

# Data Science with Python

Kelly Pierce

Research Associate | Texas Advanced Computing Center

PyLadies Tech and Coding Night 18 November 2019

# a bit about me...

... formally trained as a **biologist**

- (mostly) self-taught Python and R skills in grad school!

... left academia to work as a **data scientist** in private industry

- ask me about big career transitions!

... now work as a **research associate** at the Texas Advanced Computing Center (TACC)

- I help researchers use supercomputers!

# Agenda

- Data science overview
- Data science workflow
- Hands on exercise with Python

# What I assume you know...

- Python experience
  - you know what **variables** are and how to assign them
  - you're familiar with basic **data types** (floats, integers, strings)
- Data experience
  - you've worked with **spreadsheets** before
  - you're comfortable reading **graphs** like scatter plots or histograms
- Statistics experience
  - you're familiar with **descriptive statistics** (mean, standard deviation, etc.)
  - you've seen linear **regression** before

# Data Science Overview

Data science is used to drive  
decision making under  
**uncertainty**

**Python** is a key part of the  
data science toolbox

# Data Science / Machine Learning / Statistics

- **Statistics:** applied mathematics to **describe data** in terms of probability
  - How does this system work?
  - What is the probability of either my observed data or my hypothesis?
- **Machine Learning:** applied statistics to **predict outcomes**
  - What variables are good predictors of the outcome of interest?
  - How does adding new data improve prediction?
- **Data Science:** combine statistics, machine learning, and **domain expertise** to drive decision making.

# Statistical Models for Data

## Classification



*Iris setosa*

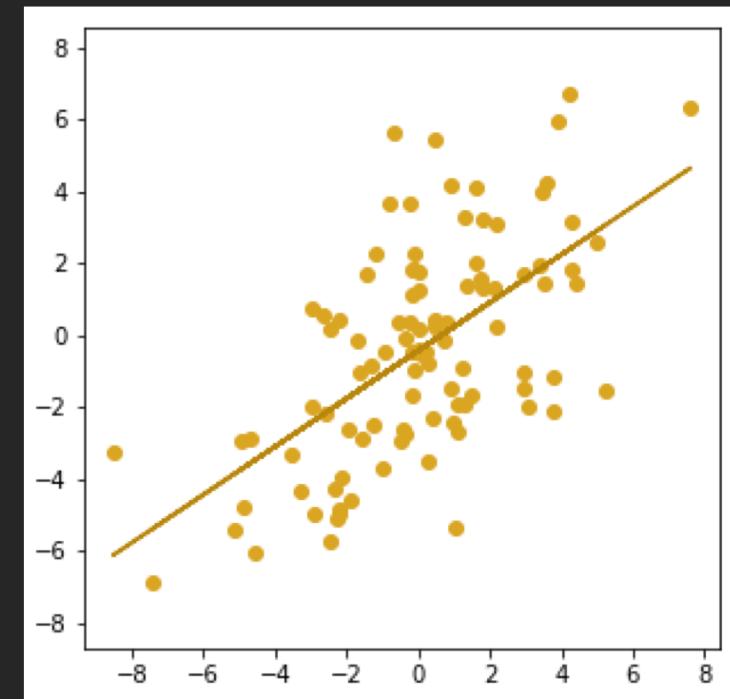


*Iris virginica*



*Iris versicolor*

## Regression



# Statistical Models for Data

## Classification

- Logistic regression
- **Multinomial regression**
- Decision tree
- Random forest
- Support vector machine
- Clustering

## Regression

- Linear regression (many types)
- Random forest

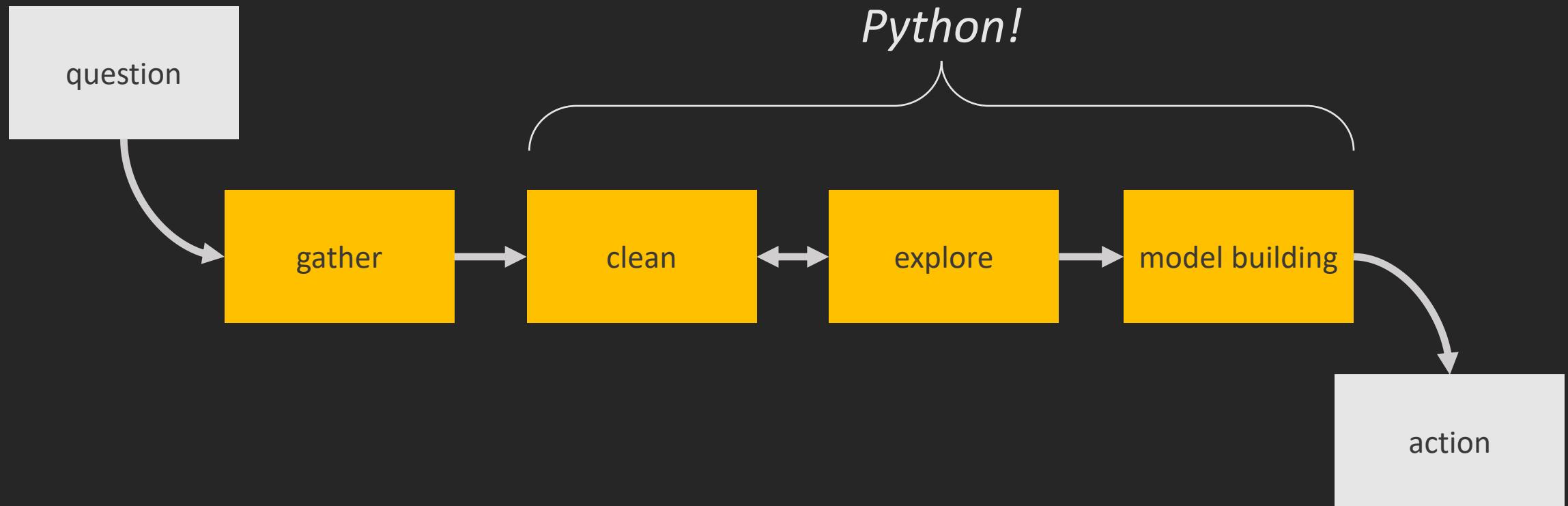
# Machine Learning Models for Data

## Supervised                      Unsupervised

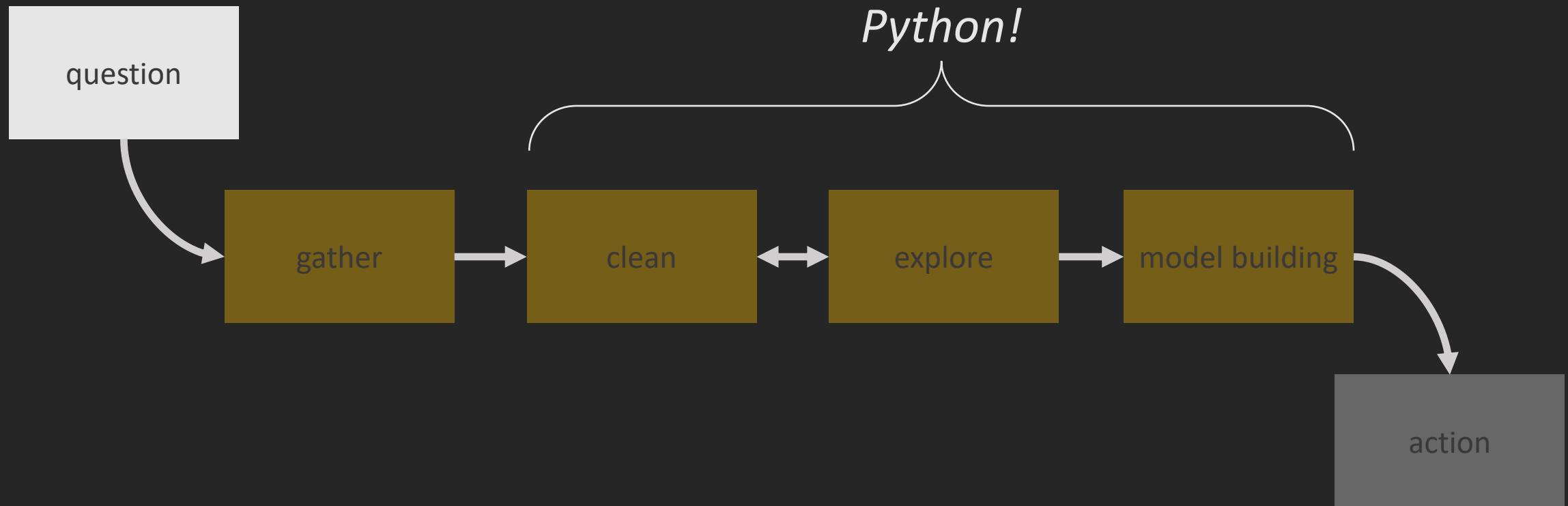
- Labeled data / known outcomes
- Learn a data **model**
- Unlabeled data / unknown outcomes
- Learn data **structure**

