PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Introduction to Data Science

MODULE II – PART I

Data Exploration

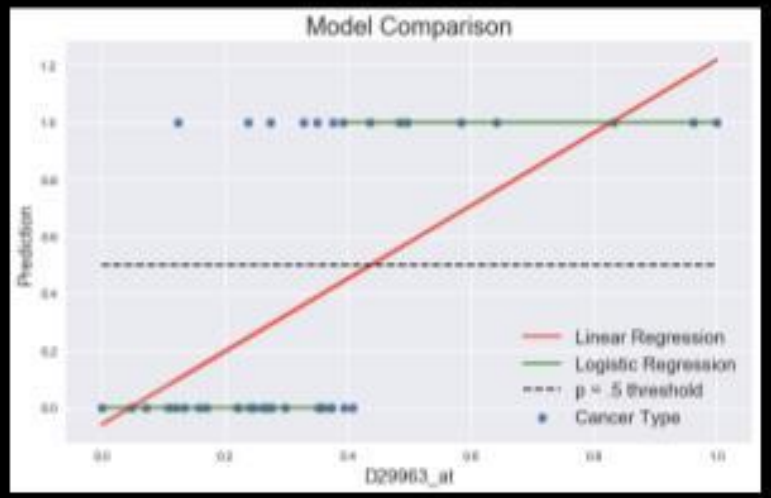
Prof Sergio Serra e Jorge Zavaleta

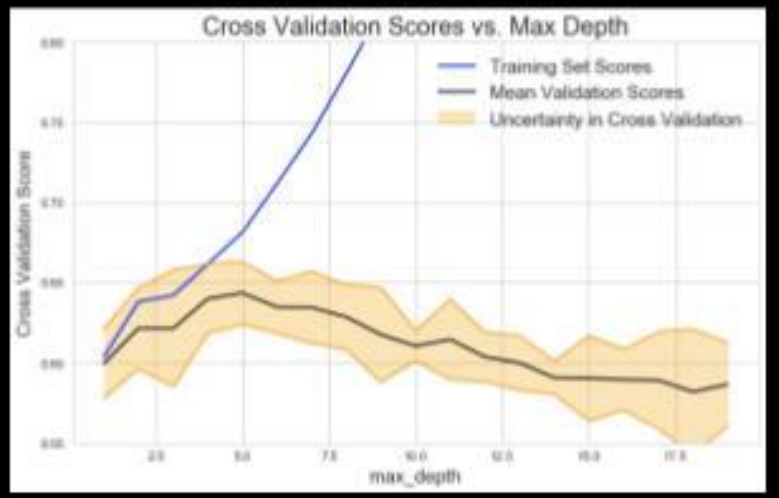
what my friends think I do

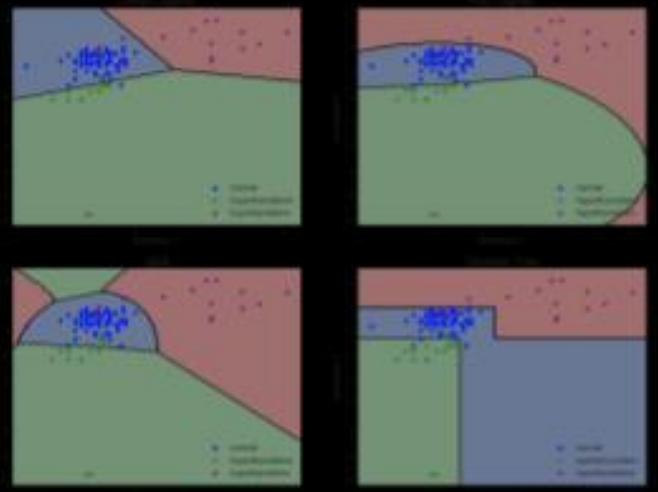
what my family thinks I do

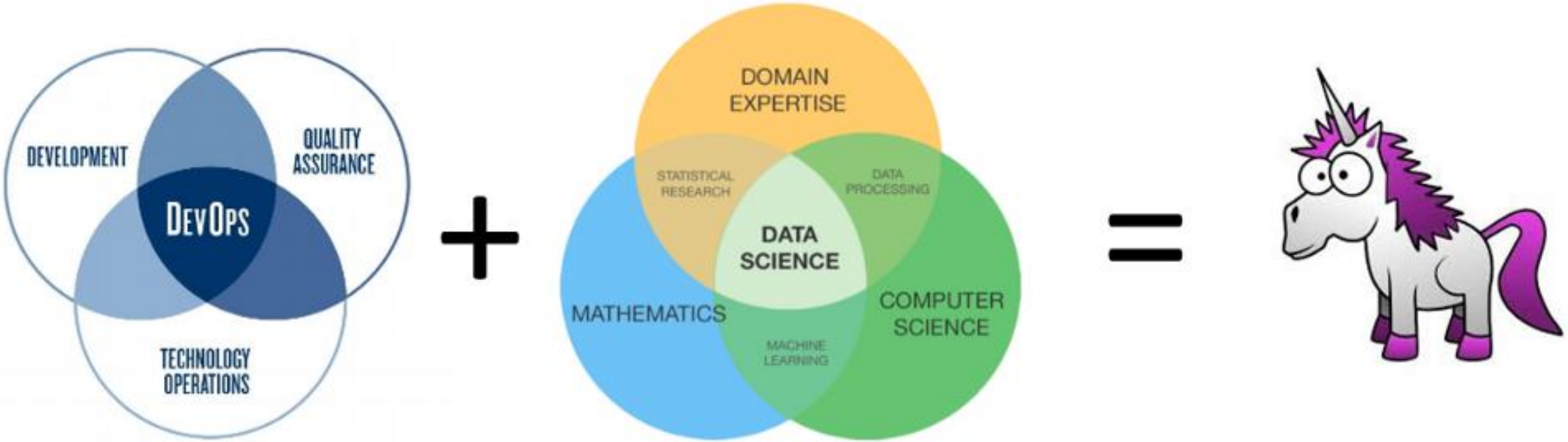
what society thinks I do

what I actually (will) do in Data Science









Fonte: https://www.kdnuggets.com/2018/07/devops-data-scientists-taming-unicorn.html

