HPE DSI 311
Introduction to Machine Learning

Summer 2021

Instructor: Ioannis Konstantinidis

UNIVERSITY of HOUSTON
DIVISION OF RESEARCH
HEWLETT PACKARD ENTERPRISE DATA SCIENCE INSTITUTE

Model evaluation

- Cross-validation (k-fold)
- Metrics and Scoring

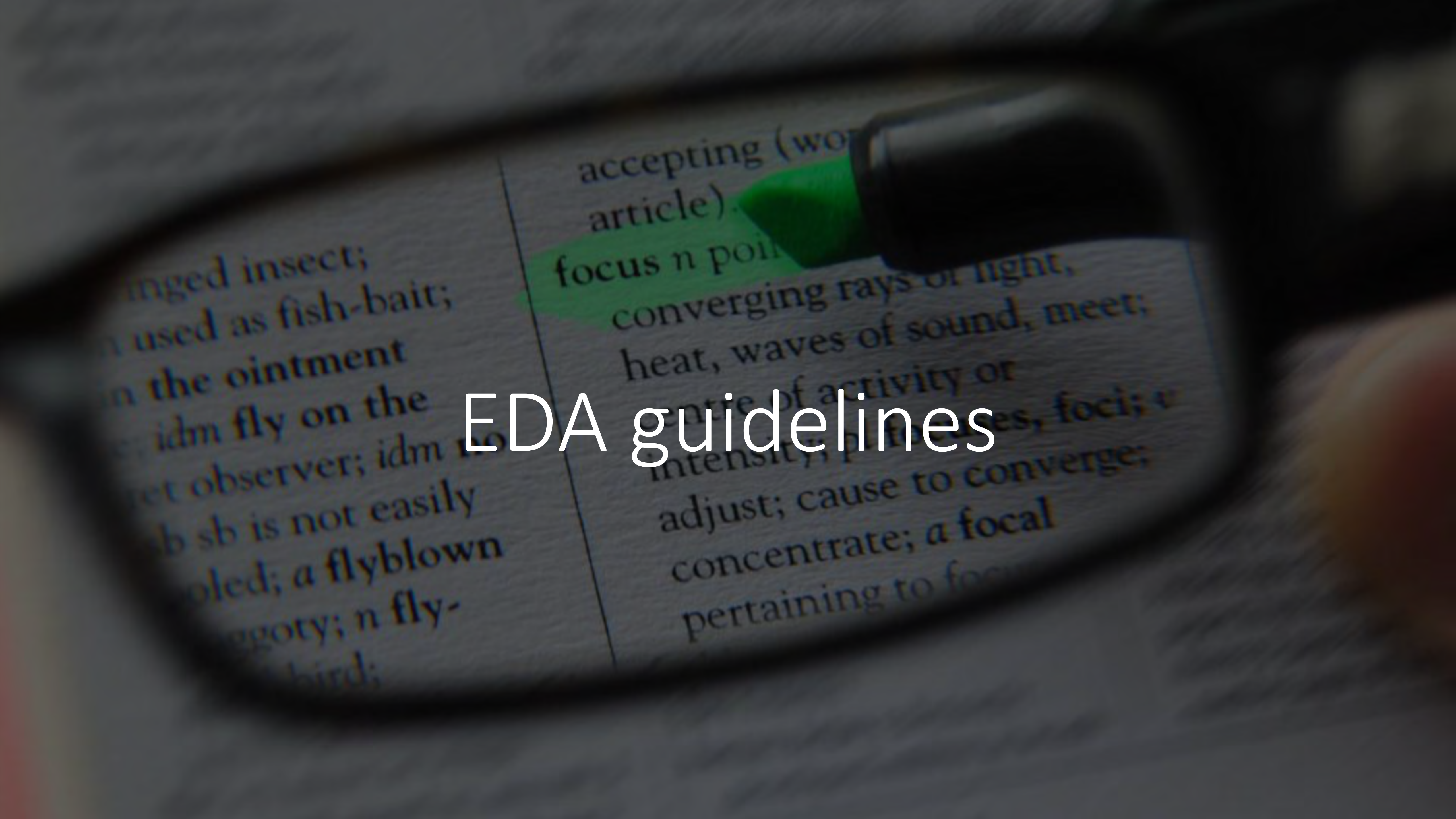
Supervised models for classification

- k-Nearest Neighbors
- Logistic regression
- Support Vector Machines
- Decision Trees
- Random Forests
- Gradient Boosting

Pit stop: organizational guidelines

- Exploratory data analysis (EDA)
  - Numerical summaries
  - Graphical summaries
- Data pre-processing
  - Data cleaning and tidying up
- Data transformations
  - Scaling (standard/MinMax)
  - Feature Extraction (PCA / dimension reduction)

EDA guidelines



EDA checklist

Sanity checks:

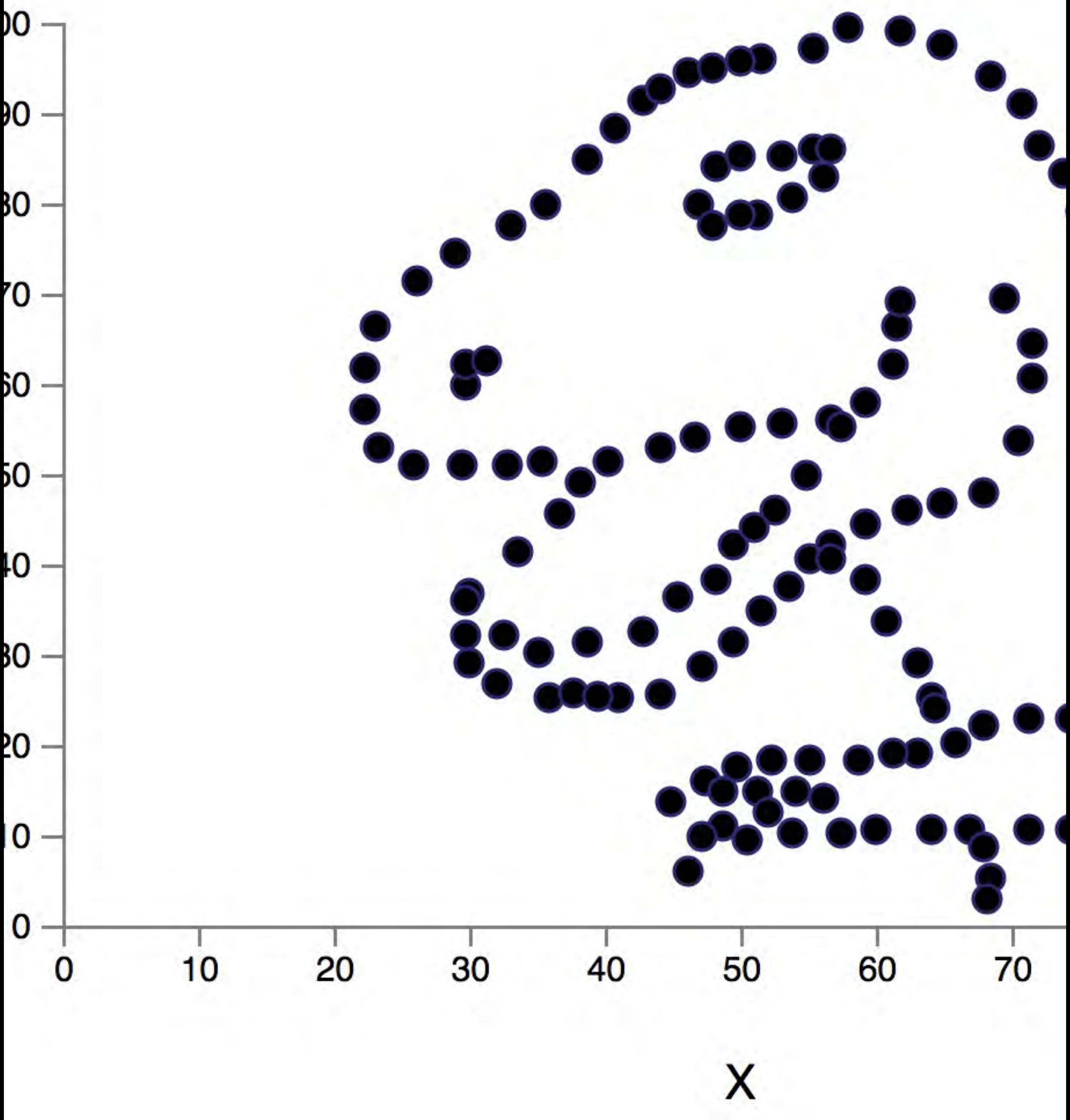
- Read in your data
- Check the packaging
- Look at the top and the bottom of your data
- Check your statistics
    - Numerical Summaries
- Check your plots
    - Graphical Summaries