# Data Science Workflow

# Data science workflow



# Data science workflow



# Question: Can we classify plant species?



*Iris setosa*



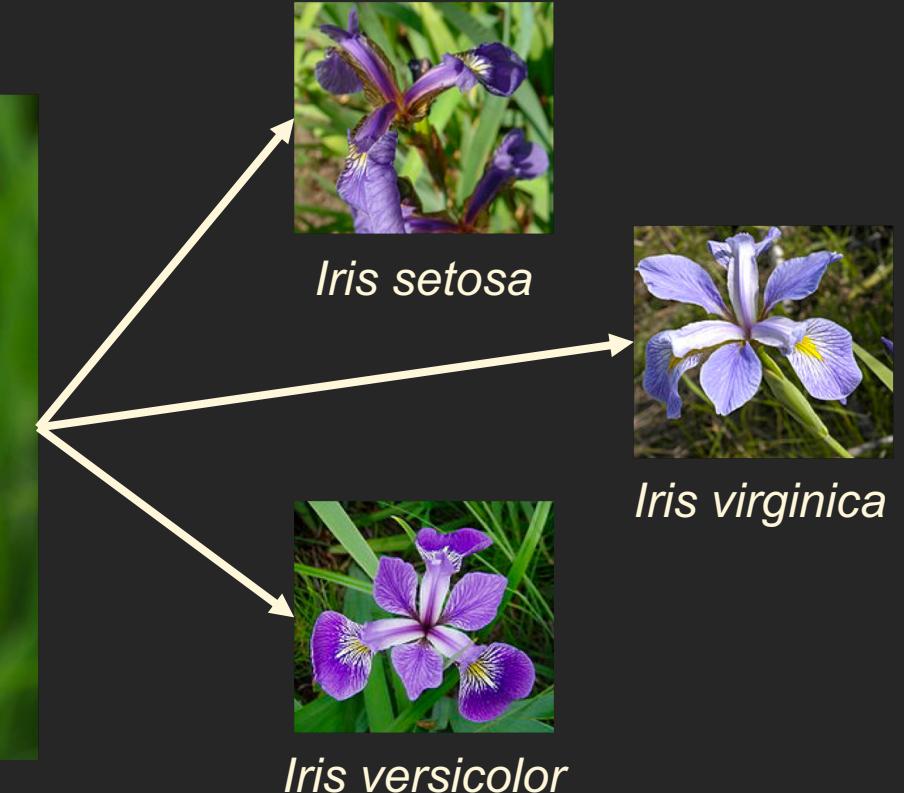
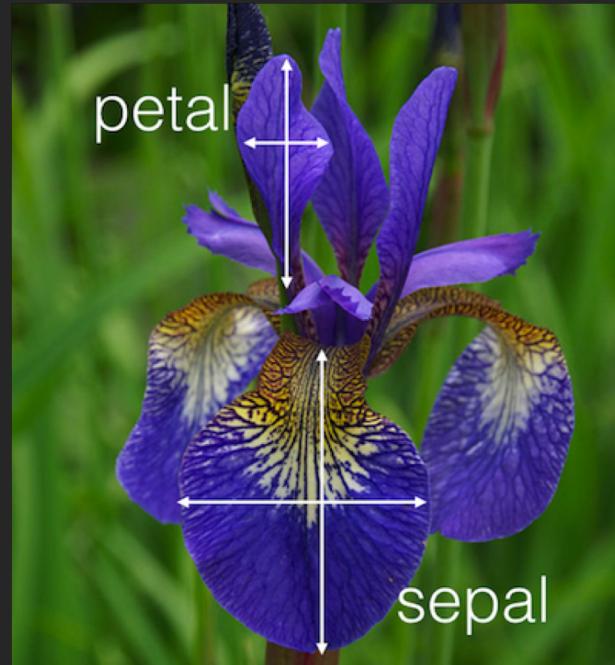
*Iris versicolor*



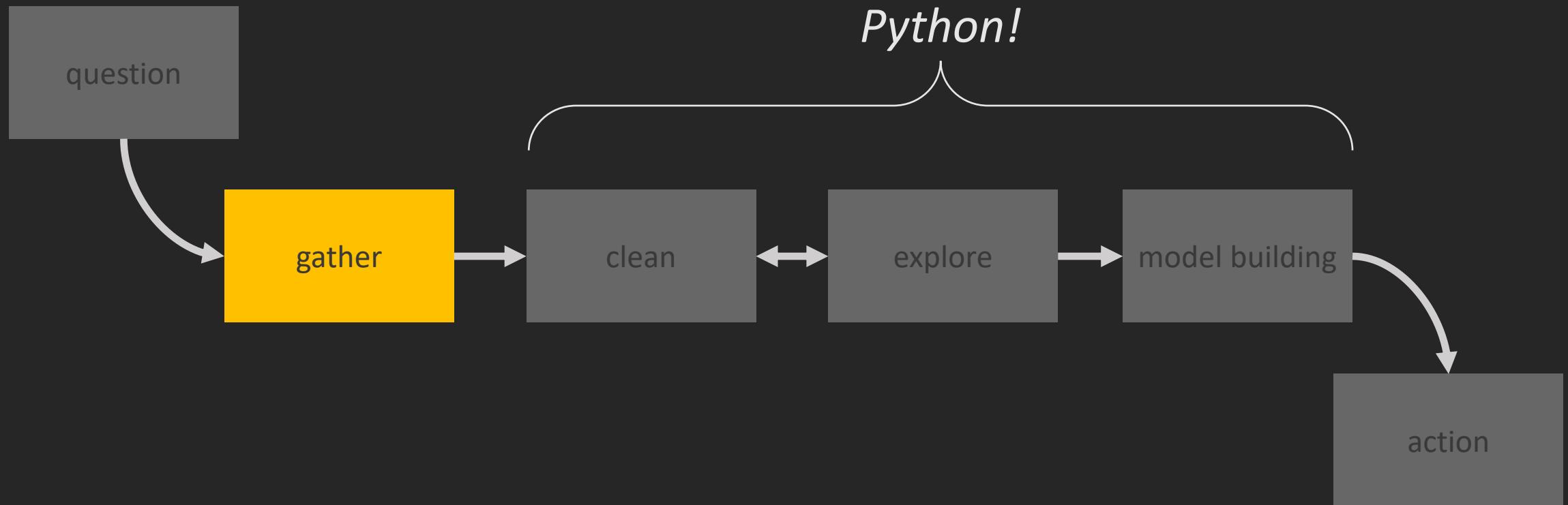
*Iris virginica*

# Question: Can we classify plant species?

Can we identify species of iris by their petal and sepal lengths and widths?

