
- Siddharth Joshi Sec (1B) F 12:00 pm - 1:50 pm, Royce Hall 154
- Wenhan Yang Sec (1D) F 4:00 pm - 5:50 pm, Franz Hall 1260
- Yihe Deng Sec (1C) F 2:00 pm - 3:50 pm, Dodd Hall 78
- Yu Yang Sec (1E) F 12:00 pm - 1:50 pm, Boelter Hall 2760

**Office hours (**Lionel Levine**):** Thursday / 1-2pm, Zoom

**Office hours** (Siddharth Joshi): Wednesday / 2:30-3:30pm, Zoom

**Office hours** (Wenhan Yang): Thursday / 2-3pm, Zoom

**Office hours** (Yihe Deng): Tuesday / 1-2pm, Zoom

**Office hours** (Yu Yang): Monday / 1-2pm, Zoom (see canvas)

**Office hours (**Baharan Mirzasoleiman**):** Monday 10am-10:30am (until 11am if needed), Zoom

All the zoom links are also posted on **Canvas**.

# Logistics

Enroll in:

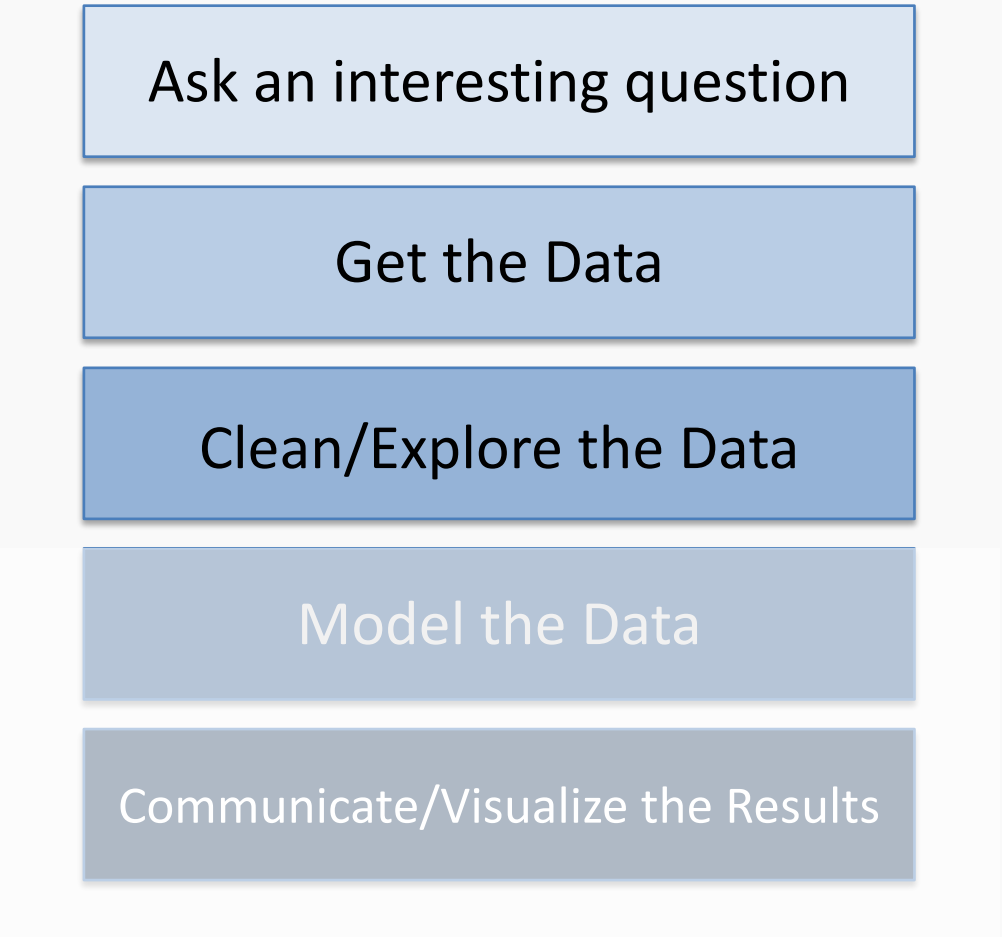
- Piazza: https://piazza.com/ucla/winter2023/csm148 **Ask questions**
- Gradescope in Bruinlearn **Upload assignments**

If you need PTE or would like to audit: shorturl.at/bxyHP

# Why Not to Become a Data Analyst?

https://www.youtube.com/watch?v=M2ySRYpo9S0

# The Data Science Process: Example



# Example:

- Let's say that we are interested in the English Premier League (football/soccer) and want to build a model to predict a player's market value.