Reasons for making plots

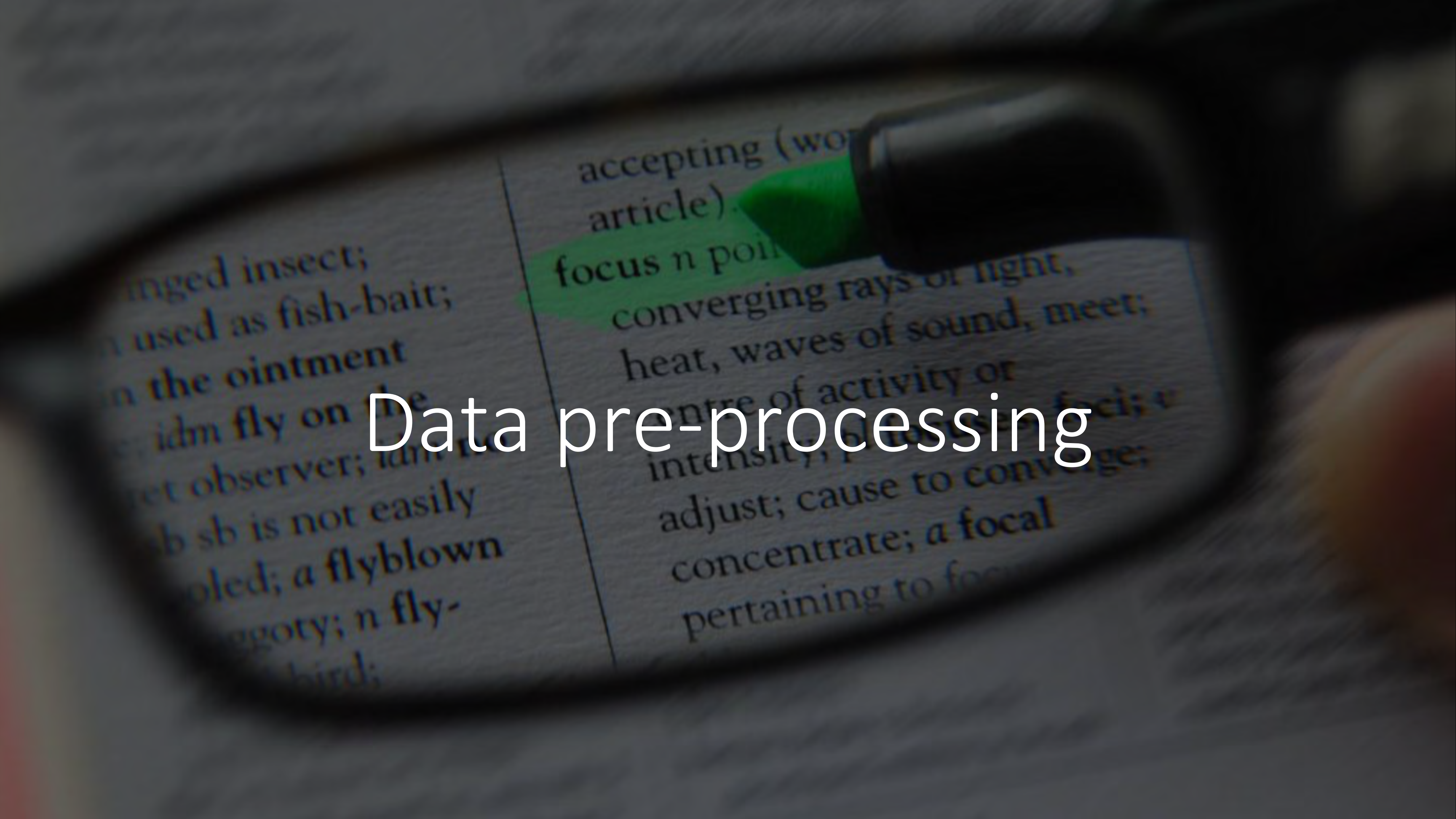
- Setting expectations for what the data should look like.
- Checking deviations from what you might expect
- Numerical summaries don’t give the whole picture

Datasaurus

https://www.autodesk.com/research/publications/same-stats-different-graphs



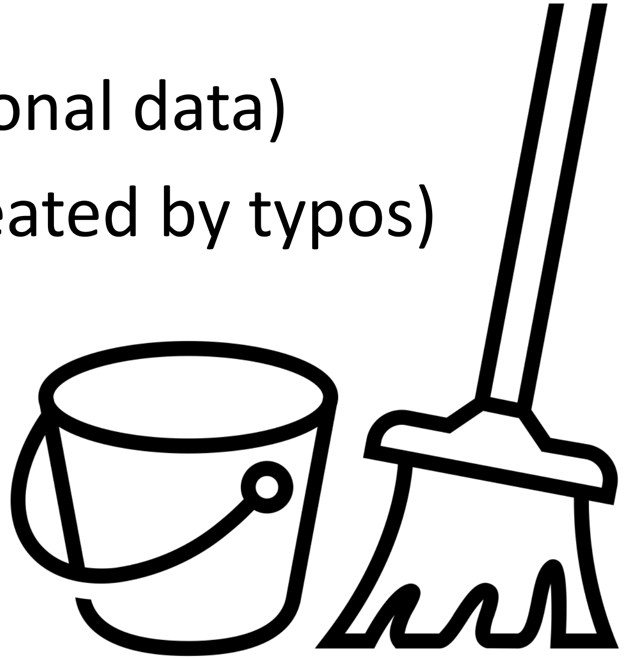
Data pre-processing



Cleaning data

No incomplete, incorrect, inaccurate, or irrelevant parts

- identifying missing values
- parsing dates and numbers
- correcting character encodings (for international data)
- matching similar but not identical values (created by typos)
- filling in structural missing values
- …



Clean ≠ Tidy

Hadley Wickham

https://www.jstatsoft.org/v59/i10/

In tidy data:

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

- Cf. Codd's 3rd normal form

Variable #1 Variable #2 Variable #3 Variable #4

Observation #1

Observation #2

Observation #3

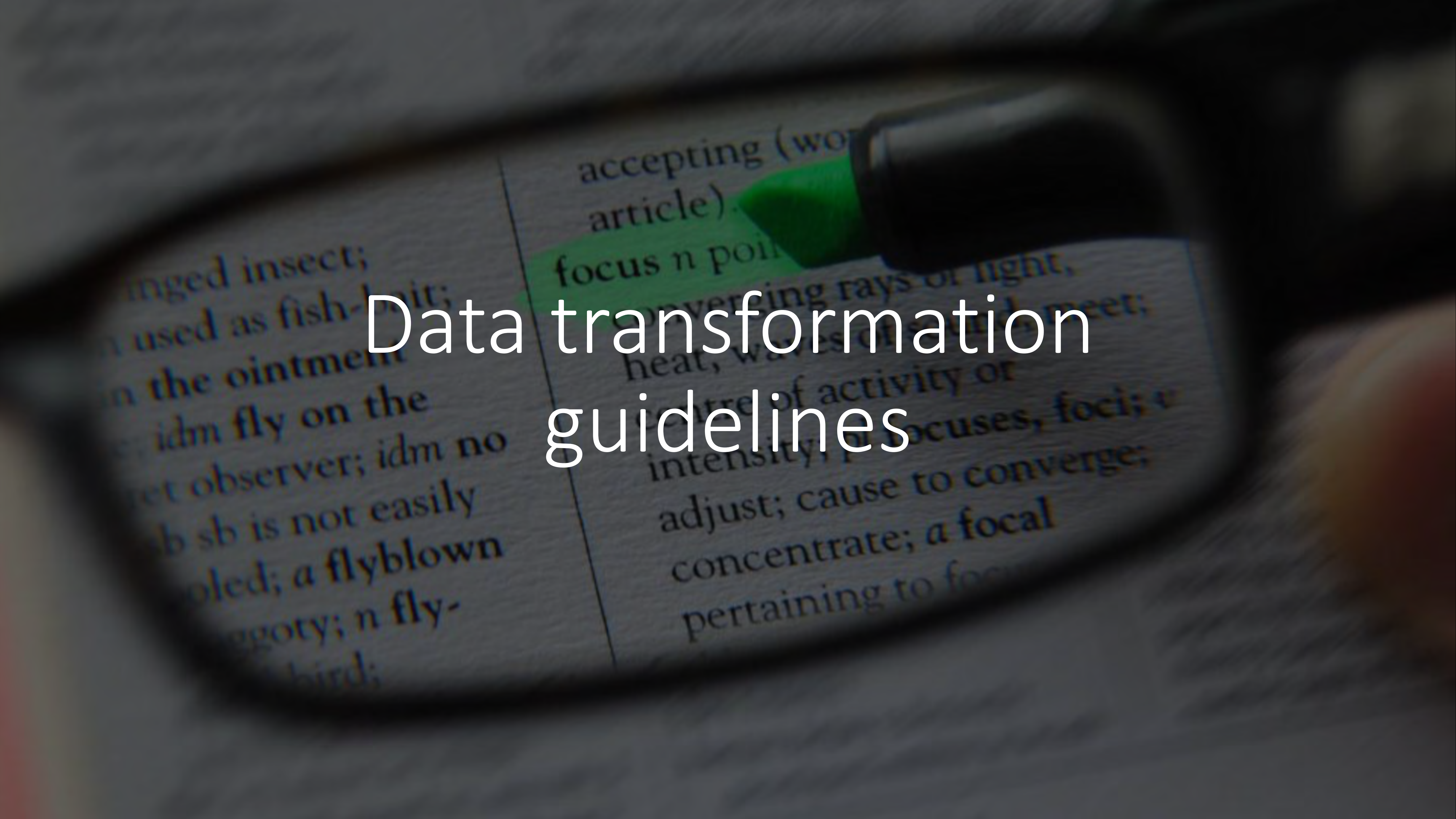
Observation #4

Observation #5

Messy data: common problems

- Column headers are values, not variable names.
- Multiple variables are stored in one column.
- Variables are stored in both rows and columns.
- Multiple types of observational units are stored in the same table.
- A single observational unit is stored in multiple tables.

Data transformation guidelines



Data transformations are data processing tasks

- They come after pre-processing the data
- They come after EDA

EDA and data pre-processing tasks are done to make the learning process possible

Features organization transformations are done to improve the learning process

Why mess with the data values?

Some objective functions rely on distance

- Algorithms like KNN and SVM (and neural nets) are using distances between data points to determine their similarity.

- Tree-based algorithms on the other hand (e.g., decision trees, random forests) do not rely on similarity.