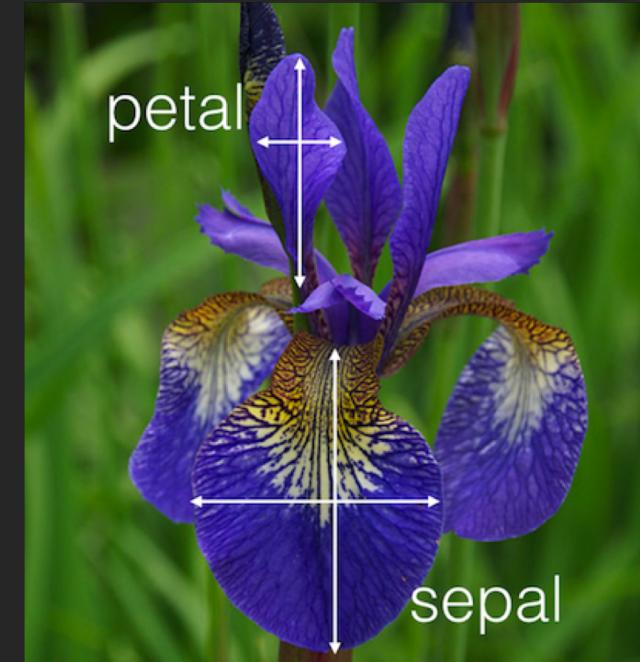


# Data science workflow

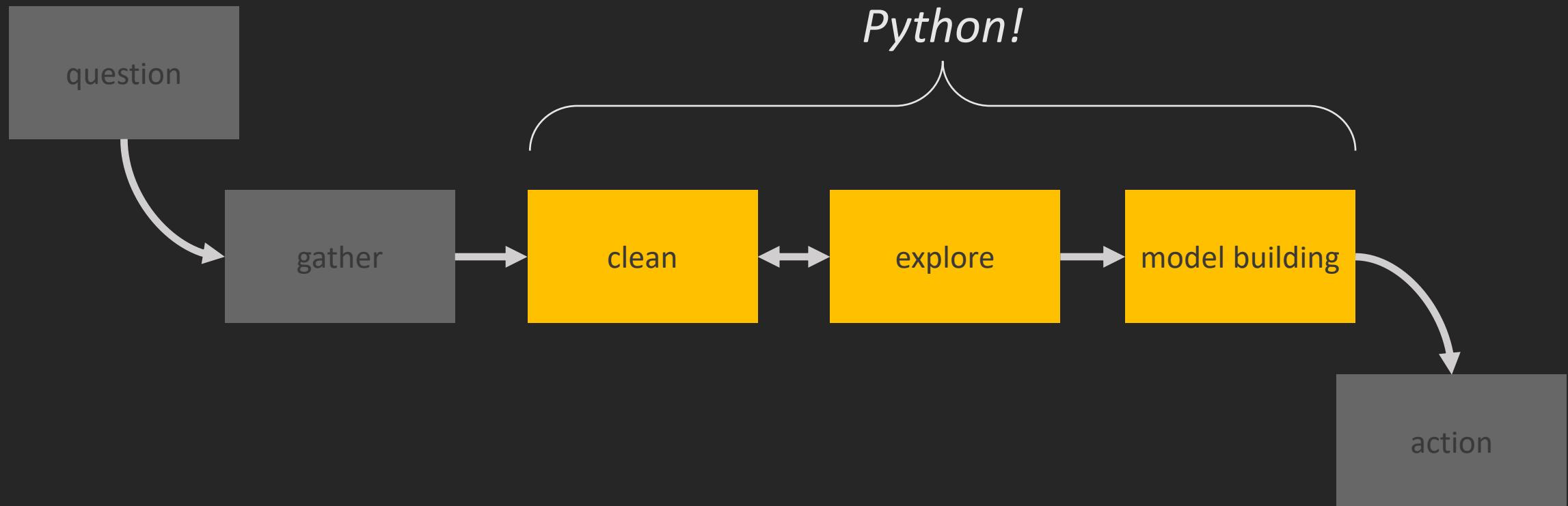


# Gather Data: The *Iris* dataset

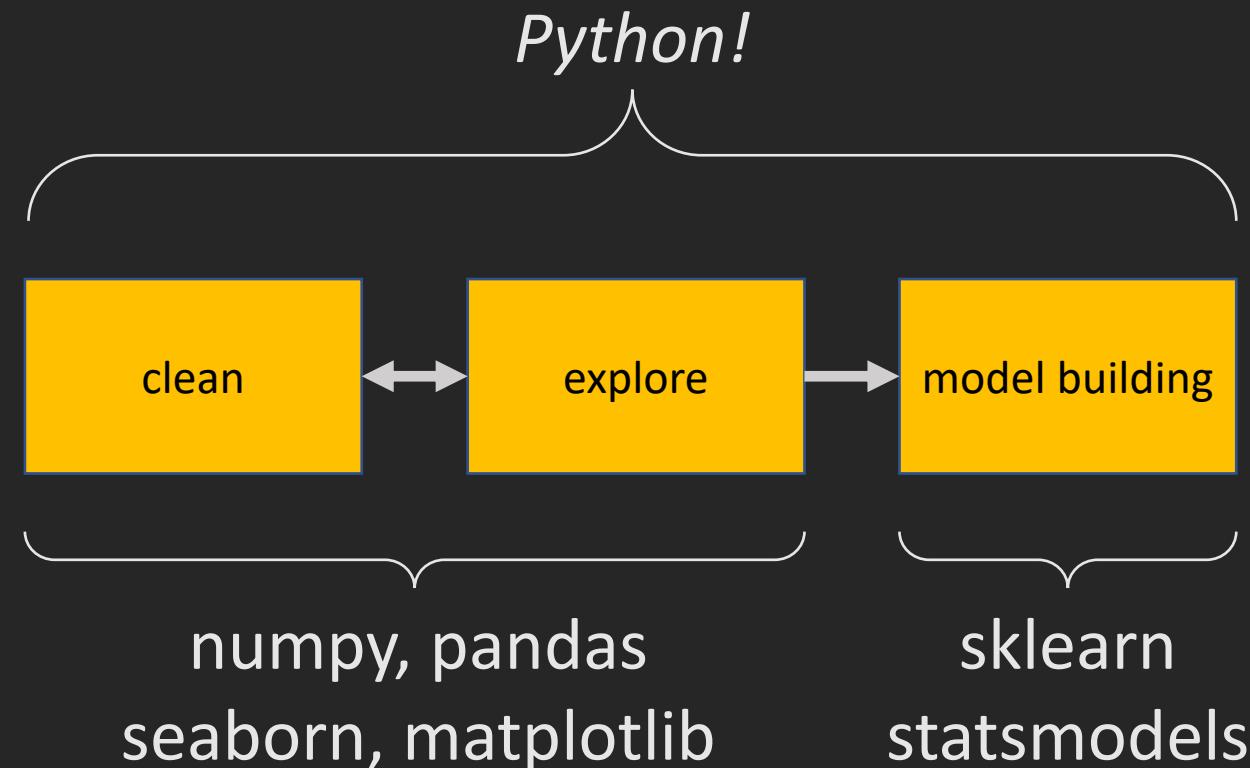
- Morphological data on three *Iris* species
- Collected by Edgar Anderson; introduced by RA Fisher in 1936
- Extensively used in statistics and machine learning training
- Distributed with sklearn and other libraries



# Data science workflow



# Python's Data Science Stack



# Data Cleaning

# Clean data are easier to analyze

- Format data in a way Python can read
  - “.csv” or “.txt” files are easiest to work with
  - Some libraries, like pandas, can handle “.xls” or “.xlsx” files
  - Excel, Numbers, etc. can export data to “.csv” or “.txt”

# Organized data are easier to analyze

- Use **long** format instead of **wide** format when possible
- Long format data have one observation per row:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