**Question** Does age affect one's market value?

# Example: Get the data

| name | club | age | position | market value |
|---|---|---|---|---|
| Alexis Sanchez | Arsenal | 28 | LW | 65 |
| Mesut Ozil | Arsenal | 28 | AM | 50 |
| Petr Cech | Arsenal | 35 | GK | 7 |
| Theo Walcott | Arsenal | 28 | RW | 20 |
| Laurent Koscielny | Arsenal | 31 | CB | 22 |

Web scraping with Python:

page = requests.get(url)

soup = BeautifulSoup(page.content, “html.parser”)

from www.transfermarkt.us

# **Example**: Explore the data

## EDA helps you:

- Ensure your data is as expected/valid/appropriate for the task
- Provides insights into a dataset
- Extract/determine important variables/attributes/features
- Detect outliers and anomalies
- Test underlying assumptions
- Make informed decisions in developing models

# **Example**: Explore the data

- EDA is an approach/philosophy **not** just a set of tools or techniques.
- Explore **global** properties: use histograms, scatter plots, and aggregation functions to summarize the data
- Explore **group** properties: group like-items together to compare subsets of the data (are the comparison results reasonable/expected?)
- This approach can be done at any time and any stage of the data science process

# Example: Explore the data

| name | club | age | position | market value |
|---|---|---|---|---|
| Alexis Sanchez | Arsenal | 28 | LW | 65 |
| Mesut Ozil | Arsenal | 28 | AM | 50 |
| | | | GK | 7 |
| | | | RW | 20 |
| | | | CB | 22 |

- Credible/Trustworthy?
- Possibly subjective market values?
- Sampled data

from www.transfermarkt.us

# **Example**: Explore the data

| name | club | age | position | market value |
|---|---|---|---|---|
| Alexis Sanchez | Arsenal | 28 | LW | 65 |
| Mesut Ozil | Arsenal | 28 | AM | 50 |
| Petr Cech | Arsenal | 35 | GK | 7 |
| Theo Walcott | Arsenal | 28 | RW | 20 |
| Laurent Koscielny | Arsenal | 31 | CB | 22 |

from www.transfermarkt.us

# **Example**: Explore the data

| name | club | age | position | market value |
|---|---|---|---|---|
| Alexis Sanchez | Arsenal | 28 | LW | 65 |
| Mesut Ozil | Arsenal | 28 | AM | 50 |
| Petr Cech | Arsenal | 35 | GK | 7 |
| Theo Walcott | Arsenal | 28 | RW | 20 |
| Laurent Koscielr | | | | 22 |

Does it contain the necessary information?

w.transfermarkt.us

# Example: Explore the data

| name | club | age | position | market value |
|---|---|---|---|---|
| Alexis Sanchez | Arsenal | 28 | LW | 65 |
| Mesut Ozil | Arsenal | 28 | AM | 50 |
| Petr Cech | Arsenal | 35 | GK | 7 |
| Theo Walcott | Arsenal | 28 | RW | 20 |
| Laurent Koscielny | Arsenal | 31 | CB | 22 |

Missing data? Imputation needed?

# Example: Explore the data

| name | club | age | position | market value |
|---|---|---|---|---|
| Alexis Sanchez | Arsenal | 28 | LW | 65 |
| Mesut Ozil | Arsenal | 28 | AM | 50 |
| Petr Cech | Arsenal | 35 | GK | 7 |
| Theo Walcott | Arsenal | 28 | RW | 20 |
| Laurent Koscielny | Arsenal | 31 | CB | 22 |

Are the data types okay (`df.dtypes`)? Should be casted?

# Example: Explore the data

| name | club | age | position | market value |
|---|---|---|---|---|
| Alexis Sanchez | Arsenal | 28 | LW | 65 |
| Mesut Ozil | Arsenal | 28 | AM | 50 |
| Petr Cech | Arsenal | 35 | GK | 7 |
| Theo Walcott | Arsenal | 28 | RW | 20 |
| Laurent Koscielny | Arsenal | 31 | CB | 22 |

Are the values reasonable? `DataFrame.describe()` …

# Example: Explore the data

| | age | page_views | fpl_value | fpl_points | market_value |
|---|---|---|---|---|---|
| count | 461.000000 | 461.000000 | 461.000000 | 461.000000 | 461.000000 |
| mean | 26.804772 | 763.776573 | 5.447939 | 57.314534 | 11.012039 |
| std | 3.961892 | 931.805757 | 1.346695 | 53.113811 | 12.257403 |
| min | 17.000000 | 3.000000 | 4.000000 | 0.000000 | 0.050000 |
| 25% | 24.000000 | 220.000000 | 4.500000 | 5.000000 | 3.000000 |
| 50% | 27.000000 | 460.000000 | 5.000000 | 51.000000 | 7.000000 |
| 75% | 30.000000 | 896.000000 | 5.500000 | 94.000000 | 15.000000 |
| max | 38.000000 | 7664.000000 | 12.500000 | 264.000000 | 75.000000 |

Are the values reasonable? `DataFrame.describe()` …

# Example: Explore the data

| | age | page_views | fpl_value | fpl_points | market_value |
|---|---|---|---|---|---|
| count | 461.0000 | | | | |
| mean | 26.8047 | | | | |
| std | 3.961892 | 931.805757 | 1.346695 | 55.113811 | 257403 |
| min | 17.000000 | 3.000000 | 4.000000 | 0.000000 | 0.050000 |
| 25% | 24.000000 | 220.000000 | 4.500000 | 5.000000 | 3.000000 |
| 50% | 27.000000 | 460.000000 | 5.000000 | 51.000000 | 7.000000 |
| 75% | 30.000000 | 896.000000 | 5.500000 | 94.000000 | 15.000000 |
| max | 38.000000 | 7664.000000 | 12.500000 | 264.000000 | 75.000000 |

This seems abnormally low. Is it correct? Who is this?