Distance measurements are affected by scaling

Temp and humidity:

- F value range is about 0 - 100
- % value range is about 0 - 100

A change of one unit in temperature value

counts the same as

a change of one unit in humidity values

Distance measurements are affected by scaling

Temp and humidity:

- F value range is about 0 - 100
- % value range is about 0 - 100

A change of one unit in temperature value

counts the same as

a change of one unit in humidity values

But this is an accident due to choice of units

Distance measurements are affected by scaling

Temp and humidity:

- C   value range is about 0 - 40
- decimal   value range is about 0 - 1

A change of one unit in temperature value

counts as a LOT less than

a change of one unit in humidity values

Distance measurements are affected by scaling

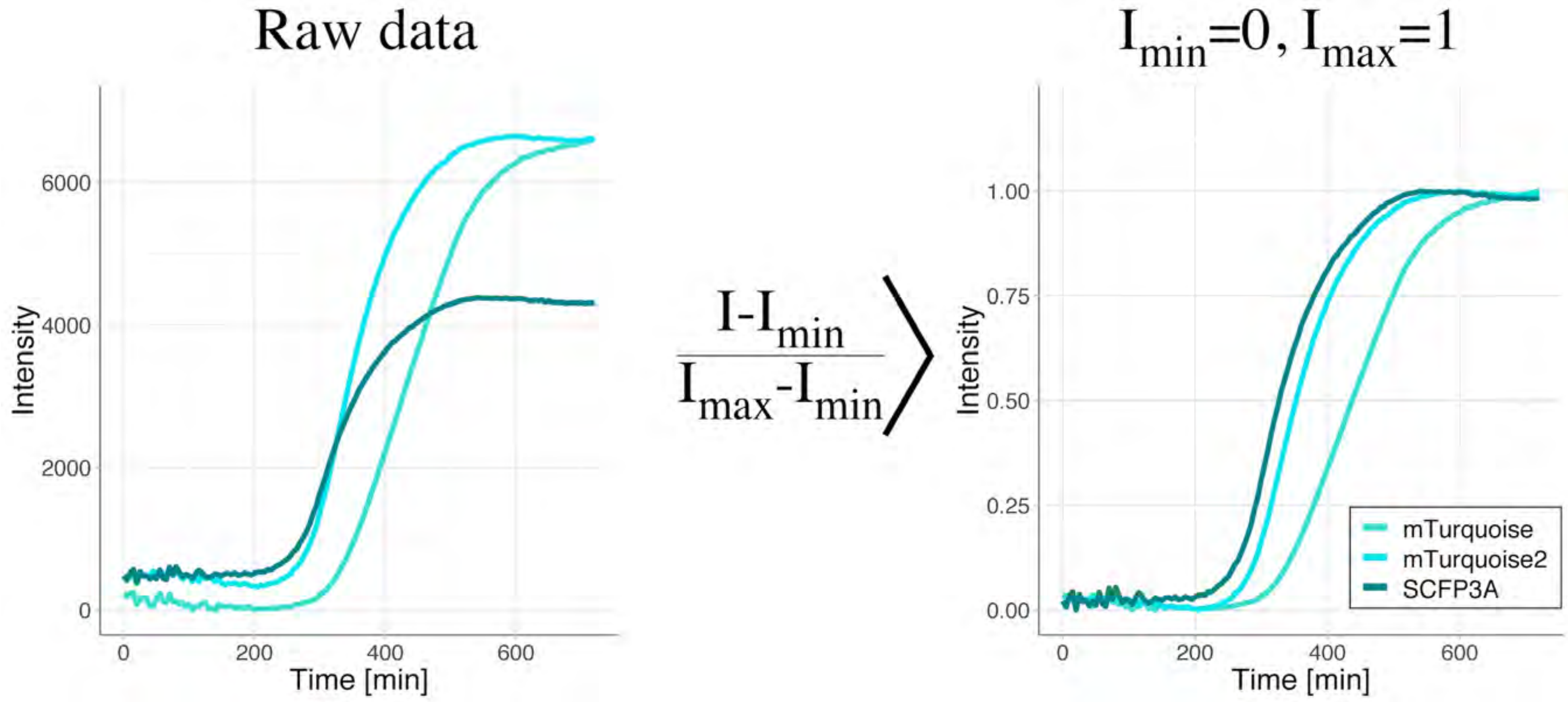
The relative influence of a variable on the total similarity/distance should not depend on an arbitrary choice of units

-> Need to scale the data

How to scale the data?

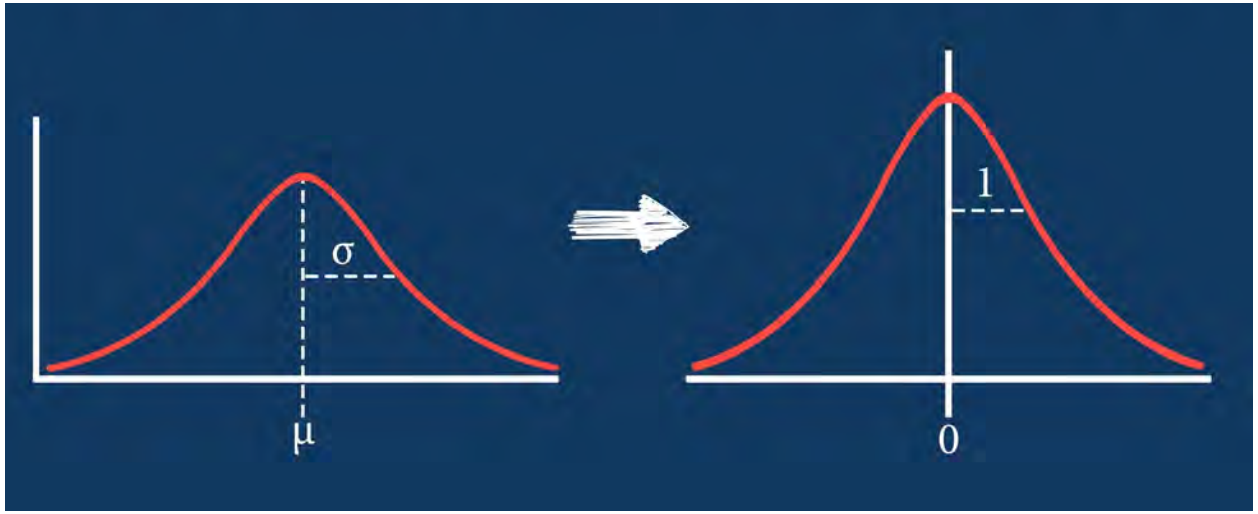
MinMaxScaler: uniform range of 0 - 1

$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}$$



StandardScaler: centered at 0

$$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation }(x)}$$



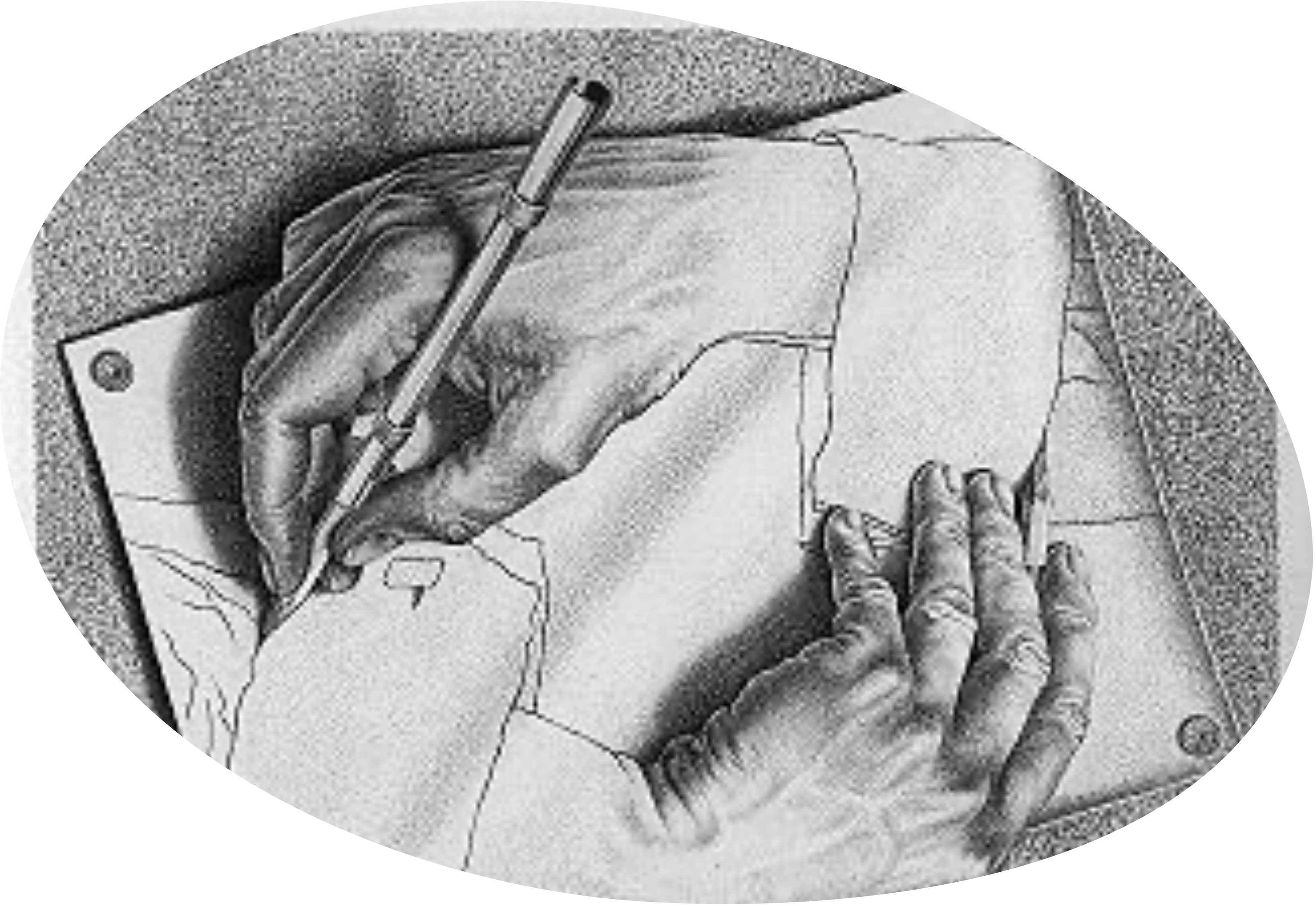
Are the data normally distributed?

Fit once, and then reuse

- **Fit the scaler using available training data**. For normalization, this means the training data will be used to estimate the minimum and maximum observable values. This is done by calling the *fit()* function.

- **Apply the scale to training data**. This means you can use the normalized data to train your model. This is done by calling the *transform()* function.