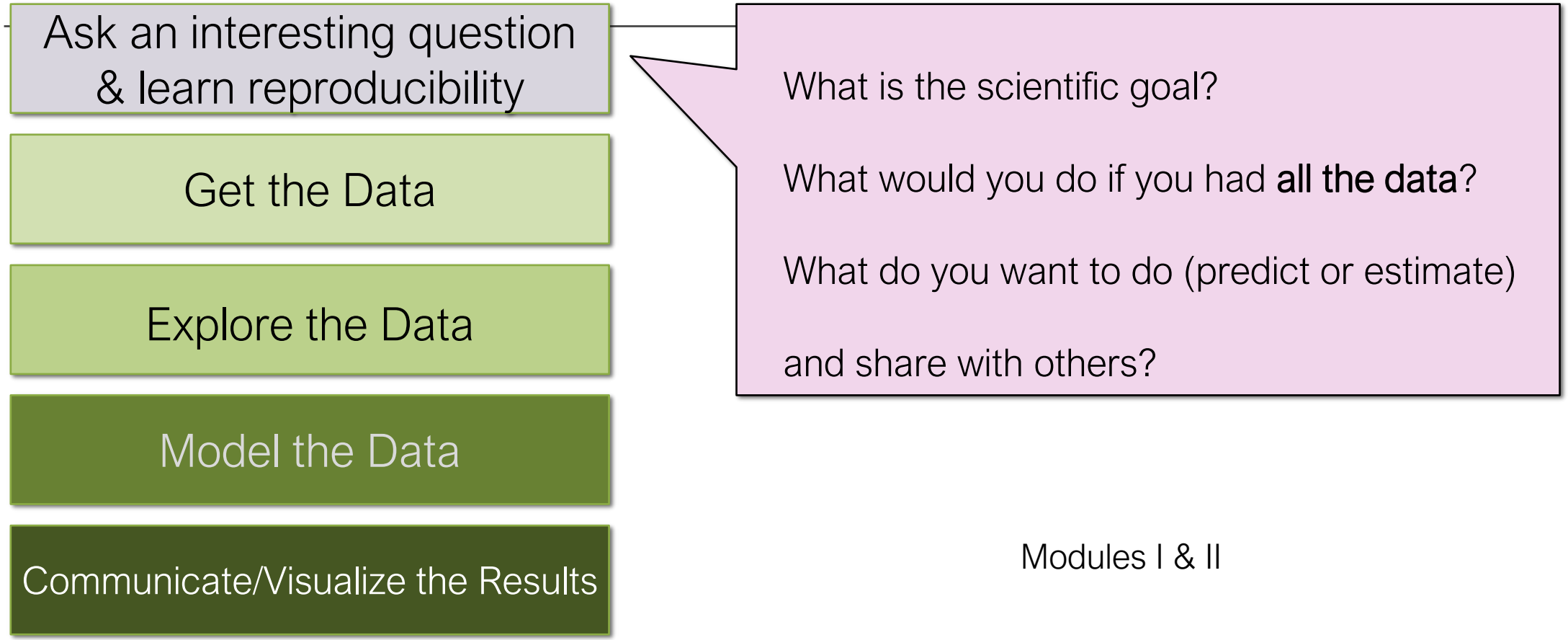
What? The Data Science Process

The Data Science Process is similar to the scientific process - one of observation, model building, analysis and conclusion:

- Ask questions
- Data Collection
- Data Exploration
- Data Modeling
- Data Analysis
- Visualization and Presentation of Results
- Persist and Share Knowledge with Others

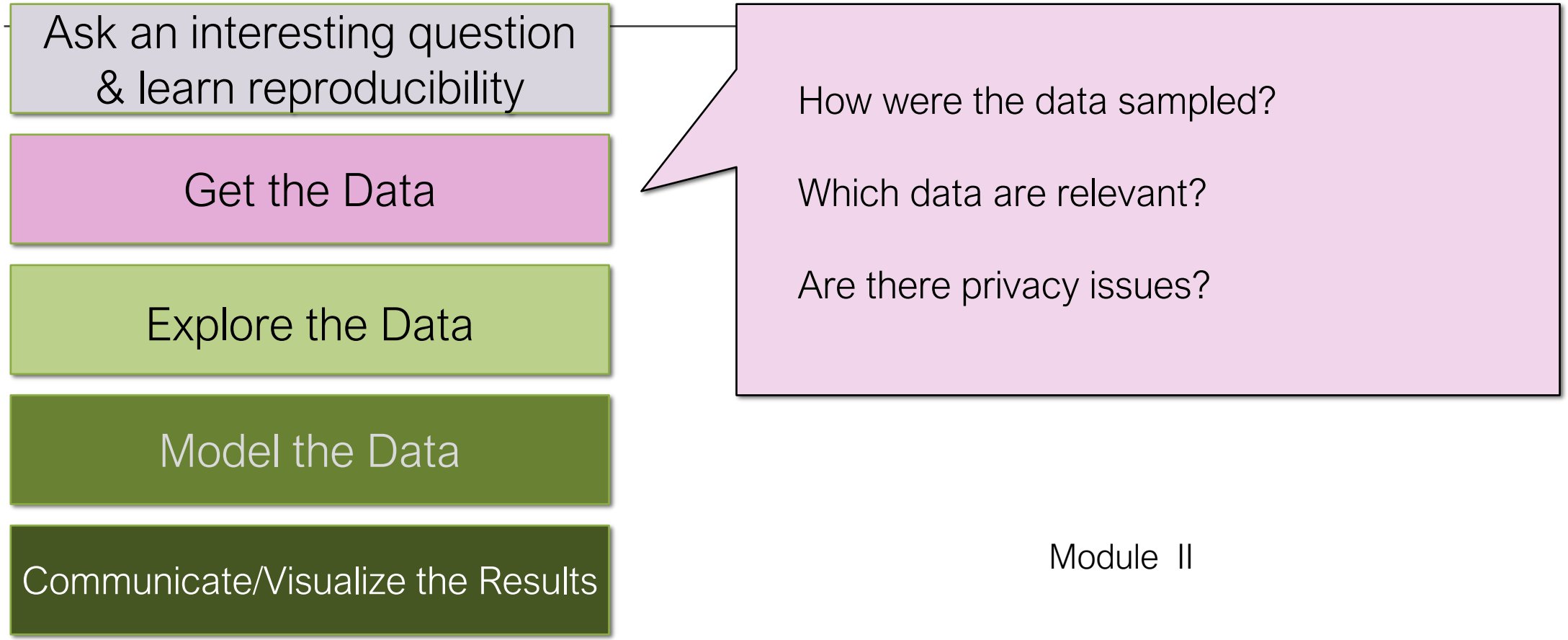
Note: This process is by no means linear!