Are the values reasonable? `DataFrame.describe()` …

# Example: Explore the data

| | age | page_views | fpl_value | fpl_points | market_value |
|---|---|---|---|---|---|
| cou | | | | | 461.000000 |
| me | | | | | 11.012039 |
| s | | | | | 12.257403 |
| min | 17.000000 | 3.000000 | 4.000000 | 0.000000 | 0.050000 |
| 25% | 24.000000 | 220.000000 | 4.500000 | 5.000000 | 3.000000 |
| 50% | 27.000000 | 460.000000 | 5.000000 | 51.000000 | 7.000000 |
| 75% | 30.000000 | 896.000000 | 5.500000 | 94.000000 | 15.000000 |
| max | 38.000000 | 7664.000000 | 12.500000 | 264.000000 | 75.000000 |

This also seems suspicious. Is it correct? Who is this?

Are the values reasonable? `DataFrame.describe()` …

# Inspecting suspicious data

This accounts for both extreme values that we noticed. But, is this data **truly accurate?** It's worth validating online, elsewhere.

```
import pandas as pd
df = pd.read_csv("epl.csv")
df.iloc[df['market_value'].idxmin()]
```

```
name                Eduardo Carvalho
club                         Chelsea
age                               34
position                          LW
position_cat                       1
market_value                    0.05
page_views                       467
fpl_value                          5
fpl_sel                        0.10%
fpl_points                         0
region                             2
nationality                 Portugal
new_foreign                        0
age_cat                            6
club_id                            5
big_club                           1
new_signing                        1
Name: 109, dtype: object
```

# Example: Explore the data

| | age | page_views | fpl_value | fpl_points | market_value |
|---|---|---|---|---|---|
| coun | | | | | |
| mea | | | | | |
| st | | | | | |
| mi | | | | | |
| 25% | 24.000000 | 220.000000 | 4.500000 | 5.000000 | .000000 |
| 50% | 27.000000 | 460.000000 | 5.000000 | 51.000000 | 7.000000 |
| 75% | 30.000000 | 896.000000 | 5.500000 | 94.000000 | 15.000000 |
| max | 38.000000 | 7664.000000 | 12.500000 | 264.000000 | 75.000000 |

**What is going on here?!** Is someone actually paid that much more than others? And someone scores that much more?

from www.transfermarkt.us

```
df.loc[df['market_value'] >= 15].sort_values(by='market_value', ascending=False).head(15)
```

| | name | club | age | position | position_cat | market_value | page_views | fpl_value | fpl_se | fpl_points |
|---|---|---|---|---|---|---|---|---|---|---|
| **92** | Eden Hazard | Chelsea | 26 | LW | 1 | 75.0 | 4220 | 10.5 | 2.30% | 224 |
| **263** | Paul Pogba | Manchester+United | 24 | CM | 2 | 75.0 | 7435 | 8.0 | 19.50% | 115 |
| **0** | Alexis Sanchez | Arsenal | 28 | LW | 1 | 65.0 | 4329 | 12.0 | 17.10% | 264 |
| **241** | Sergio Aguero | Manchester+City | 29 | CF | 1 | 65.0 | 4046 | 11.5 | 9.70% | 175 |
| **240** | Kevin De Bruyne | Manchester+City | 26 | AM | 1 | 65.0 | 2252 | 10.0 | 17.50% | 199 |
| **377** | Harry Kane | Tottenham | 23 | CF | 1 | 60.0 | 4161 | 12.5 | 35.10% | 224 |
| **104** | N%27Golo Kante | Chelsea | 26 | DM | 2 | 50.0 | 4042 | 5.0 | 13.80% | 83 |
| **1** | Mesut Ozil | Arsenal | 28 | AM | 1 | 50.0 | 4395 | 9.5 | 5.60% | 167 |
| **260** | Romelu Lukaku | Manchester+United | 24 | CF | 1 | 50.0 | 3727 | 11.5 | 45.00% | 221 |
| **93** | Diego Costa | Chelsea | 28 | CF | 1 | 50.0 | 4454 | 10.0 | 3.00% | 196 |
| **214** | Philippe Coutinho | Liverpool | 25 | AM | 1 | 45.0 | 2958 | 9.0 | 30.80% | 171 |
| **242** | Raheem Sterling | Manchester+City | 22 | LW | 1 | 45.0 | 2074 | 8.0 | 3.80% | 149 |
| **376** | Dele Alli | Tottenham | 21 | CM | 2 | 45.0 | 4626 | 9.5 | 38.60% | 225 |
| **98** | Thibaut Courtois | Chelsea | 25 | GK | 4 | 40.0 | 1260 | 5.5 | 18.50% | 141 |
| **215** | Sadio Mane | Liverpool | 25 | LW | 1 | 40.0 | 3219 | 9.5 | 5.30% | 156 |