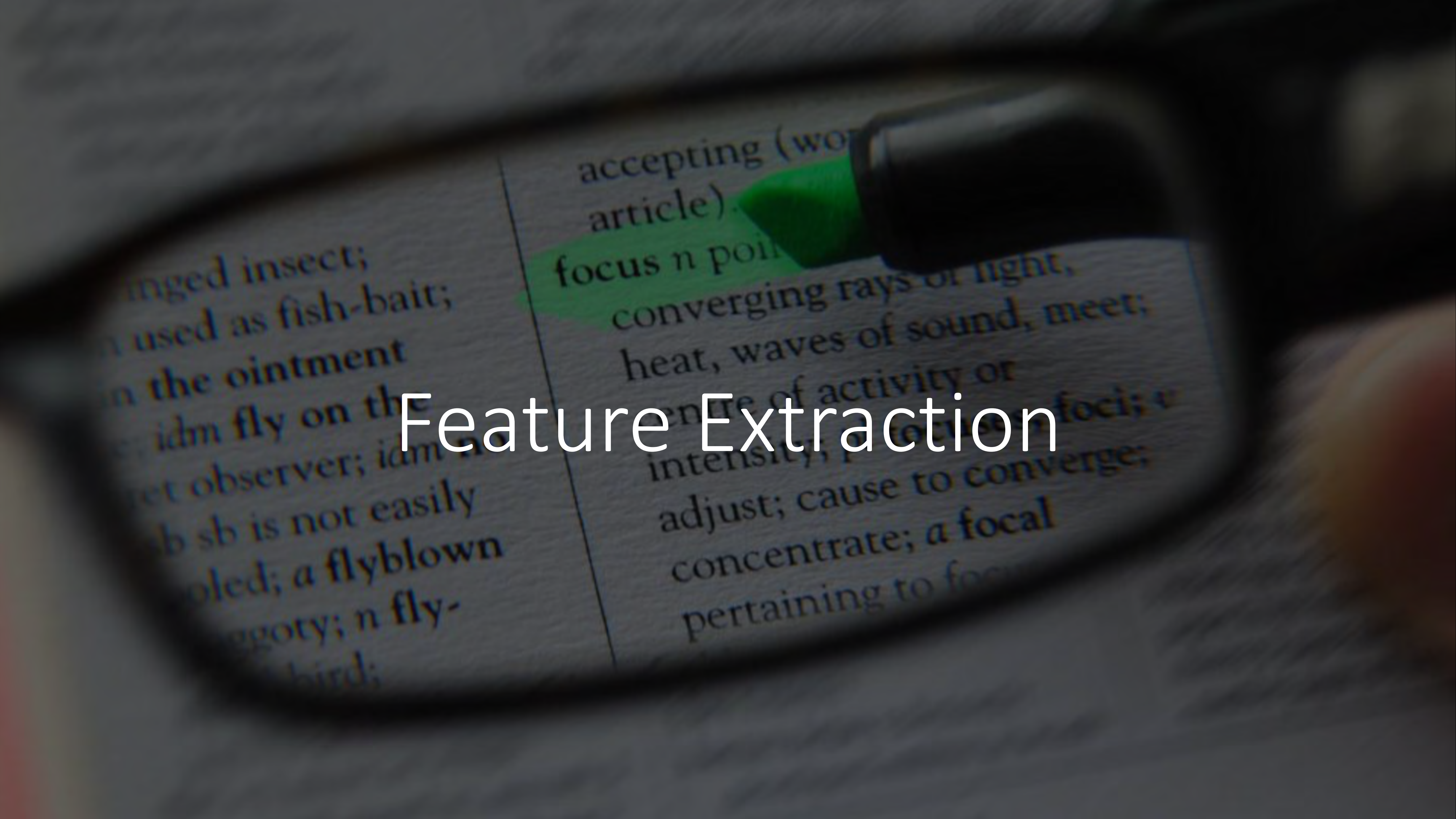
- **Apply the scale to data going forward**. This means you can prepare new data in the future on which you want to make predictions.



Hands-on Example:

Scaling

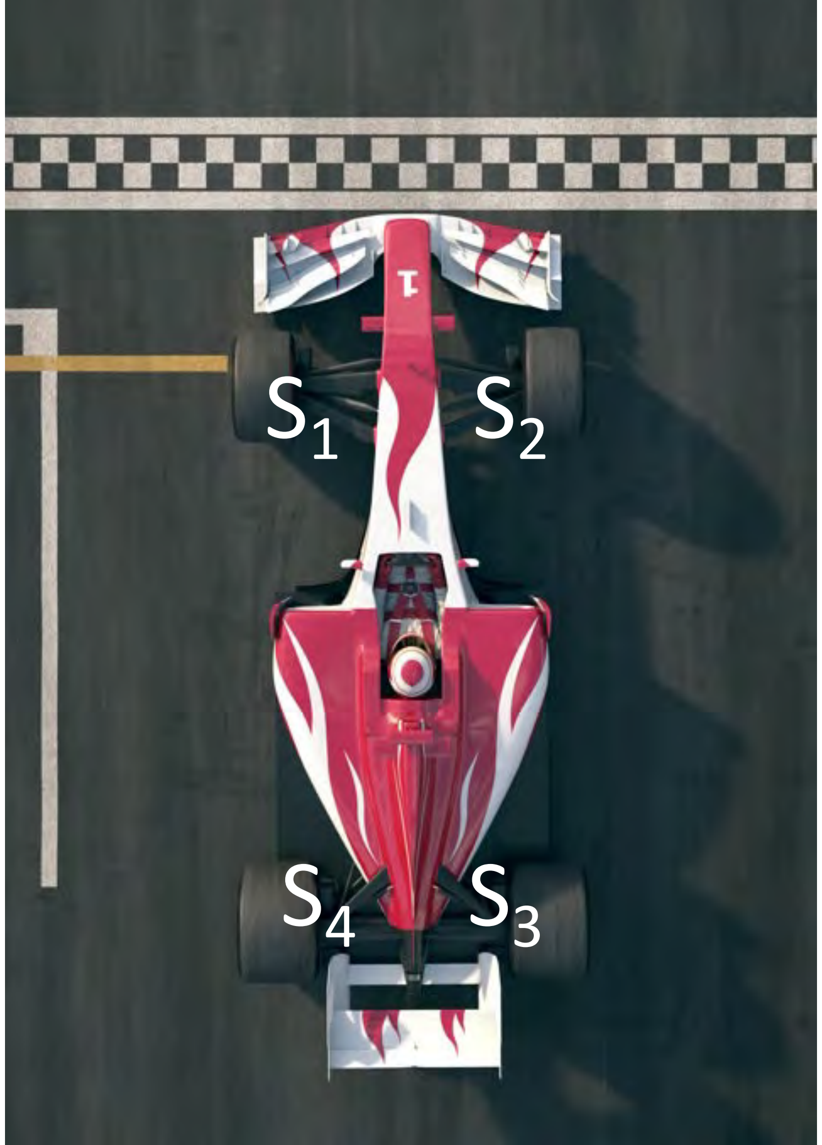
Feature Extraction



Why mess with the variables or columns?