# Organized data are easier to analyze

- Use **long** format instead of **wide** format when possible
- Wide format data have multiple observations per row:

species	sepal length (cm)			sepal width (cm)			petal length (cm)			petal width (cm)		
	setosa	versicolor	virginica	setosa	versicolor	virginica	setosa	versicolor	virginica	setosa	versicolor	virginica
0	5.1	NaN	NaN	3.5	NaN	NaN	1.4	NaN	NaN	0.2	NaN	NaN
1	4.9	NaN	NaN	3.0	NaN	NaN	1.4	NaN	NaN	0.2	NaN	NaN
2	4.7	NaN	NaN	3.2	NaN	NaN	1.3	NaN	NaN	0.2	NaN	NaN
3	4.6	NaN	NaN	3.1	NaN	NaN	1.5	NaN	NaN	0.2	NaN	NaN
4	5.0	NaN	NaN	3.6	NaN	NaN	1.4	NaN	NaN	0.2	NaN	NaN

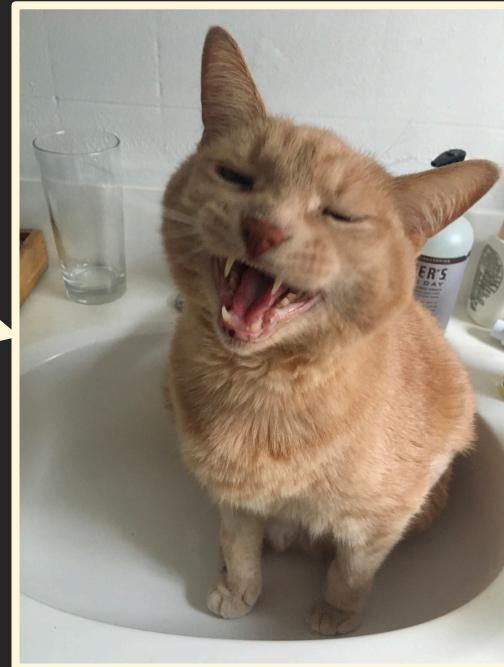
# Organized data are easier to analyze

- Use **long** format instead of **wide** format when possible
- Python can convert your data between these two formats:
  - Pandas “**melt**” converts wide -> long
  - Pandas “**pivot**” or “**pivot\_table**” convert long -> wide

# Cat behavior data (conceptual example)

This is Twinky. She likes

- Sitting in the bathroom sink
- Meowing loudly



This is Columbo. He likes

- Snuggling
- Chasing toys

# Cat behavior data, **wide** format

Name	Date	Time	Location_1	Date	Time	Location_2
Twinky	1/1/19	1:00p	sunny spot in chair	1/1/19	3:00p	food dish
Columbo	1/1/19	1:15p	cat bed	1/1/19	2:30p	litter box

# Cat behavior data, **long** format

Name	Date	Time	Location
Twinky	1/1/19	1:00p	sunny spot in chair
Twinky	1/1/19	3:00p	food dish
Columbo	1/1/19	1:15p	cat bed
Columbo	1/1/19	2:30p	litter box

# Data cleaning also involves:

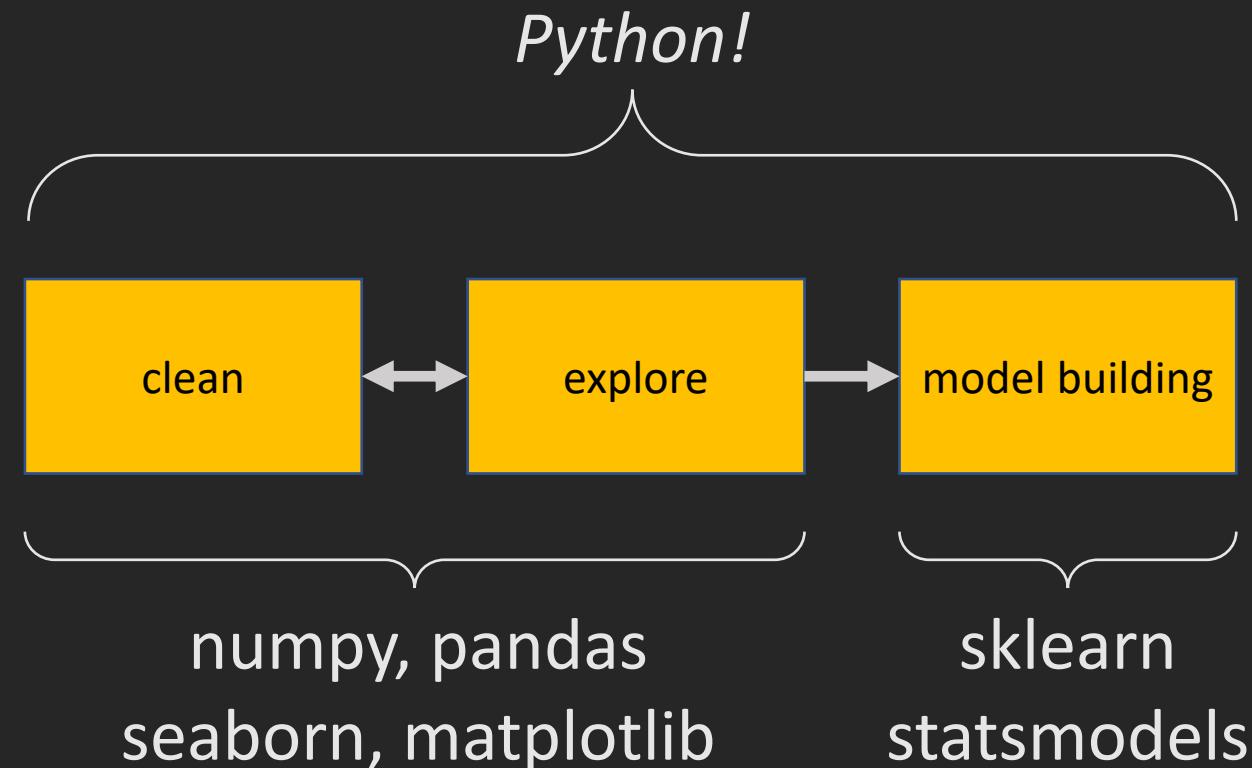
- Checking for **type consistency**
  - Numbers where you expect numbers
  - Characters where you expect characters
- Looking for **missing data**, and determining how to handle it
- **Removing contaminating** data (e.g. spaces or special characters) from your dataset

We'll explore some ways of doing this in Python in the hands-on exercise.