What? The Data Science Process



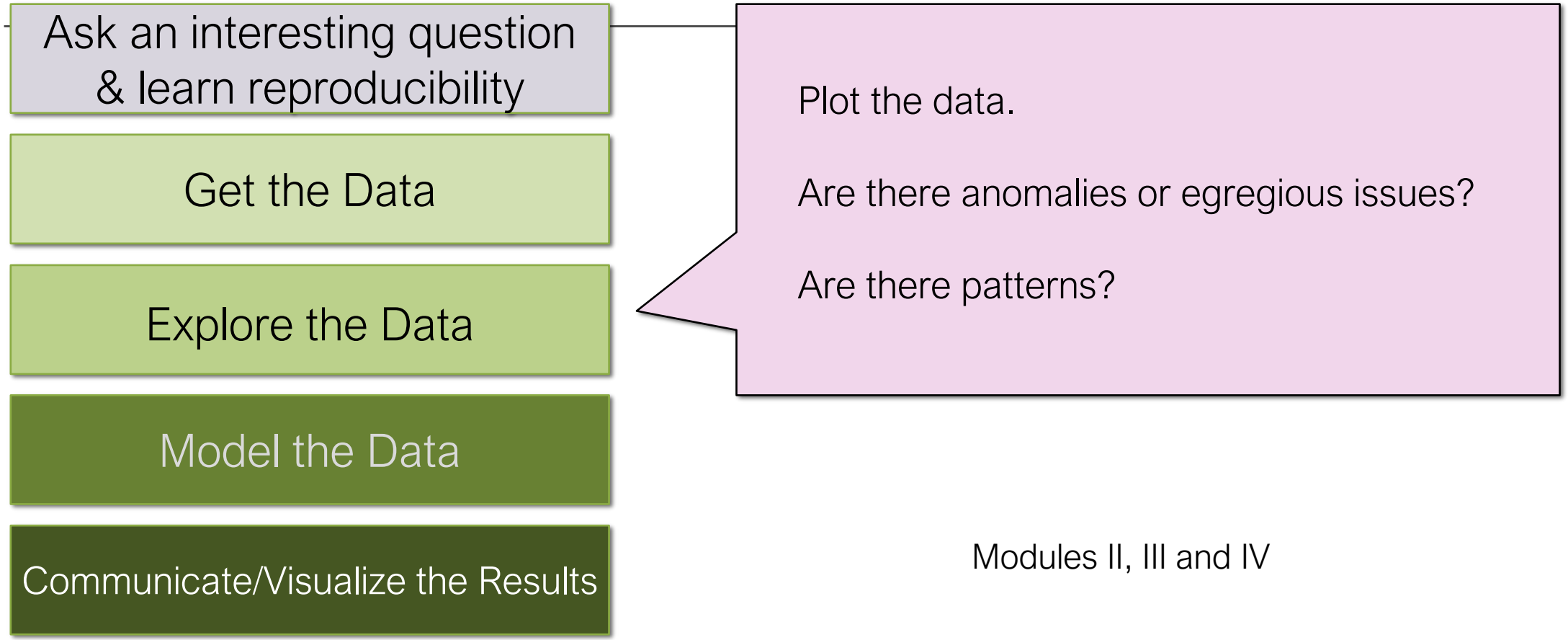
Modules I & II

What? The Data Science Process



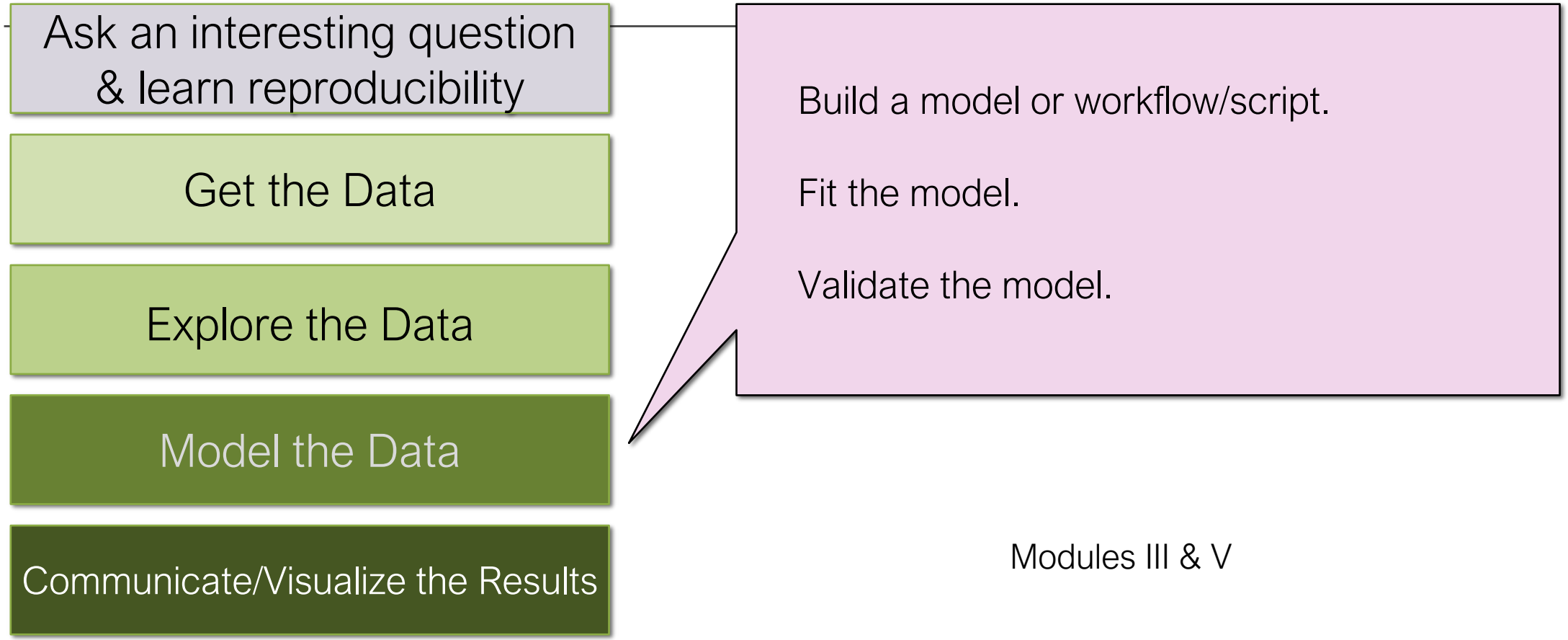
Module II

What? The Data Science Process



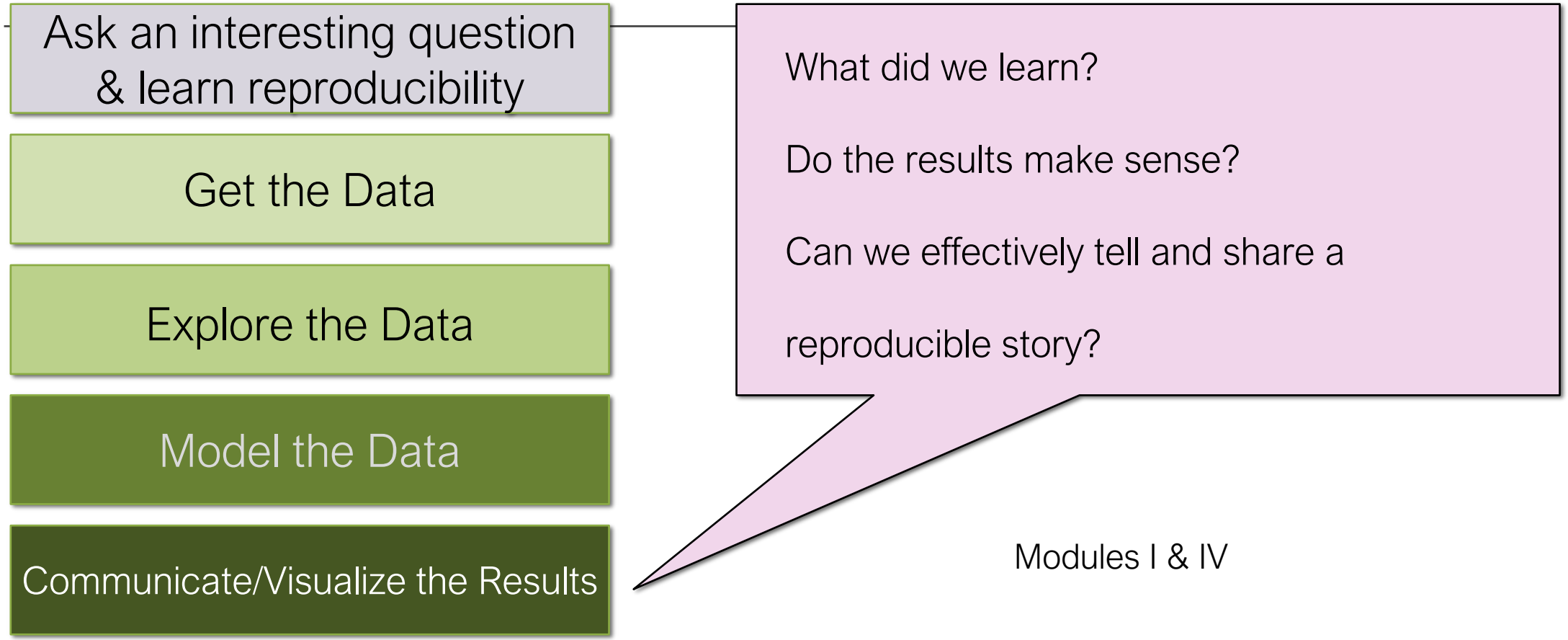
Modules II, III and IV

What? The Data Science Process



Modules III & V

What? The Data Science Process



Modules I & IV

Where do data come from?

- **Internal sources:** already collected by or is part of the overall data collection of your organization.
  - For example: **business-centric data** available in the organization DB to record day to day operations; **scientific or experimental data** get from an essay.
- **Existing External Sources:** available in ready to read format from an outside source for free or for a fee.
  - For example: public government databases, stock market data, sports, COVID-19.
- **External Sources Requiring Collection Efforts**: available from external source but acquisition requires special processing.
  - For example: data appearing only in print form, or data on websites.

Ways to gather online data

How to get data generated, published or hosted online?

- **API (Application Programming Interface):** Using a pre-builtin set of functions developed by a company to access their services. Often pay to use.
  - For example: Google Map API, Facebook API, Twitter API
- **RSS (Rich Site Summary):** summarizes frequently updated online content in standard format. Free to read if the site has one.
  - For example: news-related sites, blogs
- **Web scraping (crawling):** using software, scripts or by-hand extracting data from what is displayed on a page or what is contained in the HTML file (often in tables).

Web scraping

- Why do it?
  - Older government or smaller news sites might not have APIs for accessing data, OR publish RSS feeds or have databases for download. OR, you don't have $$ to pay to use the API or the database.
- Should you do it?
  - You just want to explore: Are you violating their terms of service? Privacy concerns for website and their clients?