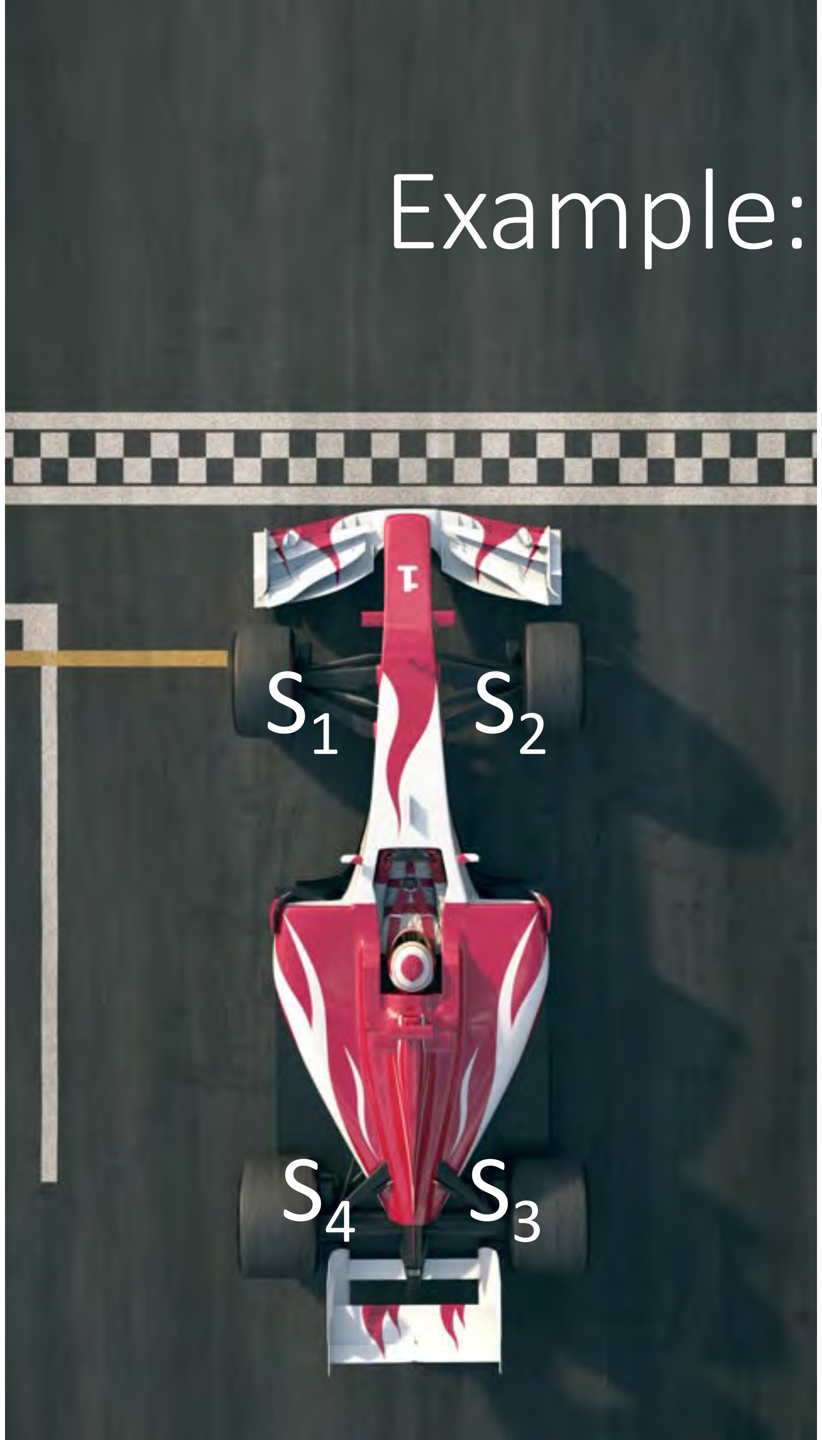
Example: Composite Measures

Example: Original variables



Four sensors measuring rotation speed (spin) at each wheel: S_1, S_2, S_3, S_4

Example: Features



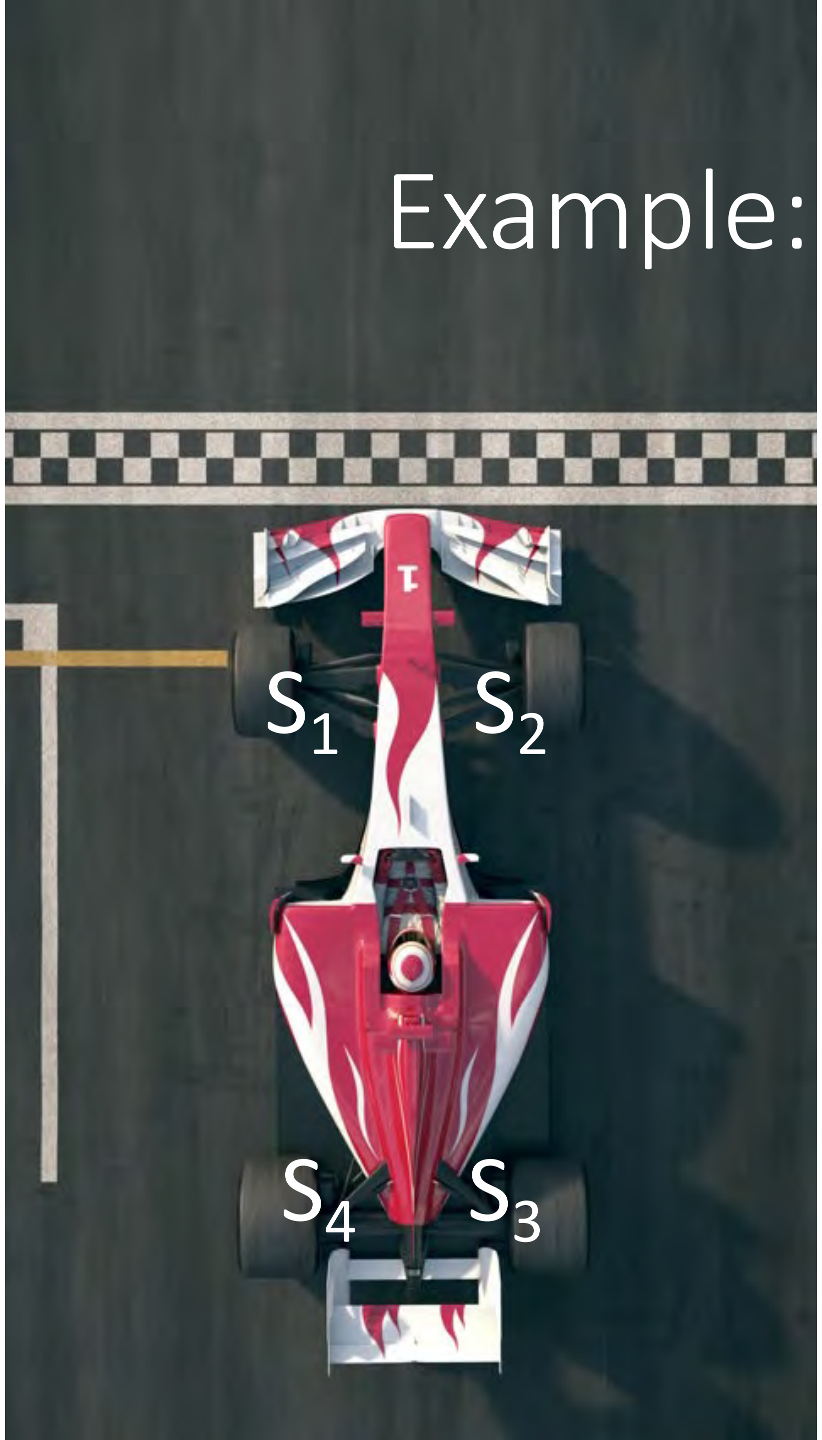
Four sensors measuring rotation speed (spin) at each wheel: S_1, S_2, S_3, S_4

New composite measure (feature):

$T_1 = (S_1 + S_2 + S_3 + S_4) / 4 = \frac{1}{4}S_1 + \frac{1}{4}S_2 + \frac{1}{4}S_3 + \frac{1}{4}S_4$

This is a more reliable indicator of car speed

Example: Features



Four sensors measuring rotation speed (spin) at each wheel: S_1, S_2, S_3, S_4

New composite measure (feature):

$T_1 = (S_1 + S_2 + S_3 + S_4) / 4 = \frac{1}{4}S_1 + \frac{1}{4} S_2 + \frac{1}{4} S_3 + \frac{1}{4} S_4$

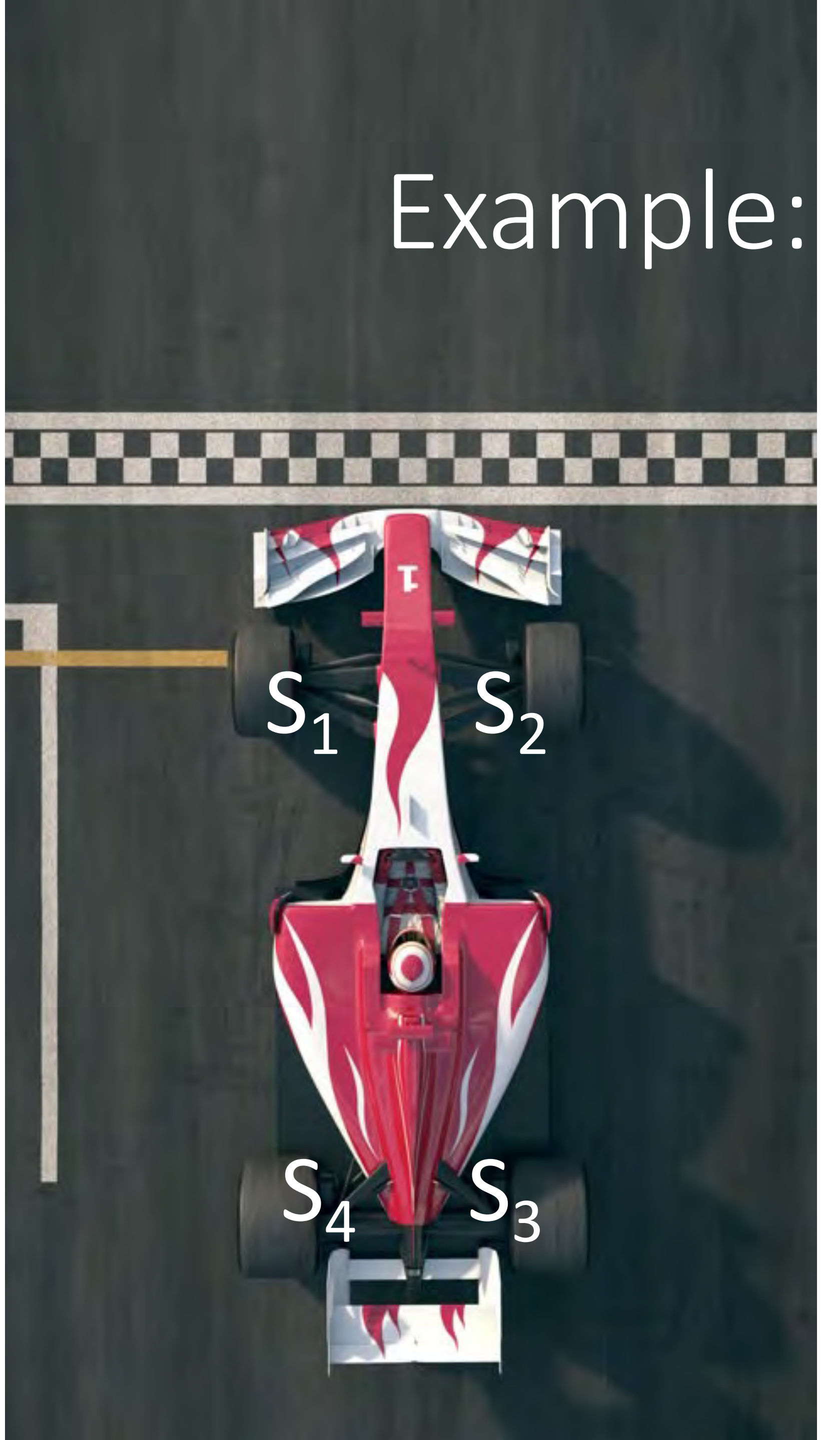
This is a more reliable indicator of car speed

New composite measure (feature):

$T_2 = 0.5 \{ (\frac{S_1 + S_3 + S_4}{3}) - S_2\} = \frac{1}{6}S_1 - \frac{1}{2} S_2 + \frac{1}{6} S_3 + \frac{1}{6} S_4$

If this starts to veer away from zero, then tire #2 is spinning faster than the others (possible flat)

Example: Features



Four sensors measuring rotation speed (spin) at each wheel: S_1, S_2, S_3, S_4

New composite measure (feature):

$T_1 = (S_1 + S_2 + S_3 + S_4) / 4 = \frac{1}{4}S_1 + \frac{1}{4}S_2 + \frac{1}{4}S_3 + \frac{1}{4}S_4$