# Example: Explore the data

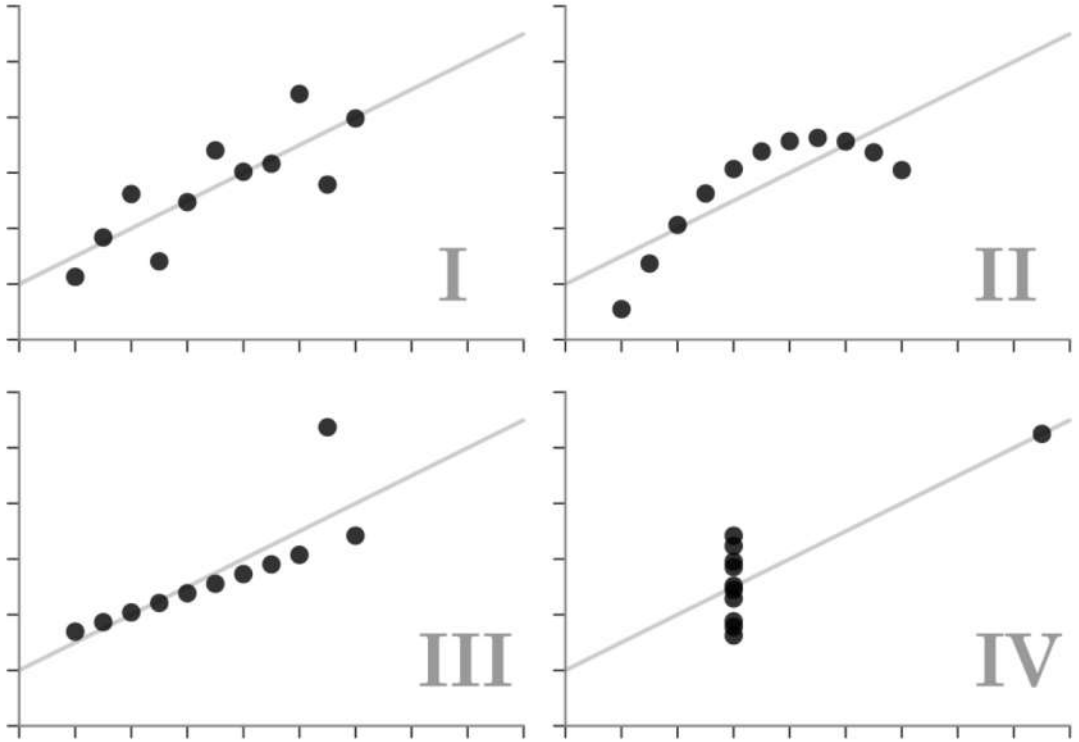
| | age | page_views | fpl_value | fpl_points | market_value |
|---|---|---|---|---|---|
| count | 461.000000 | 461.000000 | 461.000000 | 461.000000 | 461.000000 |
| mean | 26.804772 | 763.776573 | 5.447939 | 57.314534 | 11.012039 |
| std | 3.961892 | 931.805757 | 1.346695 | 53.113811 | 12.257403 |
| min | 17.000000 | 3.000000 | 4.000000 | 0.000000 | 0.050000 |
| 25% | 24.000000 | 220.000000 | 4.500000 | 5.000000 | 3.000000 |
| 50% | 27.000000 | 460.000000 | 5.000000 | 51.000000 | 7.000000 |
| 75% | 30.000000 | 896.000000 | 5.500000 | 94.000000 | 15.000000 |
| max | 38.000000 | 7664.000000 | 12.500000 | 264.000000 | 75.000000 |