  - You want to publish your analysis or product: Do they have an API or fee that you are bypassing? Are they willing to share this data? Are you violating their terms of service? Are there privacy concerns?
- How do you do it?
  - See HW1 (ByHand and beautifulsoup)

Data Types

What kind of values are in your data (data types)?

Simple or atomic:

- **Numeric:** integers, floats
- **Boolean:** binary or true false values
- **Strings:** sequence of symbols

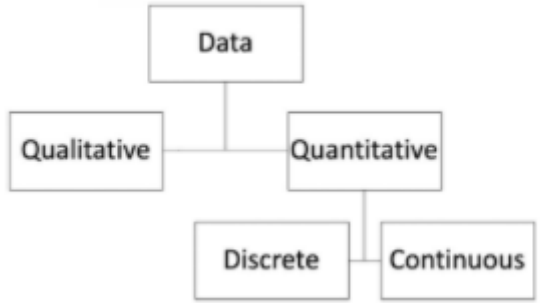
Compound, composed of a bunch of atomic types:

- **Date and time:** compound value with a specific structure
- **Lists:** a list is a sequence of values
- **Dictionaries:** A dictionary is a collection of key-value pairs, a pair of values *x* : *y* where *x* is usually a string called the key representing the “name” of the entry, and *y* is a value of any type.

Types of Data

It is important to distinguish between **classes of variables or attributes** based on the **type of values** they assume.

- **Quantitative variable:** is numerical and can be either:
  - **Discrete -** finite number of values are possible in any bounded interval.
    - Example: “Number of brothers” is a discrete variable
  - **Continuous -** infinite number of values are possible in any bounded interval.
    - Example: “Height of the brothers” is a continuous variable
- **Categorical variable:** No inherent order among the values
  - Example: “What kind of pet you have” is a categorical variable



Data storage

How is your data represented and stored (data structures)?

- **Tabular Data:** a dataset that is a two-dimensional table, where each row typically represents a single data record, and each column represents one type of measurement
  - Example: txt, csv, dat, xlsx, etc...
- **Structured Data:** each data record is presented in a form of a [possibly complex and multi-tiered] dictionary
  - Example: json, xml, etc…

Semistructured Data: not all records are represented by the same set of keys or some data records are not represented using the key-value pair structure.