This is a more reliable indicator of car speed

New composite measure (feature):

$T_2 = 0.5 \{ (\frac{S_1 + S_3 + S_4}{3}) - S_2\} = \frac{1}{6}S_1 - \frac{1}{2}S_2 + \frac{1}{6}S_3 + \frac{1}{6}S_4$

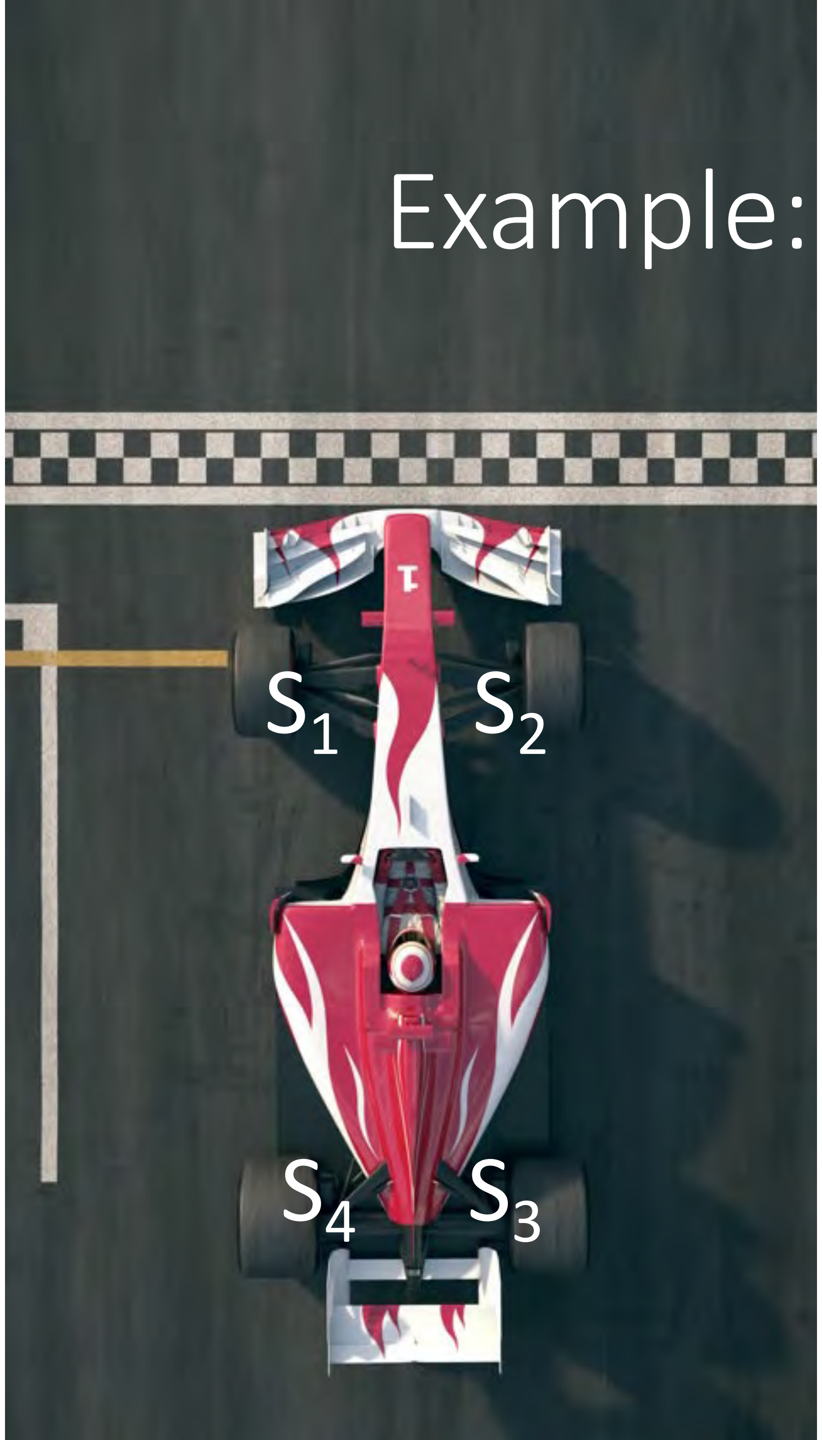
If this starts to veer away from zero, then tire #2 is spinning faster than the others (possible flat)

Similarly,

$T_3 = 0.5 \{ (\frac{S_1 + S_2 + S_4}{3}) - S_3 \}$

$T_4 = 0.5 \{ (\frac{S_1 + S_2 + S_3}{3}) - S_4 \}$

Example: Features



Original measures (variables):

S_1, S_2, S_3, S_4

New composite measures (features):

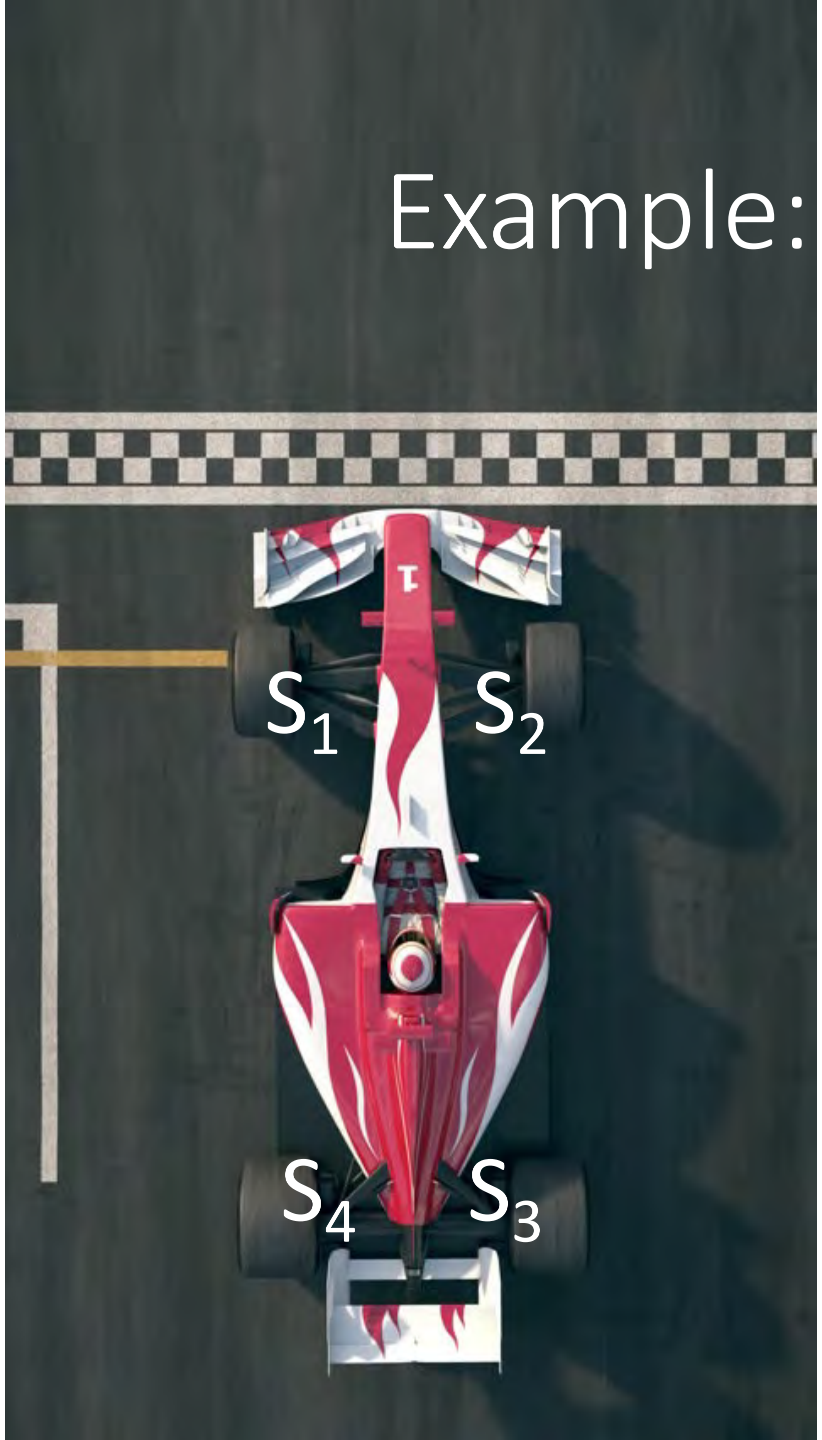
$$T_1 = \frac{1}{4}S_1 + \frac{1}{4}S_2 + \frac{1}{4}S_3 + \frac{1}{4}S_4$$