Summary statistics can only reveal so much

# Visualization

**Anscombe's Quartet**

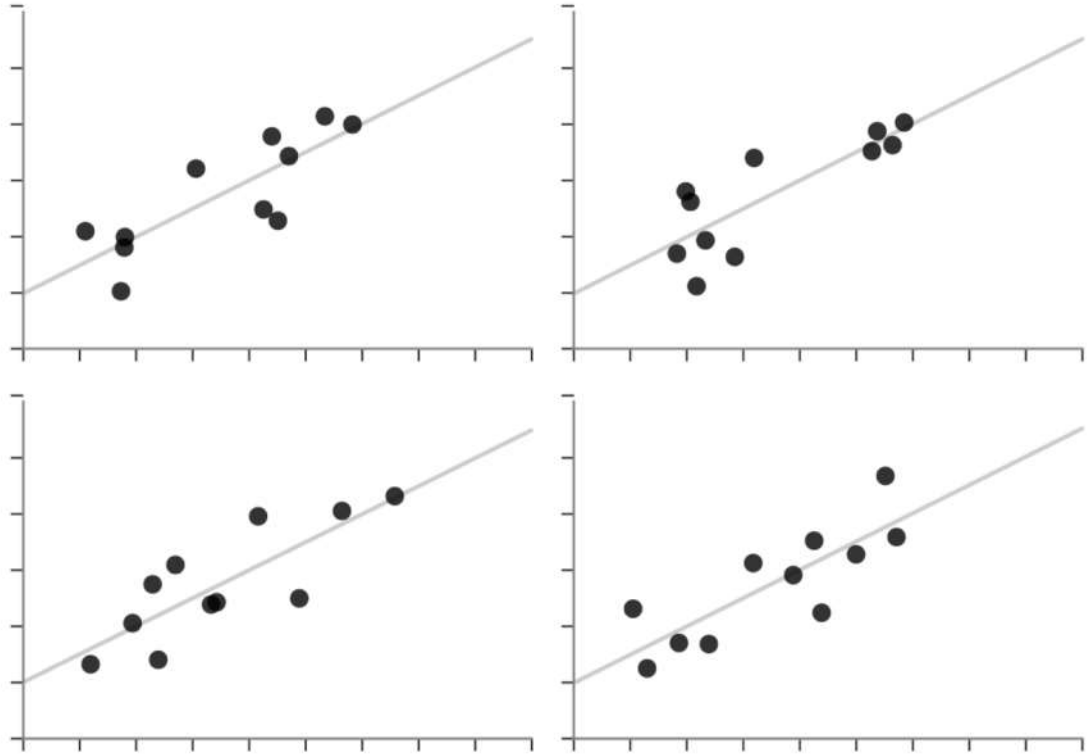
Each dataset has the same summary statistics (mean, standard deviation, correlation), and the datasets are *clearly different*, and *visually distinct*.




**Same stats** do not imply same graphs

**Unstructured Quartet**

Each dataset here also has the same summary statistics. However, they are not *clearly different* or *visually distinct*.