---

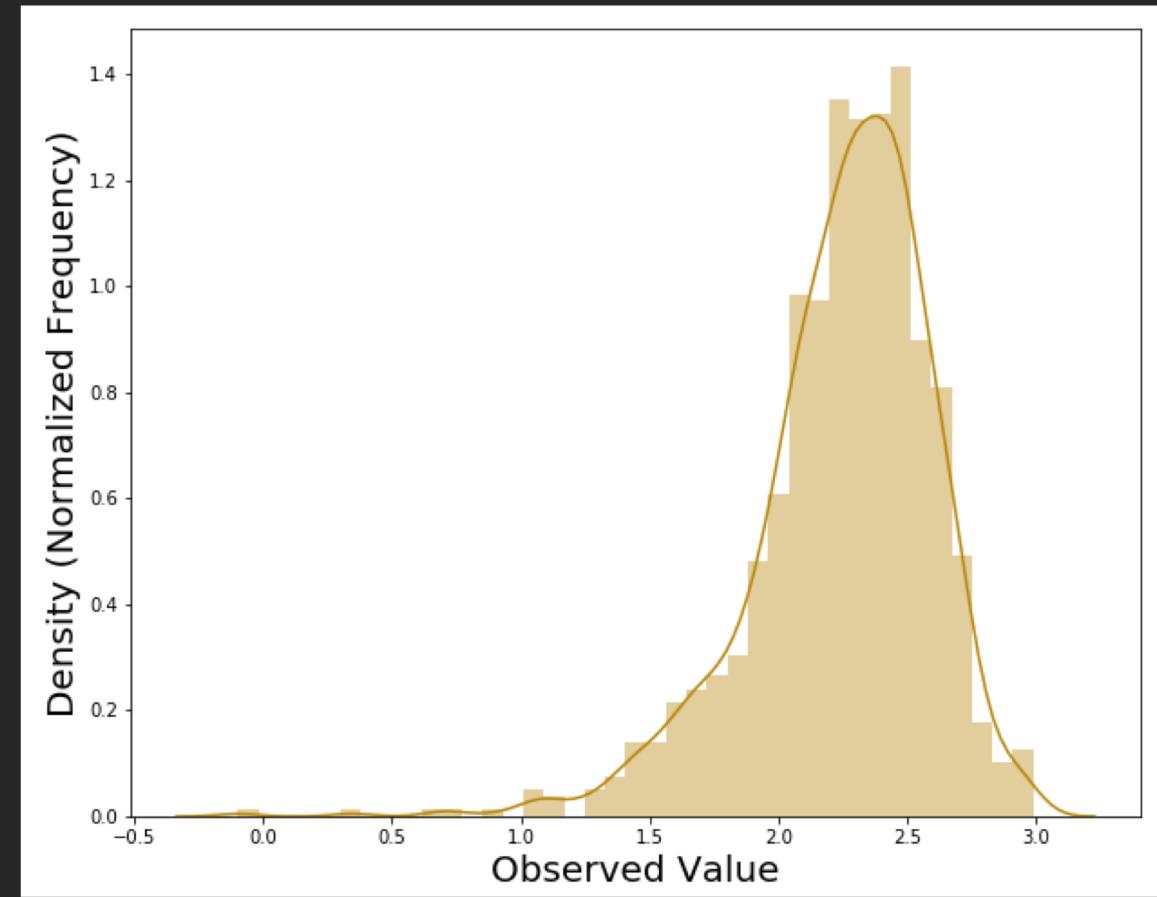
# Data Exploration

# Python's Data Science Stack



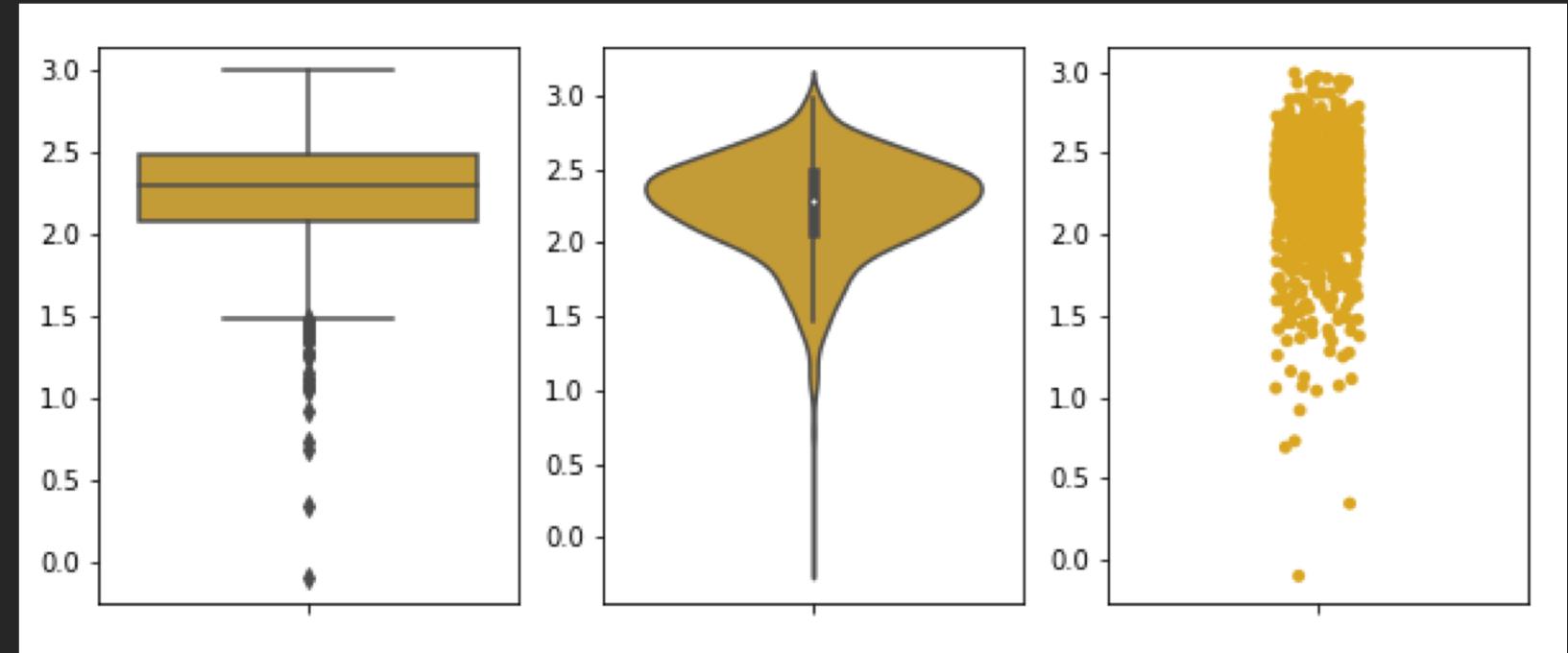
# Exploring Data through Visualization

**Histograms** show the frequency of observations of a single value



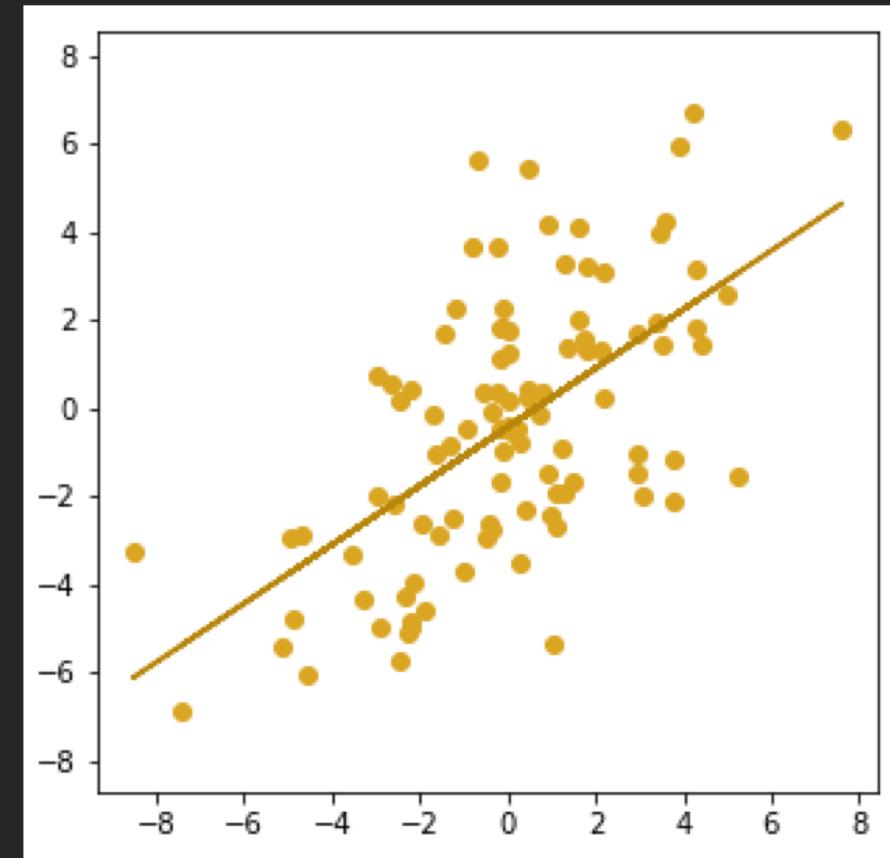
# Exploring Data through Visualization

**Boxplots,**  
**violin plots,**  
**and strip plots**  
also show data  