$$T_2 = \frac{1}{6}S_1 - \frac{1}{2}S_2 + \frac{1}{6}S_3 + \frac{1}{6}S_4$$

$$T_3 = \frac{1}{6}S_1 + \frac{1}{6}S_2 - \frac{1}{2}S_3 + \frac{1}{6}S_4$$

$$T_4 = \frac{1}{6}S_1 + \frac{1}{6}S_2 + \frac{1}{6}S_3 - \frac{1}{2}S_4$$

Example: Features



Original measures (variables):
S_1, S_2, S_3, S_4

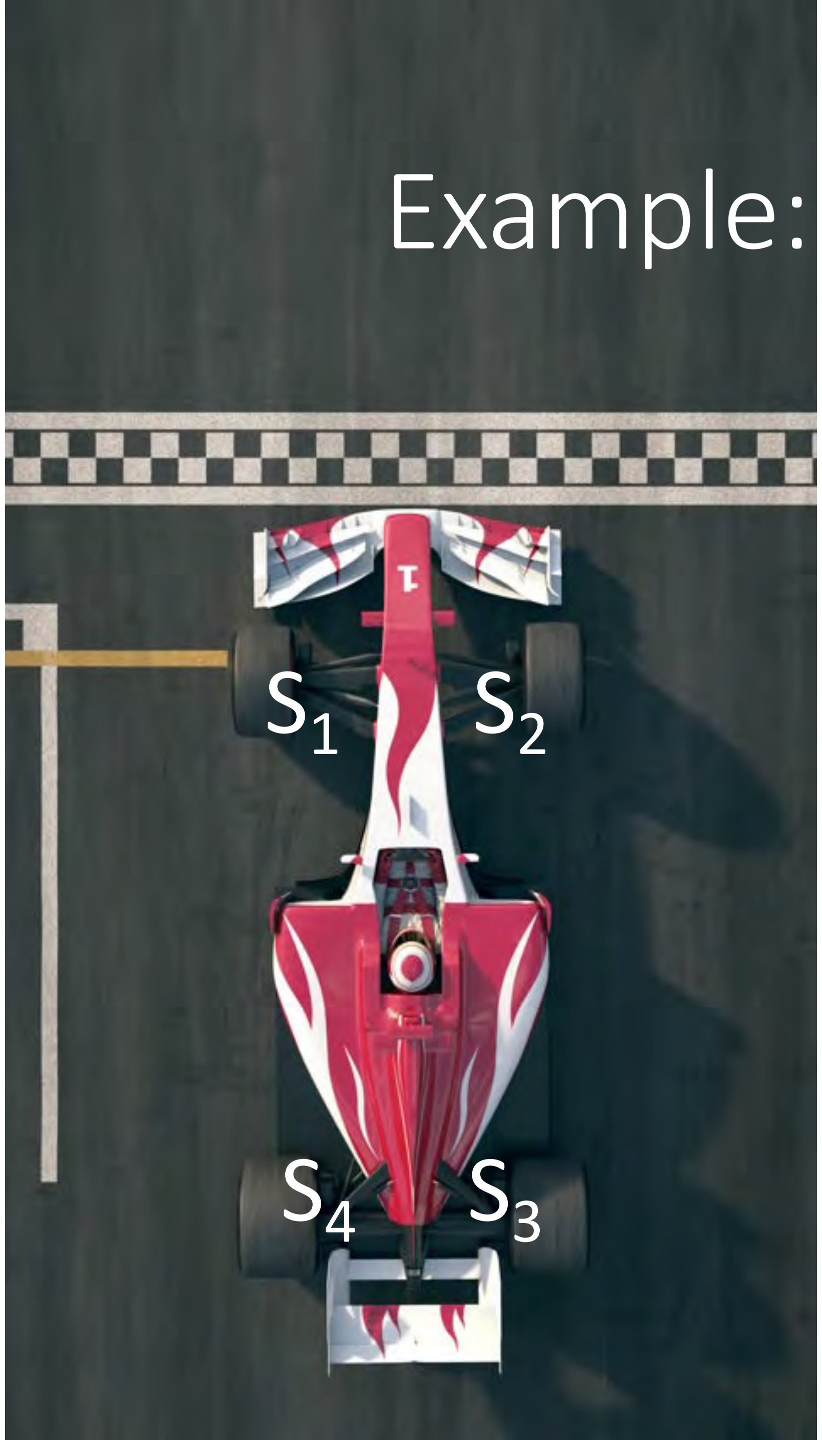
New composite measures (features):

$$T_1 = +\frac{1}{4}S_1 + \frac{1}{4}S_2 + \frac{1}{4}S_3 + \frac{1}{4}S_4$$

$$T_2 = +\frac{1}{6}S_1 - \frac{1}{2}S_2 + \frac{1}{6}S_3 + \frac{1}{6}S_4$$

$$T_3 = +\frac{1}{6}S_1 + \frac{1}{6}S_2 - \frac{1}{2}S_3 + \frac{1}{6}S_4$$

$$T_4 = +\frac{1}{6}S_1 + \frac{1}{6}S_2 + \frac{1}{6}S_3 - \frac{1}{2}S_4$$



Example: Features

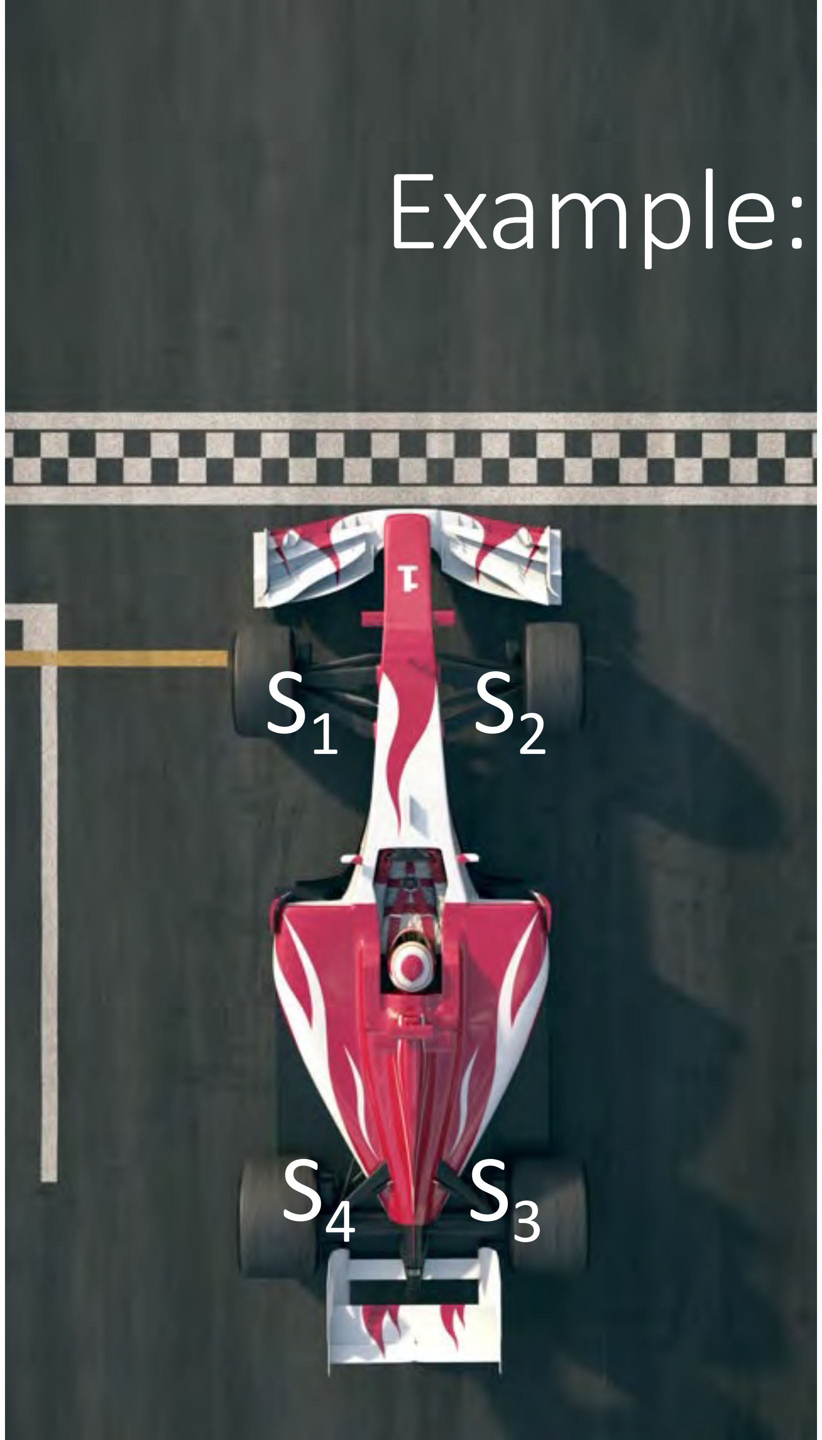
Original measures (variables):

S_1, S_2, S_3, S_4

New composite measures (features):

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \begin{matrix} = \\ = \\ = \\ = \end{matrix} \begin{bmatrix} +\frac{1}{4} & +\frac{1}{4} & +\frac{1}{4} & +\frac{1}{4} \\ +\frac{1}{6} & -\frac{1}{2} & +\frac{1}{6} & +\frac{1}{6} \\ +\frac{1}{6} & +\frac{1}{6} & -\frac{1}{2} & +\frac{1}{6} \\ +\frac{1}{6} & +\frac{1}{6} & +\frac{1}{6} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix}$$

Example: Features



Original measures (variables):
S_1, S_2, S_3, S_4

New composite measures (features):

$$T = W^T S, \qquad W^T = \begin{bmatrix} +\frac{1}{4} & +\frac{1}{4} & +\frac{1}{4} & +\frac{1}{4} \\ +\frac{1}{6} & -\frac{1}{2} & +\frac{1}{6} & +\frac{1}{6} \\ +\frac{1}{6} & +\frac{1}{6} & -\frac{1}{2} & +\frac{1}{6} \\ +\frac{1}{6} & +\frac{1}{6} & +\frac{1}{6} & -\frac{1}{2} \end{bmatrix}$$

Principal Component Analysis (PCA)

PCA is a method for computing new features from existing variables according to a generic principle.