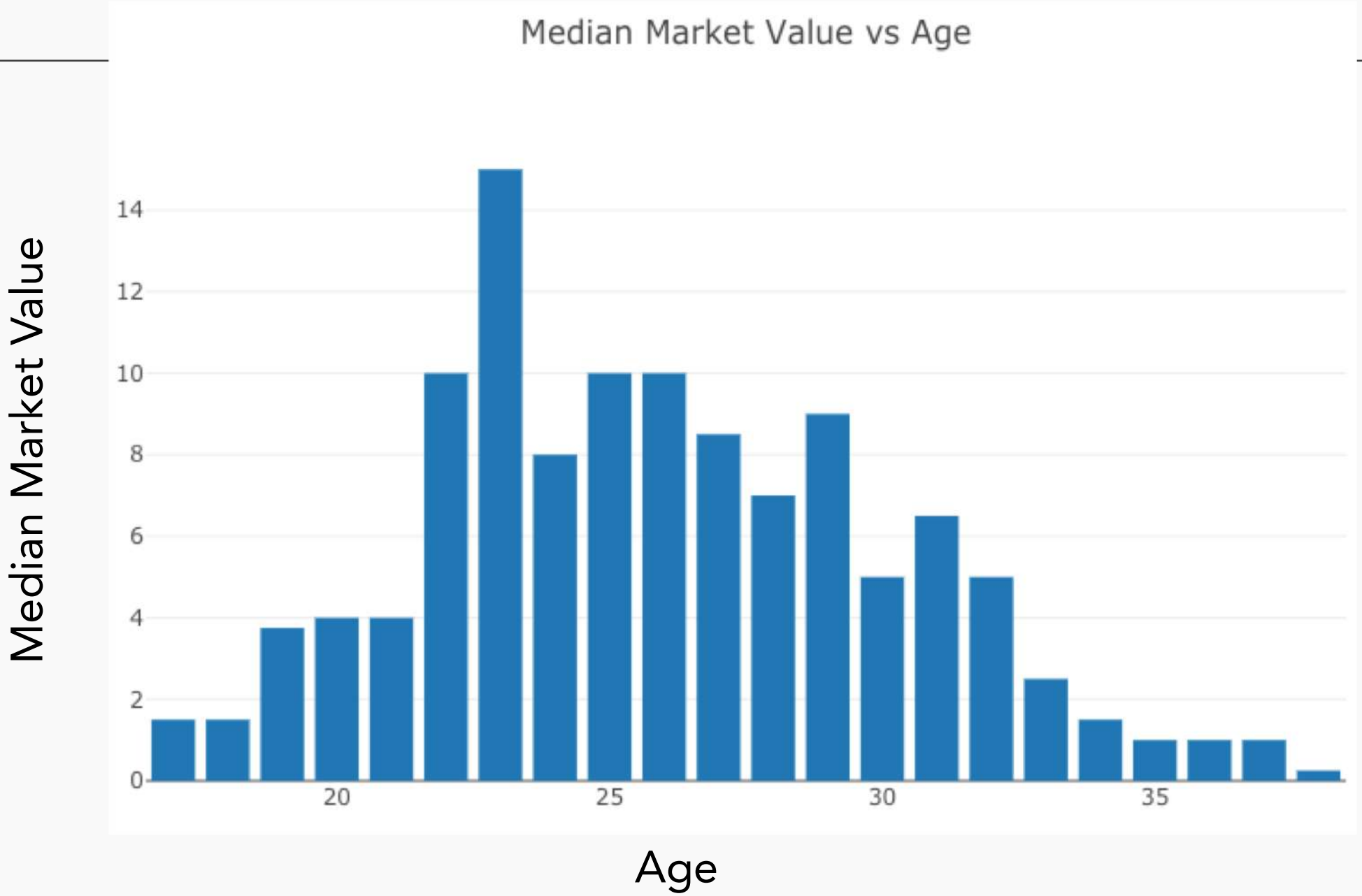
**Same graphs** do not imply same stats

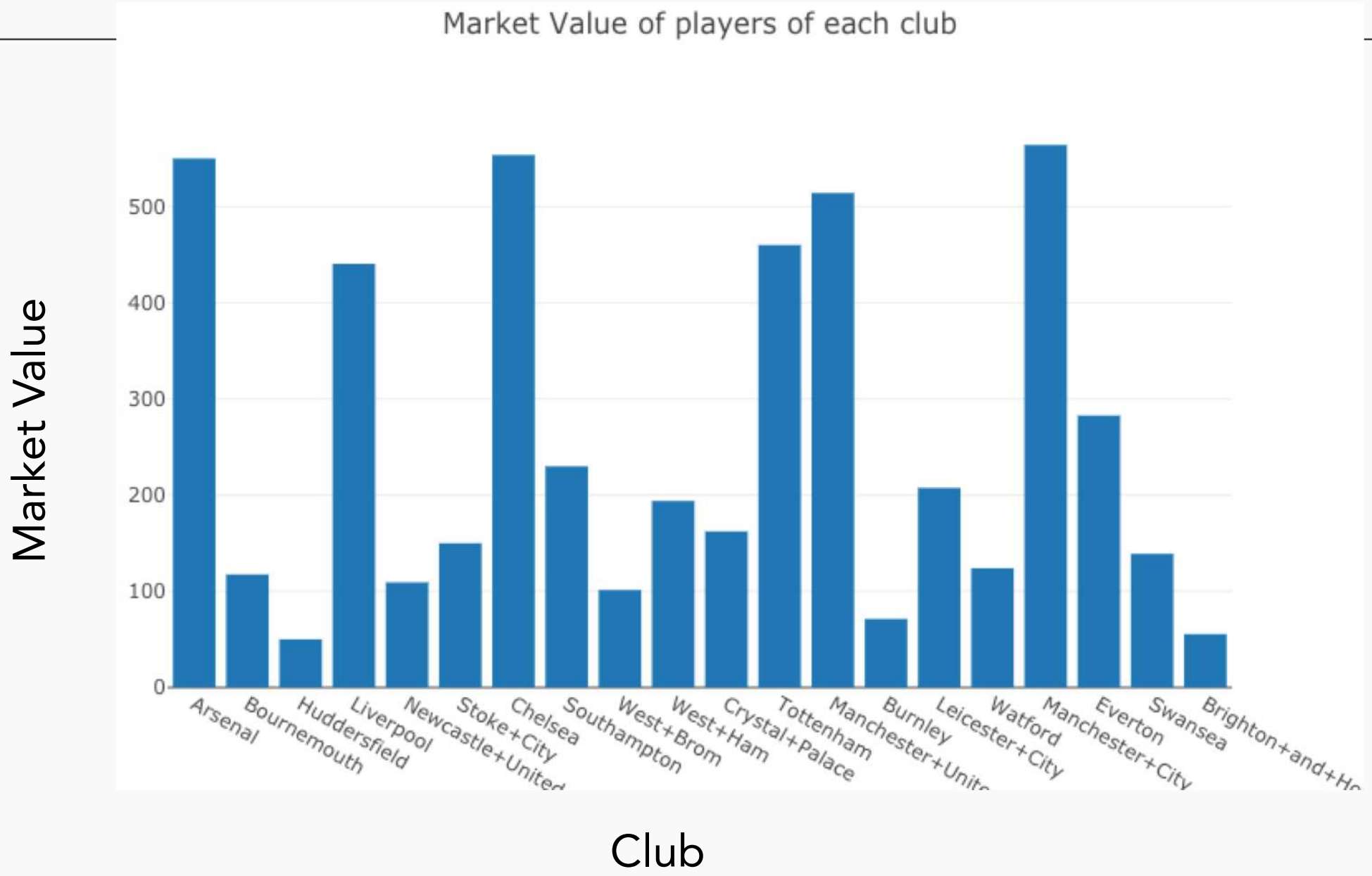
# Visualization

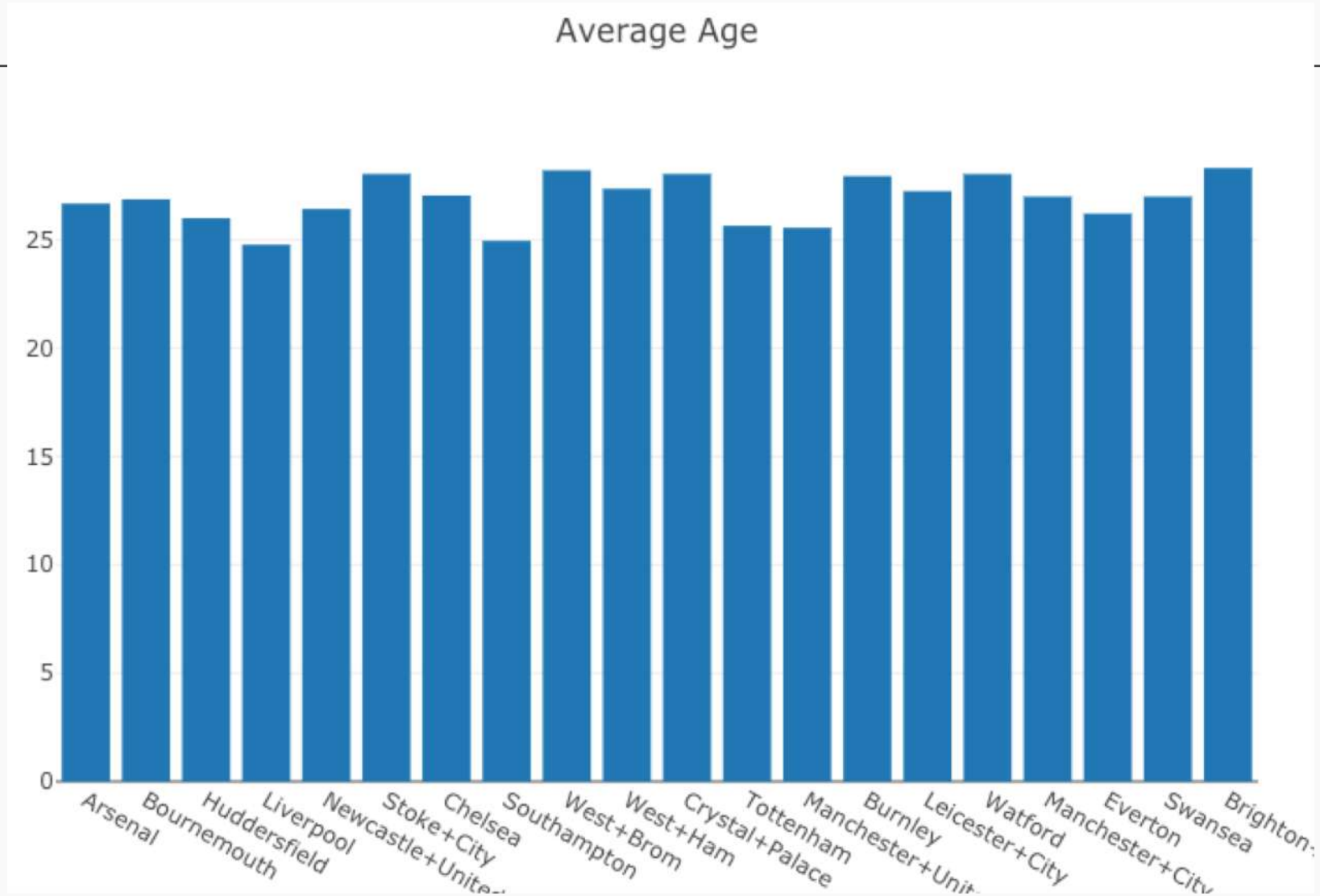
Visualization is incredibly important, both for EDA and for communicating your results to others.

Visualization packages will be used throughout the semester.


Mean Market Value vs Age
Mean Market Value
16
14
12
10
8
6
4
2
0
20
25
30
35
Age


Median Market Value vs Age
Median Market Value
14
12
10
8
6
4
2
0
20
25
30
35
Age


Market Value of players of each club
Market Value
500
400
300
200
100
0
Arsenal
Bournemouth
Huddersfield
Liverpool
Newcastle+United
Stoke+City
Chelsea
Southampton
West+Brom
West+Ham
Crystal+Palace
Tottenham
Manchester+Unite
Burnley
Leicester+City
Watford
Manchester+City
Everton
Swansea
Brighton+and+He
Club


Average Age
Market Value
25
20
15
10
5
0
Arsenal
Bournemouth
Huddersfield
Liverpool
Newcastle+Unite
Stoke+City
Chelsea
Southampton
West+Brom
West+Ham
Crystal+Palace
Tottenham
Manchester+Unit
Burnley
Leicester+City
Watford
Manchester+City
Everton
Swansea
Brighton
Club