distributions



# Exploring Data through Visualization

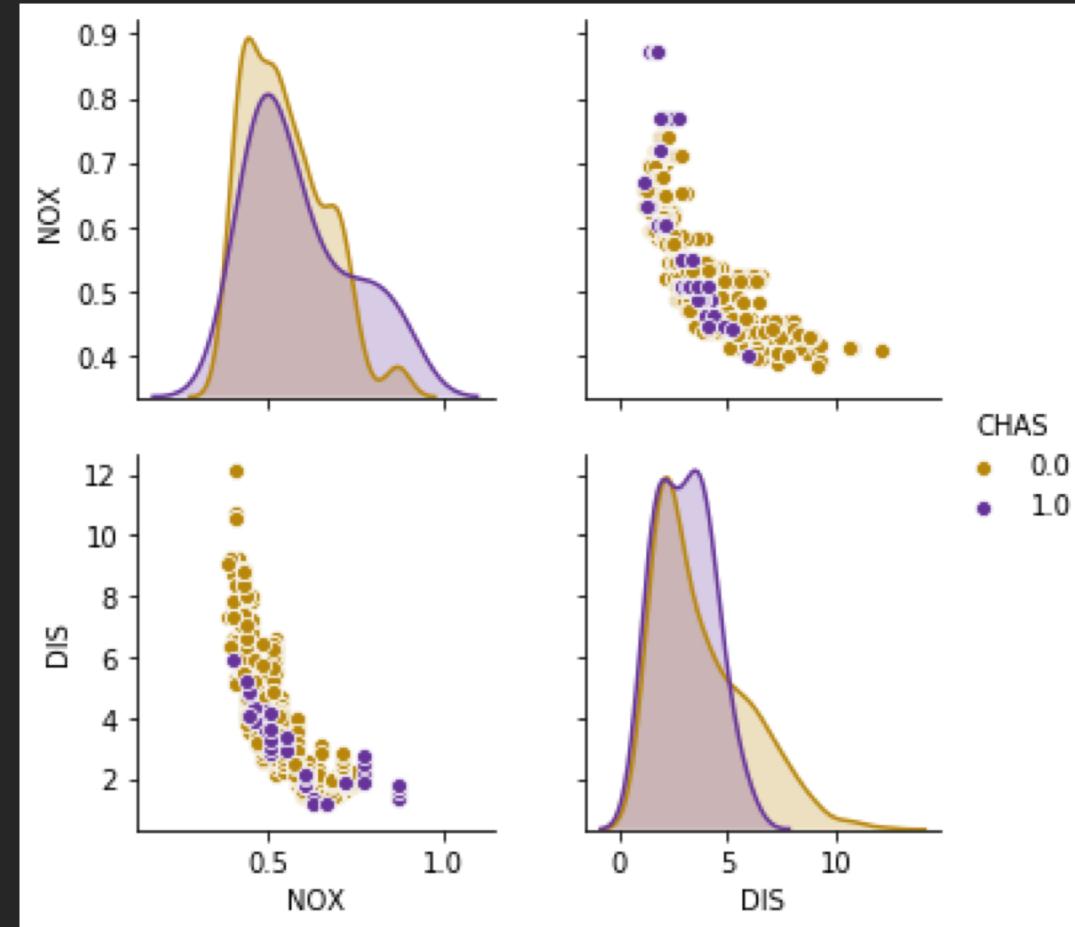
**Scatterplots** show the correlation between two variables



# Exploring Data through Visualization

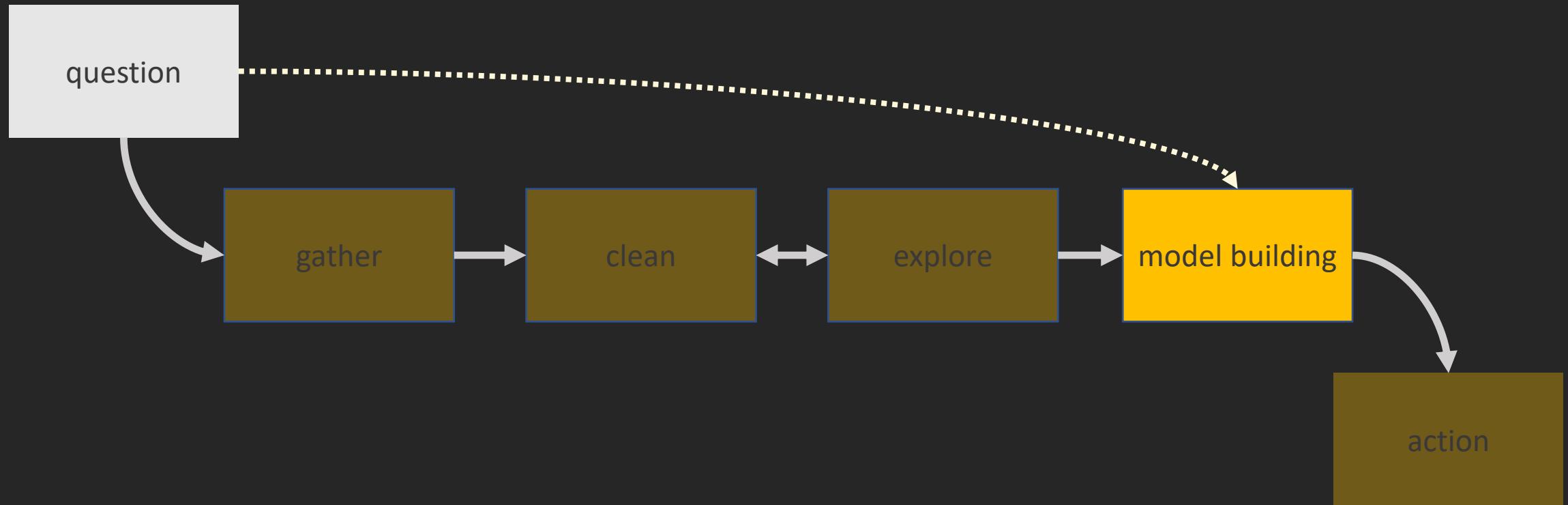
**Pair plots** make scatterplots for each pair of variables in a dataset, and histograms for each variable.

They can also be colored by categorical variables.