Common Data Issues

Common issues with data and datasets:

- Missing provenance: Who, When, Where, How it was produced
- Missing values: how do we fill in?
- Wrong values: how can we detect and correct?
- Messy or unknown format
- Not usable data: the data cannot answer the asked question
  - Example: meteorological data, health data, bioinformatics data
- …
- ….
- Errors in data sampling

Example… Messy data

We're measuring individual deliveries of COVID-19 Vaccines; the variables are Time, Day, Number of shots.

	Friday	Saturday	Sunday
Morning	15	158	10
Afternoon	2	90	20
Evening	55	12	45

Problem: Each column header represents a single value rather than a variable. Row headers are "hiding" the Day variable. The values of the variable, "Number of Produce", is not recorded in a single column… (**not to mention the lack of metadata**!)

How to Fix Messy Data?

Solution: Need to reorganize the information to make explicit the event we're observing, and the variables associated to this event.

Operation: Tabularize the data and register the operation.

Common causes of messiness of data are:

- Column headers are values, not variable names
- Variables are stored in both rows and columns
- Multiple variables are stored in one column/entry
- Multiple types of experimental units stored in same table

ID	Time	Day	Number
1	Morning	Friday	15
2	Morning	Saturday	158
3	Morning	Sunday	10
4	Afternoon	Friday	2
5	Afternoon	Saturday	9
6	Afternoon	Sunday	20
7	Evening	Friday	55
8	Evening	Saturday	12
9	Evening	Sunday	45

In general, we want each file to correspond to a dataset, each column to represent a single variable and each row to represent a single observation.

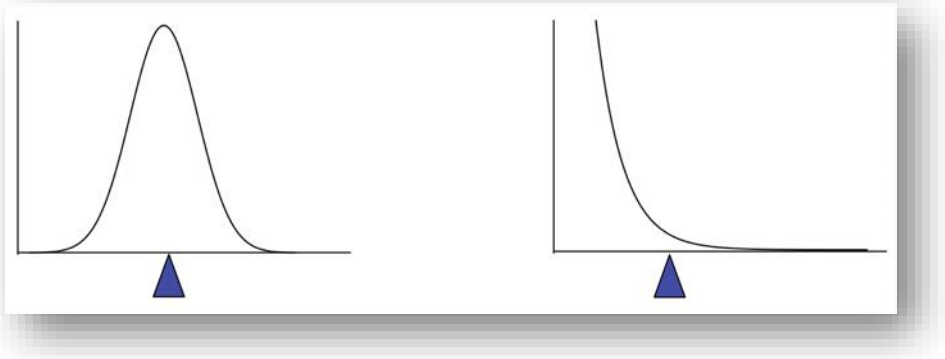
Basics of Sampling

- **Population** is the entire set of objects or events under study.
  - Population can be hypothetical “all students” or all students in this class.
- **Sample** is a “representative” subset of the objects or events under study.
  - Needed because it’s impossible or intractable to obtain or compute with population data.
- **Bias** is a disproportionate weight in favor of or against an idea or thing. Biases in samples is a problem
  - **Selection bias**: some subjects or records are more likely to be selected
  - Volunteer/**nonresponse** bias: subjects or records who are not easily available are not represented
  - There are **many types of data bias** : e.g. Selection Bias, Loss Aversion, Framing Bias, Anchoring Bias, etc...

Basics of Descriptive Statistics – **Mean**

Sample mean - the mean of a set of n observations of a variable is denoted $\bar{x}$ and is defined as:

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i$$



- The mean describes what a "typical" sample value looks like, or where is the "center" of the distribution of the data.
  - **Key theme:** there is always uncertainty involved when calculating a sample mean to estimate a population mean.

Basics of Descriptive Statistics – **Median**

The median of a set of n number of observations in a sample, ordered by value, of a variable is is defined by

$$\text{Median} = \begin{cases} x_{(n+1)/2} & \text{if } n \text{ is odd} \\ \frac{x_{n/2} + x_{(n+1)/2}}{2} & \text{if } n \text{ is even} \end{cases}$$

Example (already in order):

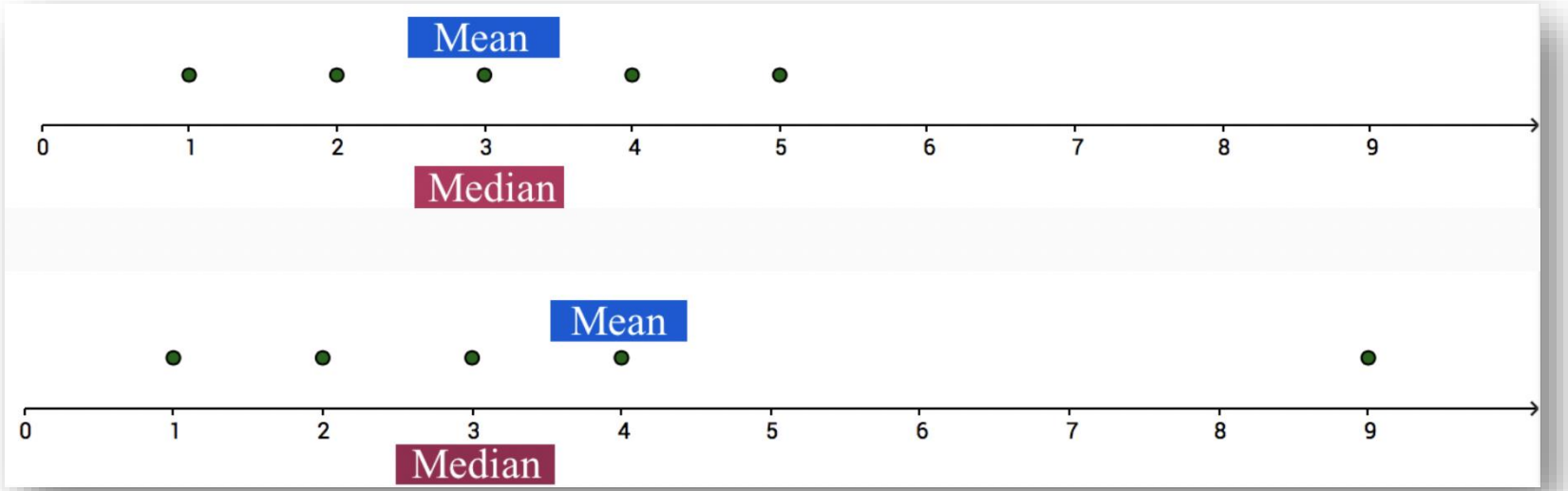
Ages: 17, 19, 21, 22, 23, 23, 23, 38

Median = (22+23)/2 = 22.5

- The median also describes what a typical observation looks like, or where is the center of the distribution of the sample of observations.