# Useful PANDAS functions

- .read_csv() # loads a .csv file

**Accessing/processing:**

- df[“column_name”]
  - .max(), .min(),
  - .idxmax(), .idxmin()
- <dataframe> <conditional>
- .loc[] – label-based accessing
- .iloc[] – index-based accessing
- .sort_values()
- .isnull(), .notnull()
- .dropna()
- .any()
- .values() E.g., df[‘column’].values()
  - (df[‘name’] == “Chris”).any()
- [0:3] *# grab the first 3 rows of the DataFrame*

**Grouping/Splitting/Aggregating:**

- .groupby(),
- .get_groups()
- .drop()
- .merge(), .concat() .aggregate()
- .append()
- .sum() .median() .mean()

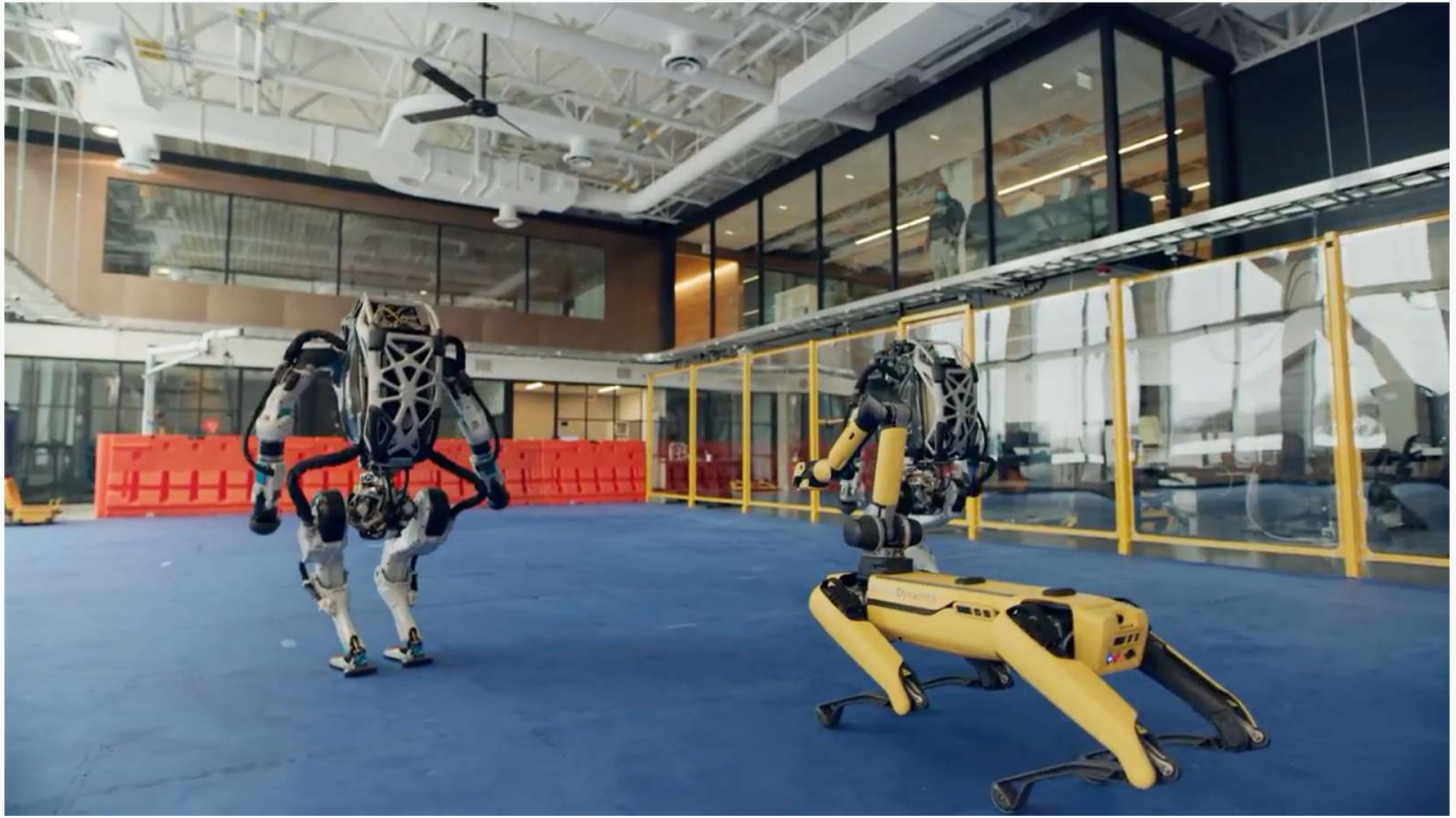
**High-level viewing:**

- .head() – first N observations
- .tail() – last N observations
- .describe() – statistics of the quantitative data
- .dtypes – the data types of the columns

.columns – names of the columns

.shape – the # of (rows, columns)

# Let’s get started!



https://www.youtube.com/watch?v=fn3KWM1kuAw