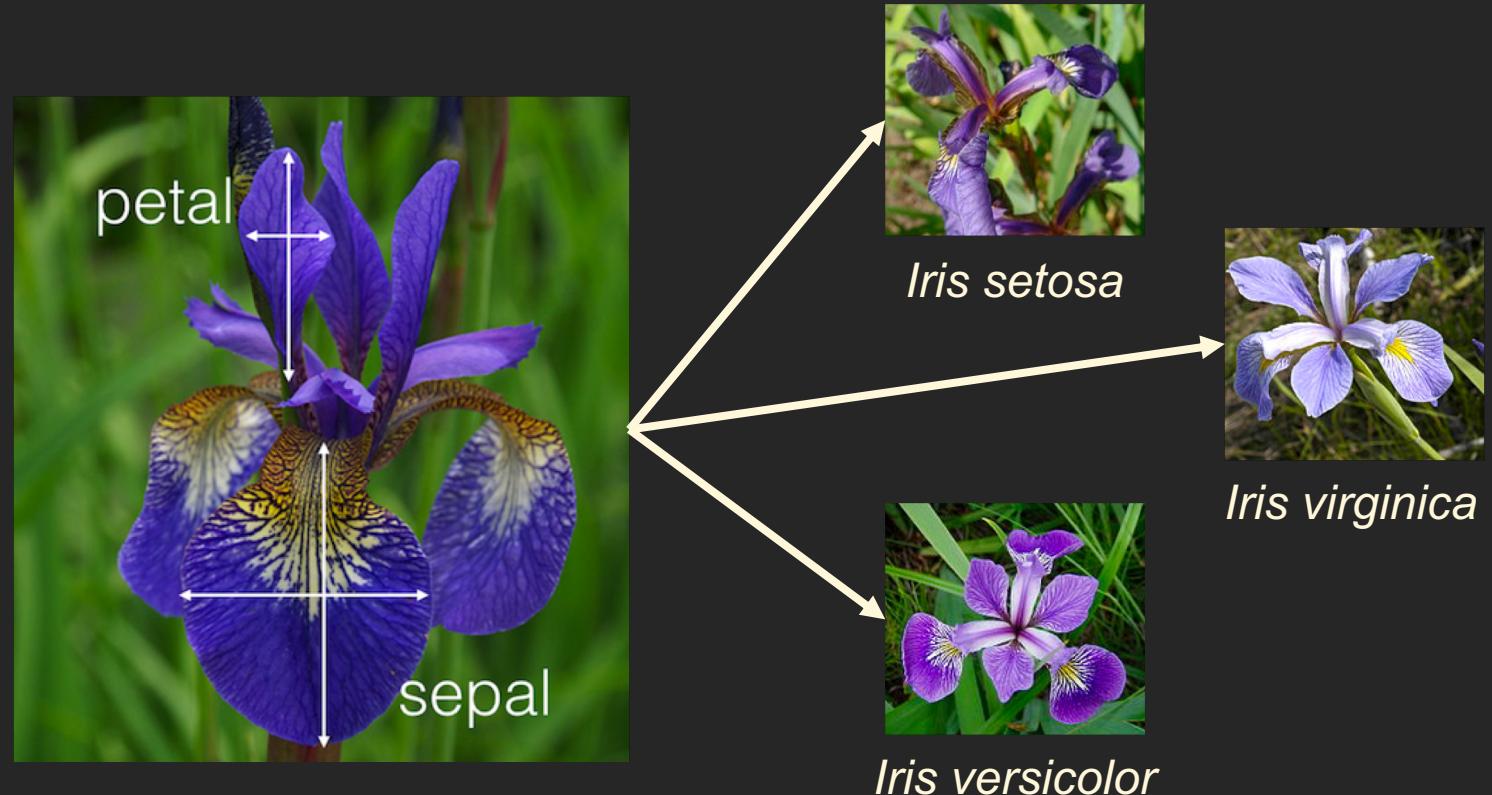
# Model Building

# Model Building



# Model Selection: Plant classification

Our example is a **supervised classification** problem: we want to group observations by a category (species), and we have labeled training data.



# Model Training and Testing

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Most of the data (70-80%) are used for model **training**.

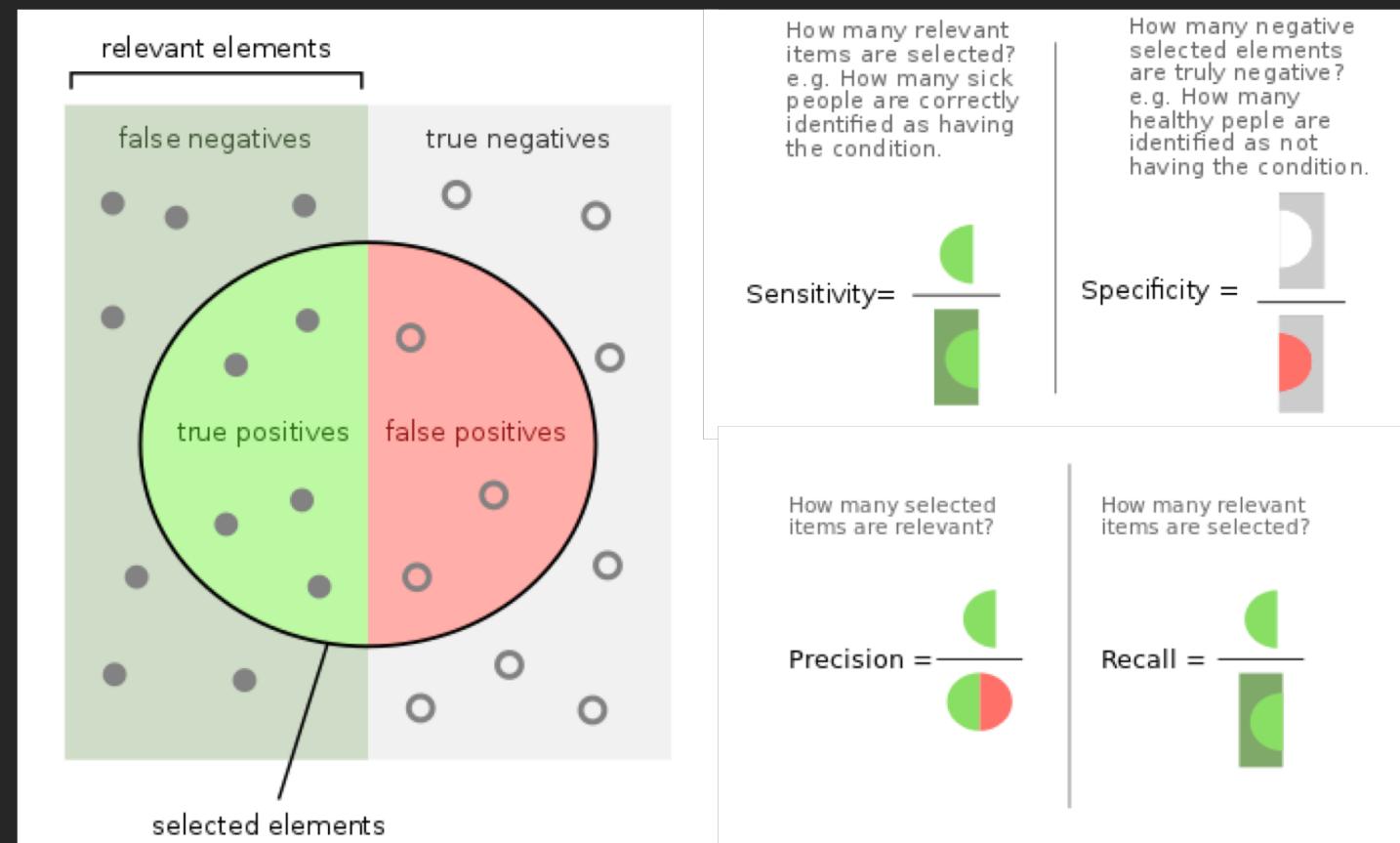
The remainder are set aside (not used), and reserved for **testing** the model.

# Model Training and Testing

Measure model prediction **accuracy** on the testing set

Classification Accuracy:

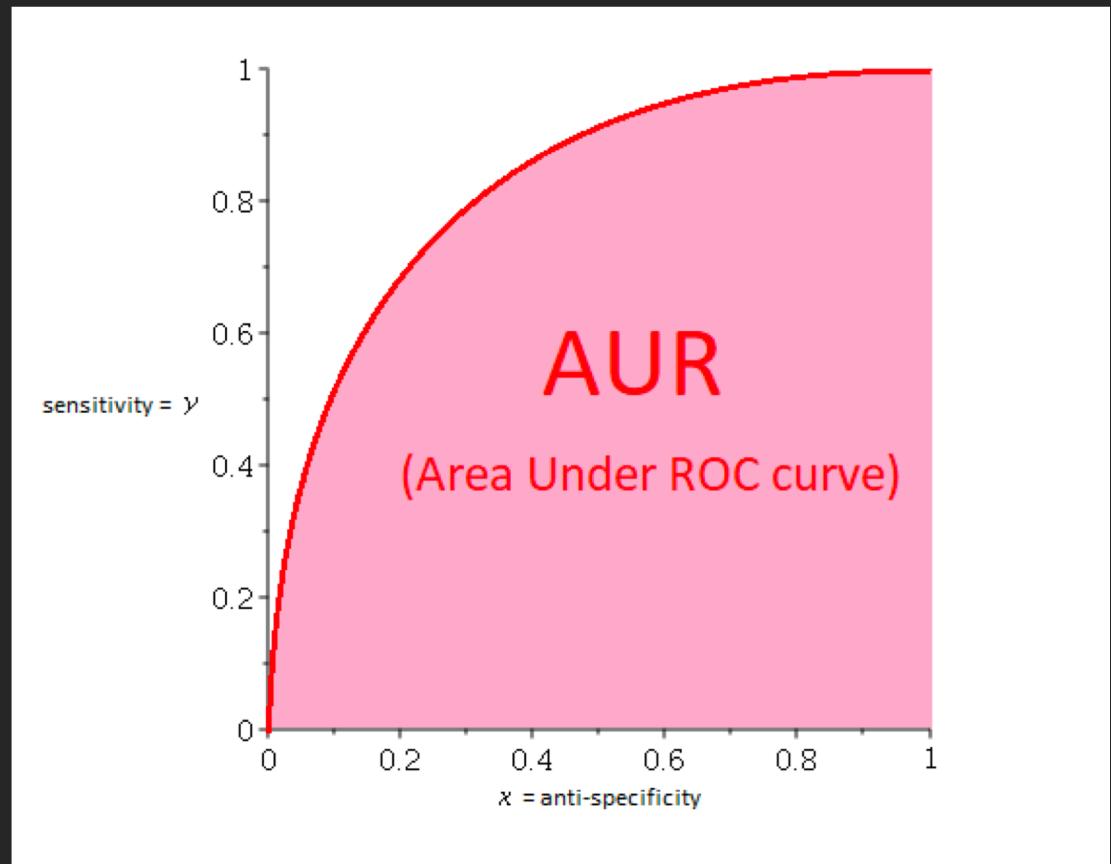
- **Precision**
- **Sensitivity (Recall)**
- **Specificity**



# Model Training and Testing

## Receiver Operating Characteristic

- As you increase your false positive rate, how quickly do you increase your true positive rate?
- Goal is Area Under Curve = 1



# Model Training and Testing

Measure model prediction **accuracy** on the testing set

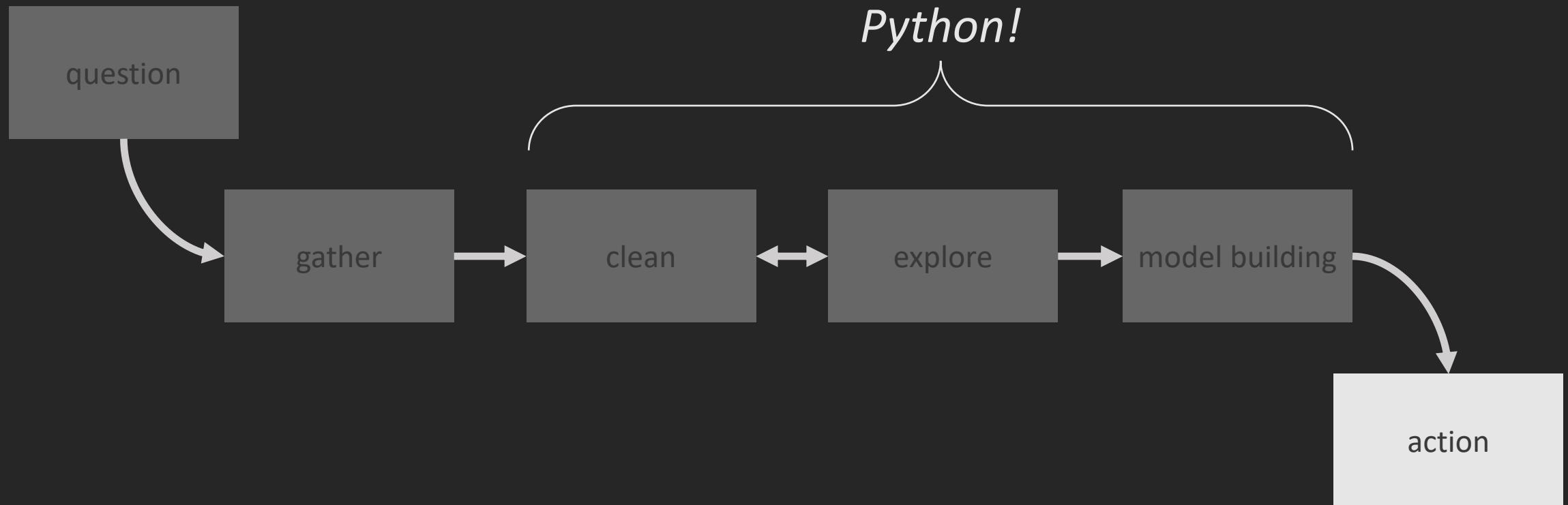
Regression Accuracy (brief)

- **(Normalized) Root mean squared error**
- **R<sup>2</sup>**

# Model Training and Testing

- Other train/test strategies
  - K-fold cross-validation
  - Leave-one-out validation

# Data science workflow



# Actionability of Data

- Plant classifier could be used to help gardeners, scientists, or anyone with an interest in irises.
- Did we fairly and equitably represent all *Iris* species?
- Does our model introduce bias into classification? If so, what consequences might there be?

Our *Iris* example is a bit contrived, but remember that **machine learning and data science drive actions daily** in a number of domains: healthcare, commerce, education, etc.

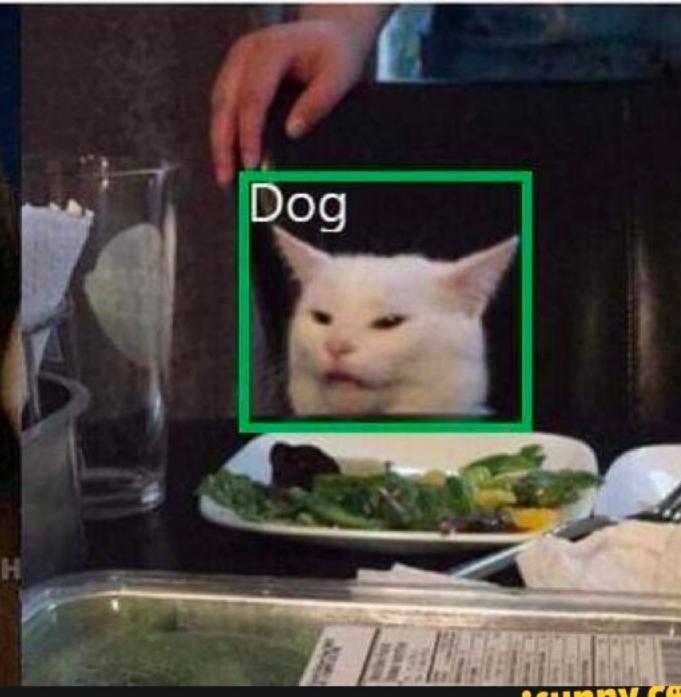
# Classification Error

# Classification Error

People with no idea  
about AI, telling me my  
AI will destroy the world



Me wondering why my  
neural network is  
classifying a cat as a dog..



# Classification Error

```
Probability% is <label>
0.3375208 % is towel
0.0611235 % is Aid
0.0420474 % is Pekingese, Peke
0.0291371 % is Angora rabbit
0.0224534 % is tissue, toilet paper, bathroom tissue
0.0200325 % is dough
0.0150566 % is handbasin, washbowl, lavabo, wash-hand basin
0.0145770 % is powder
0.0145677 % is welcome mat
0.0142751 % is nematode worm, roundworm
```



```
Probability% is <label>
0.5441126 % is cat
0.1973237 % is tabby cat
0.1003607 % is cat
0.0492031 % is cat, Siamese
0.0410424 % is Highland white terrier
0.0118241 % is heater
0.0117978 % is catamount
0.0083975 % is cairn terrier
0.0074549 % is Pekingese, Peke
0.0067524 % is cat
```



# Classification Error



# Hands-On Exercises

- Clone or download the GitHub repo:  
[https://github.com/kellyp2738/python\\_data\\_science](https://github.com/kellyp2738/python_data_science)
- How to use a Jupyter Notebook

Press the “Run” button to run code cell