Basics of Descriptive Statistics
Mean x Median

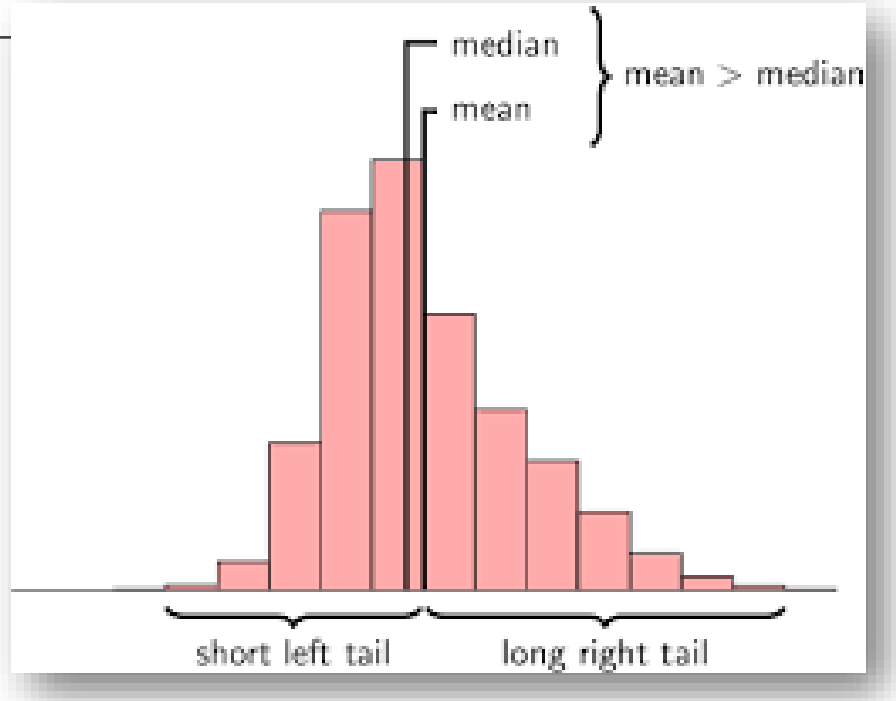
The mean is sensitive to extreme values (**outliers**)



Basics of Descriptive Statistics
Mean x Median x Skewness

Skewness is a measure of the asymmetry of the probability distribution of a real-valued or random variable about its mean.

The skewness value can be positive, zero, negative, or undefined.



Example: The above distribution is called **right-skewed** since the mean is greater than the median.

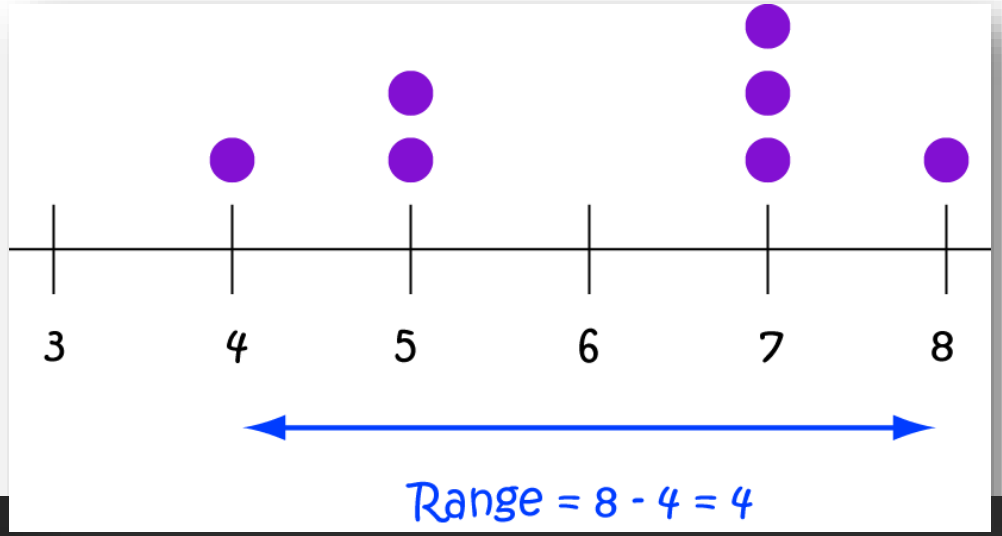
Note: **skewness** often “follows the longer tail”.

Basics of Descriptive Statistics – Measures of Spread: **Range**

The spread of a sample of observations measures how well the mean or median describes the sample.

One way to measure spread of a sample of observations is via the range.

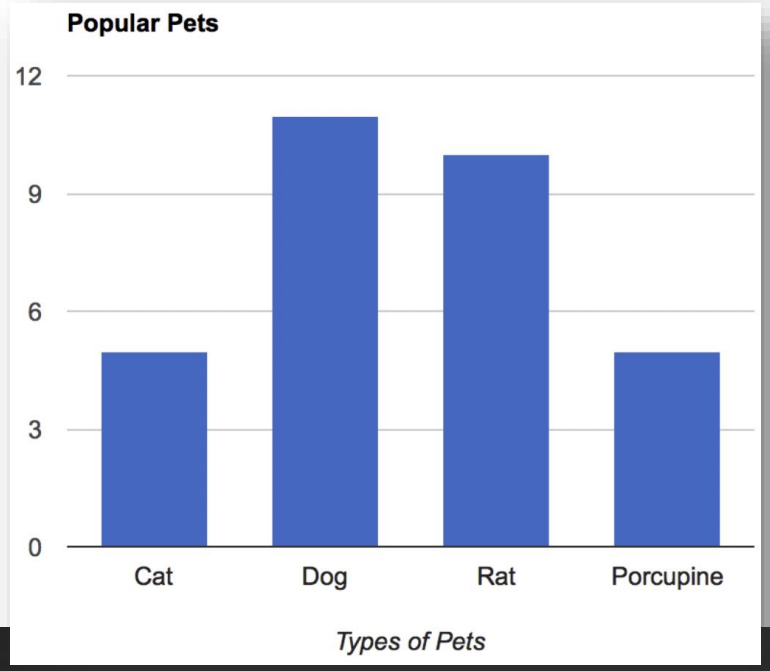
Range = Maximum Value - Minimum Value



Basics of Descriptive Statistics – **Mode**

The mode is the value that appears most often in a set of data values. For **categorical variables,** neither mean or median make sense. Why?

- The mode might be a better way to find the most "representative" value.

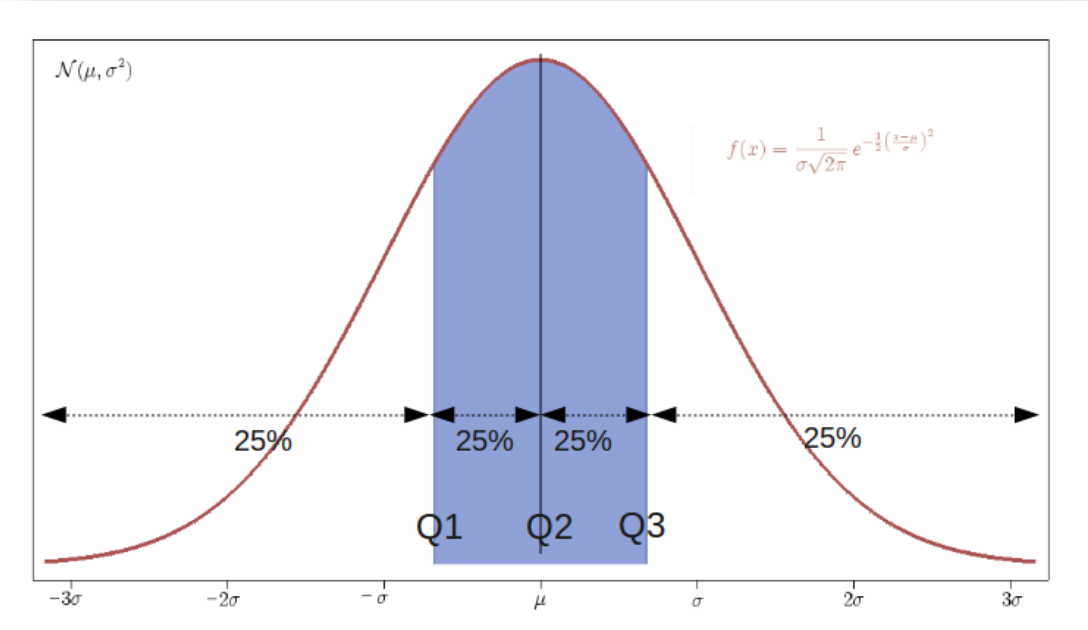


- Like the **mean** and **median**, mode is a way of expressing, in a single number, important information about a random variable or a population.
  - The numerical value of the mode is the same as that of the mean and median in a normal distribution and may be very different in highly skewed distributions.

Basics of Descriptive Statistics – Measures of Spread: **Quantiles**

Quantiles are cut points dividing the range of a (probability) distribution into continuous intervals with equal probabilities or dividing the observations in a sample in the same way.

- Example: a quantile determines how many values in a distribution are above or below a certain limit). In general, the X% th percentile is the point where X% of the data is below that value and (1 – X)% of the data points are over that value.



Basics of Descriptive Statistics – Measures of Spread: **Variance**

The variance of the sample, denoted s^2, measures how much on average the sample values deviate from the mean value

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n} |x_i - \bar{x}|^2$$

Note: the term $|x_i - \bar{x}|$ measures the amount by which each x_i deviates from the mean $\bar{x}$. Squaring these deviations means that s^2 is sensitive to extreme values (**outliers**).

Note: s^2 doesn't have the same units as the x_i

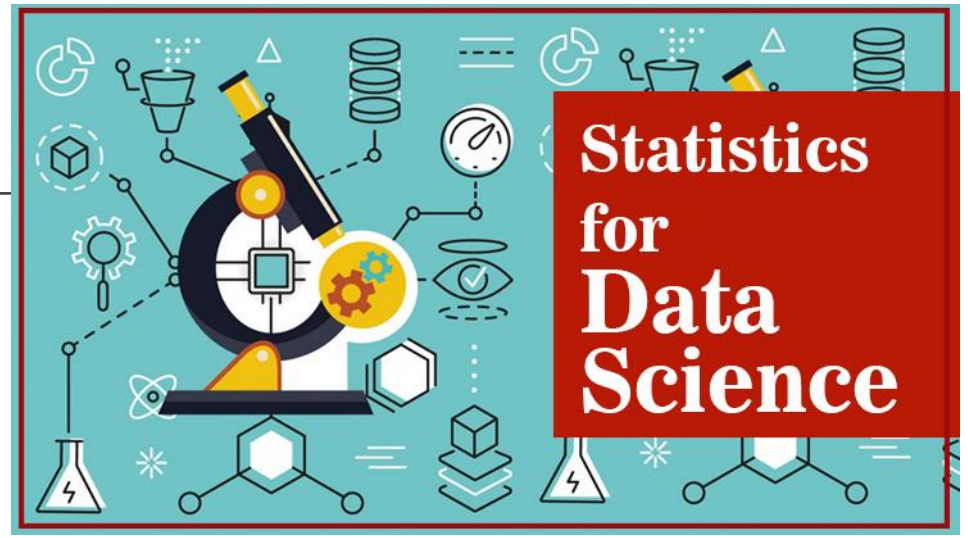
Basics of Descriptive Statistics – Measures of Spread: **Standard Deviation**

The standard deviation of a sample, denoted s, is the square root of the variance

$$s = \sqrt{s^2} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} |x_i - \bar{x}|^2}$$

Note: s does have the same units as the x_i.

Hands on...



NumPy

TensorFlow

matplotlib
Version 3.3.4

NumPy - Numerical Python

NumPy is the fundamental package required for high performance computing and data analysis

NumPy is important for numerical computations in Python is because it is designed for efficiency on large arrays of data.

NumPy provides

- **ndarray** for creating multiple dimensional arrays
- Internally stores data in a contiguous block of memory, independent of other built-in Python objects, use much less memory than built-in Python sequences.
- Standard math functions for fast operations on entire arrays of data without having to write loops
- **NumPy arrays enable you to express batch operations on data without writing any *for* loops. We call this *vectorization* or *vectorized computation.***

NumPy: **ndarray** x **list**

One of the key features of NumPy is its **N-dimensional array object**, or ndarray, which is a fast, flexible container for large datasets in Python.

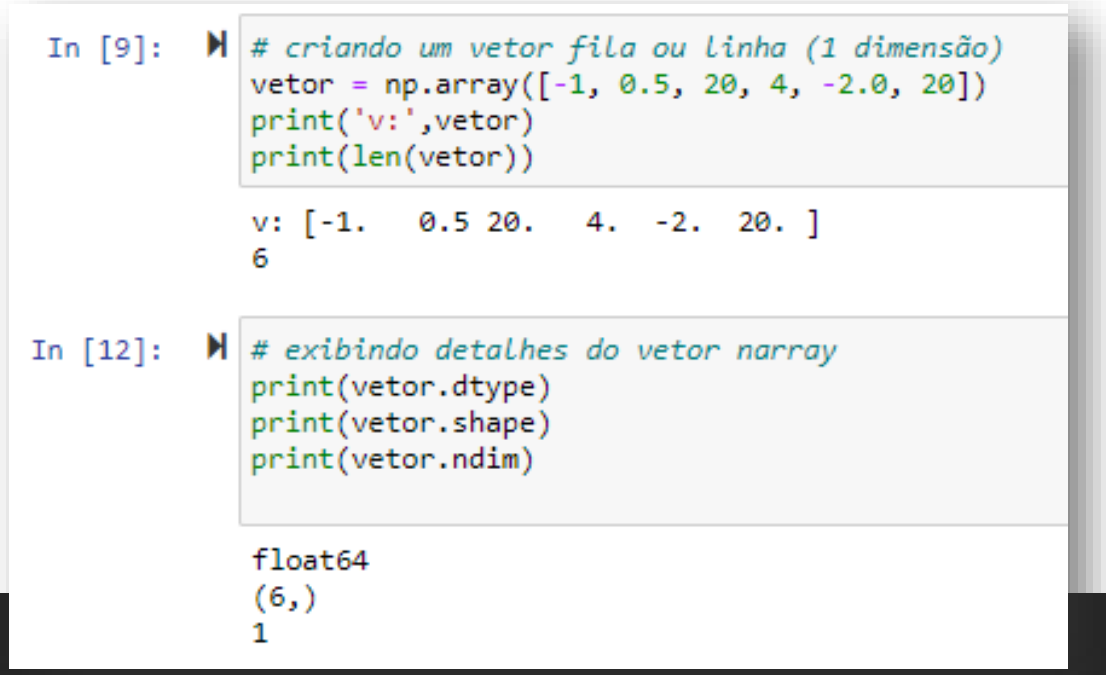
NumPy-based algorithms are generally 10 to 100 times faster (or more) than their pure Python counterparts and use significantly less memory.

```
In [1]:
# importando o pacote
import numpy as np
# version do numpy
print(np.version.version)

1.19.2
```

```
In [5]:
my_arr = np.arange(1000000)        # ndArray
my_list = list(range(1000000))     #Lista

In [6]:
print (my_arr)

[     0      1      2 ... 999997 999998 999999]

In [7]:
print (my_list)

IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

ndarray

- **ndarray** is used for storage of homogeneous data
  - Example: all elements the same type
- Every array must have a **shape** and a **dtype.**
- Supports convenient **slicing**, **indexing** and efficient vectorized computation



```
In [9]:
# criando um vetor fila ou linha (1 dimensão)
vetor = np.array([-1, 0.5, 20, 4, -2.0, 20])
print('v:',vetor)
print(len(vetor))

v: [-1.   0.5 20.   4.  -2.  20. ]
6

In [12]:
# exibindo detalhes do vetor narray
print(vetor.dtype)
print(vetor.shape)
print(vetor.ndim)

float64
(6,)
1
```

Creating ndarrays – vector and matrix

```
# criar vetor coluna
vetor_c = np.array([[-1],[0.5],[20],[4],[-2.0]])
print(vetor_c)
print(vetor_c.ndim)                    #2
print(vetor_c.shape)                   #(5,1)
```

```
[[-1. ]
 [ 0.5]
 [20. ]
 [ 4. ]
 [-2. ]]
2
(5, 1)
```

```
# criando uma matriz
matriz_3x3 = np.array([[4,6,2],[8,-1,0],[3,4,5]])
print(matriz_3x3)
print(matriz_3x3.ndim)                 #2
print(matriz_3x3.shape)                #(3,3)
```

```
[[ 4  6  2]
 [ 8 -1  0]
 [ 3  4  5]]
2
(3, 3)
```

Creating ndarrays – vector and matrix

```
array = np.array([[0,1,2],[2,3,4]])
[[0 1 2]
 [2 3 4]]
```

```
array = np.zeros((2,3))
[[0. 0. 0.]
 [0. 0. 0.]]
```

```
array = np.ones((2,3))
[[1. 1. 1.]
 [1. 1. 1.]]
```

```
array = np.eye(3)
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

```
array = np.arange(0, 10, 2)
[0, 2, 4, 6, 8]
```

```
array = np.random.randint(0,
10, (3,3))
[[6 4 3]
 [1 5 6]
 [9 8 5]]
```

arange is an array-valued version of the built-in Python range function

Arithmatic with NumPy Arrays

Any arithmetic **operations between equal-size arrays** applies the operation element-wise:

```
In [110]: mat = np.array([[1., 2., 3.], [4., 5., 6.]])
          print(mat)

          [[1. 2. 3.]
           [4. 5. 6.]]

In [111]: print(mat * mat)

          [[ 1.  4.  9.]
           [16. 25. 36.]]

In [112]: print(mat * 2)

          [[ 2.  4.  6.]
           [ 8. 10. 12.]]

In [114]: print(mat + (mat *2) -10)

          [[-7. -4. -1.]
           [ 2.  5.  8.]]

In [115]: mat1 = np.array([[1., 2., 3.], [4., 5., 6.]])
          mat2 = np.array([[5, 3, 1.], [-1, 5., 6.]])
          print(mat1 * mat2)

          [[ 5.  6.  3.]
           [-4. 25. 36.]]
```

Indexing and Slicing

One-dimensional arrays are simple; on the surface they act similarly to Python lists:

```
arr = np.arange(10)
print(arr)         # [0 1 2 3 4 5 6 7 8 9]
print(arr[5])     #5
print(arr[5:8]) #[5 6 7]
arr[5:8] = 12
print(arr)      #[ 0 1 2 3 4 12 12 12 8 9]
```

Basics of Descriptive Statistics

In [ ]:
```
# exemplo: narray
peso_kg = np.linspace(55,90,num=40)
print('Peso em Kg:')
print(peso_kg)
```

In [ ]:
```
# funcoes estatisticas
# media aritmetica
media = peso_kg.mean()
print('Média da turma :',media)
# maximo
maximo = peso_kg.max()
print('Máximo da turma :',maximo)
# minimo
minimo = peso_kg.min()
print('Mínimo da turma :',minimo)
# somatorio
soma = peso_kg.sum()
print('Somatorio dos valores da turma :',soma)
# desvio padrao
desv = peso_kg.std()
print('Desvio padrão dos valores da turma :',desv)
vart = peso_kg.var()
print('Variância dos valores da turma :',vart)
```

Extra Content

References

https://numpy.org/doc/stable/contents.html

https://www.tutorialspoint.com/numpy/numpy_advanced_indexing.htm

https://matplotlib.org/stable/tutorials/introductory/usage.html#sphx-glr-tutorials-introductory-usage-py

https://matplotlib.org/stable/gallery/index.html