PCA will compute the weight matrix W for the new composite measures T_i, which are called principal components, so the data are now measured according to these new composite measures

NOTES: The original variables must be centered (i.e., have mean zero)

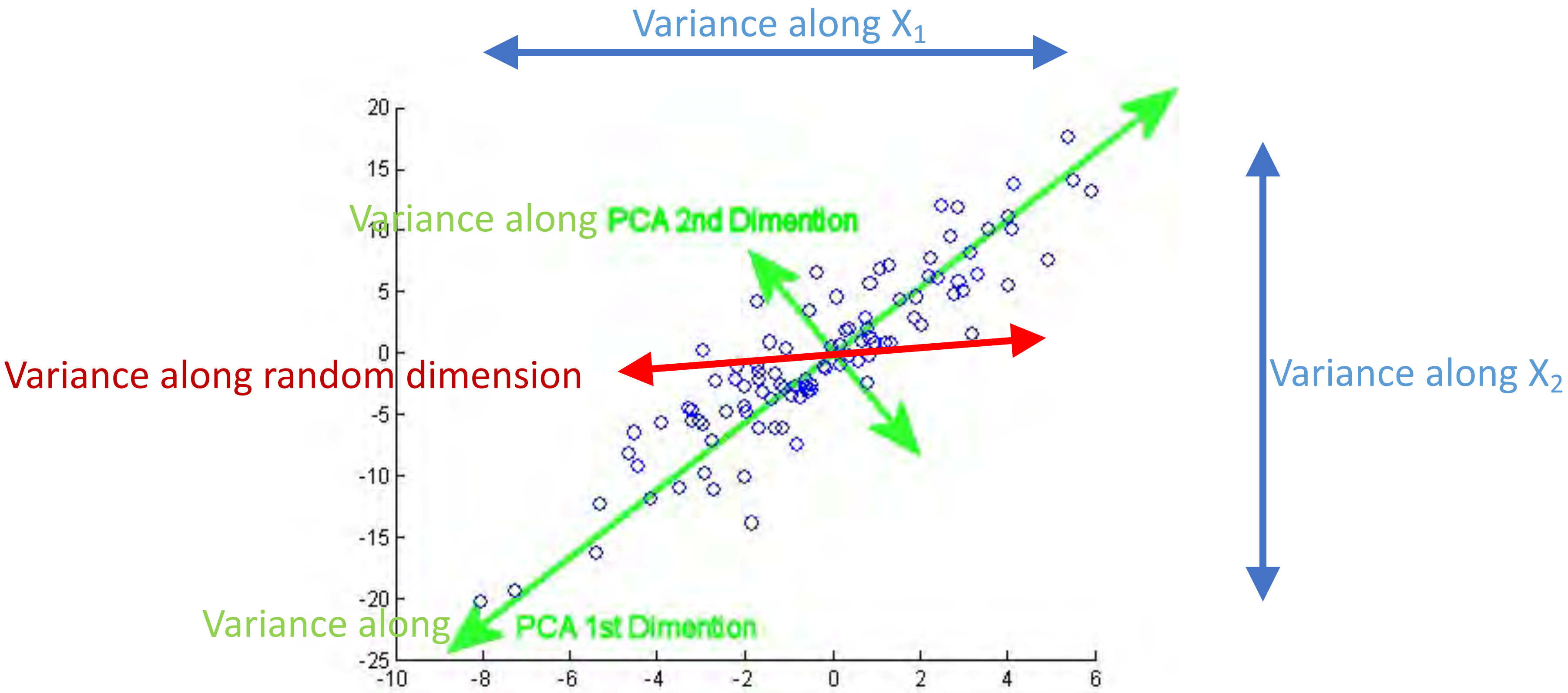
Original X (variables):

Observation ID	S_1	S_2	S_3	S_4
1				
…				
N				

Transformed X (features/components):

Observation ID	T_1	T_2	T_3	T_4
1				
…				
N				

PCA principle:
PC_1 is the direction of maximum variance



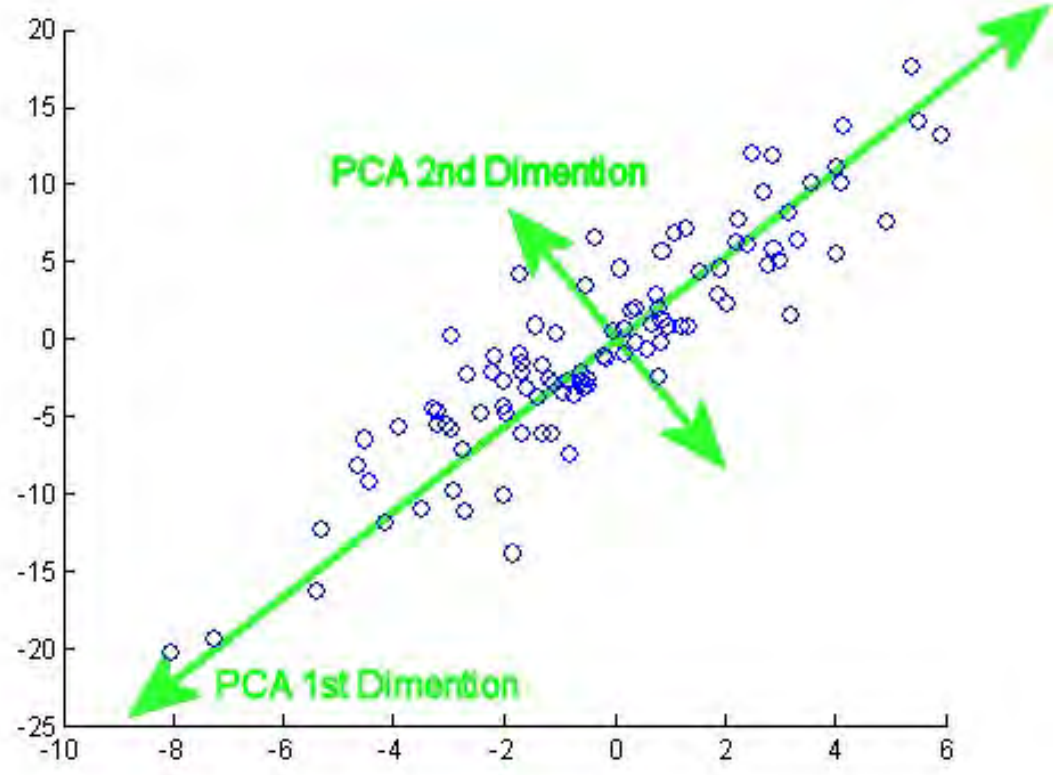
$$Var(X_1) + Var(X_2) = Var(PC_1) + Var(PC_2)$$

PCA principles, continued

Principal Components are orthogonal:

Principal Components are ordered:

- every principal component captures less variance than the ones before, i.e., $Var(PC_1) \geq Var(PC_2) \geq \ldots$



What is the curse of dimensionality?

More dimensions, more problems

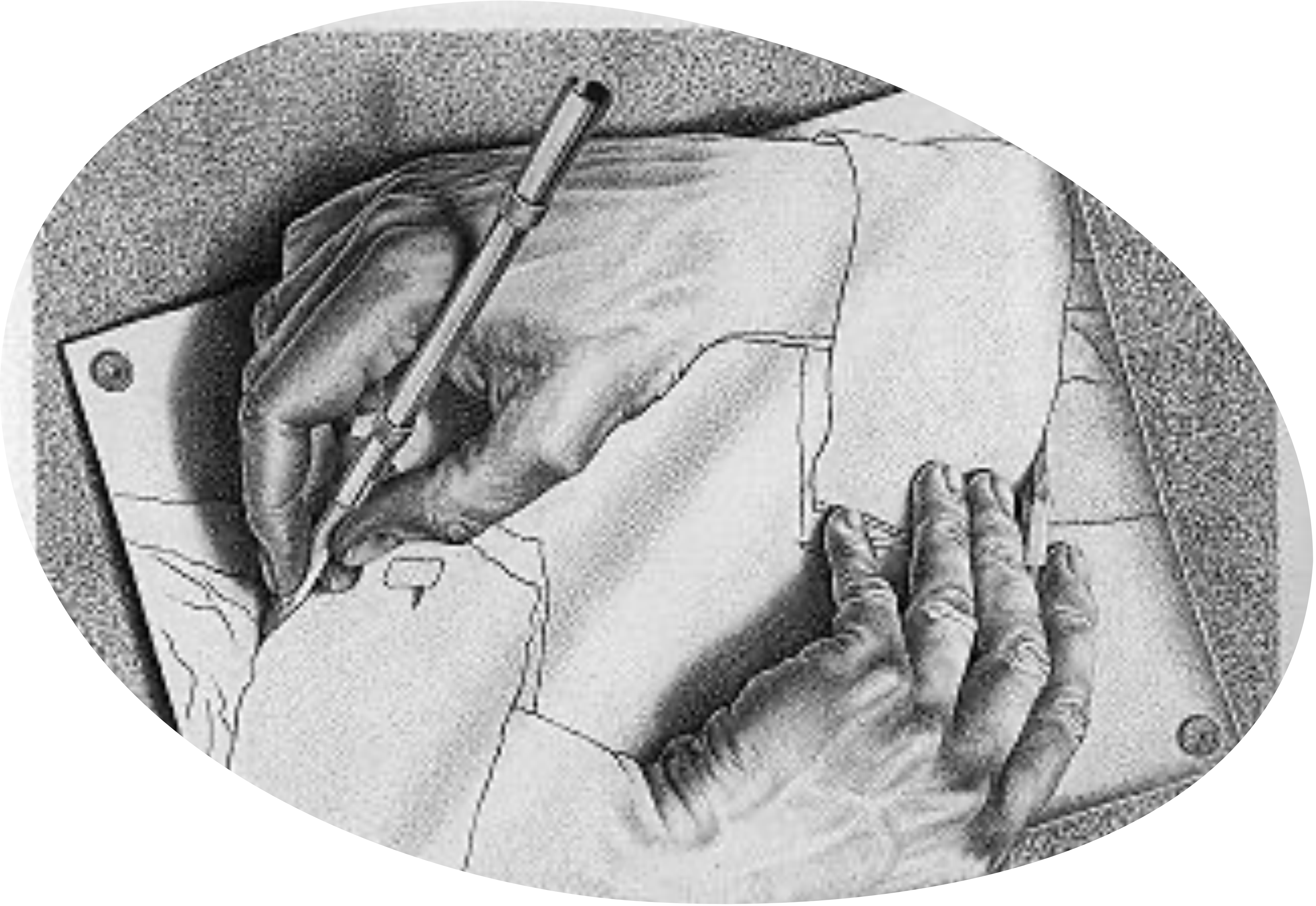
1-D
If I dropped my keys somewhere along the path between my car and my house, it would take only a few minutes to walk the straight path and find them.

2-D
If I dropped my keys somewhere in my yard while mowing my lawn, it could take me hours to search the whole yard to find them.

3-D
If a dropped my keys in one of the offices in PGH while going door-to-door delivering girl scout cookies, it would take days to search all the building floors to find them.

Dimension Reduction with PCA

Work with only the top few principal components, since they capture most of the variance

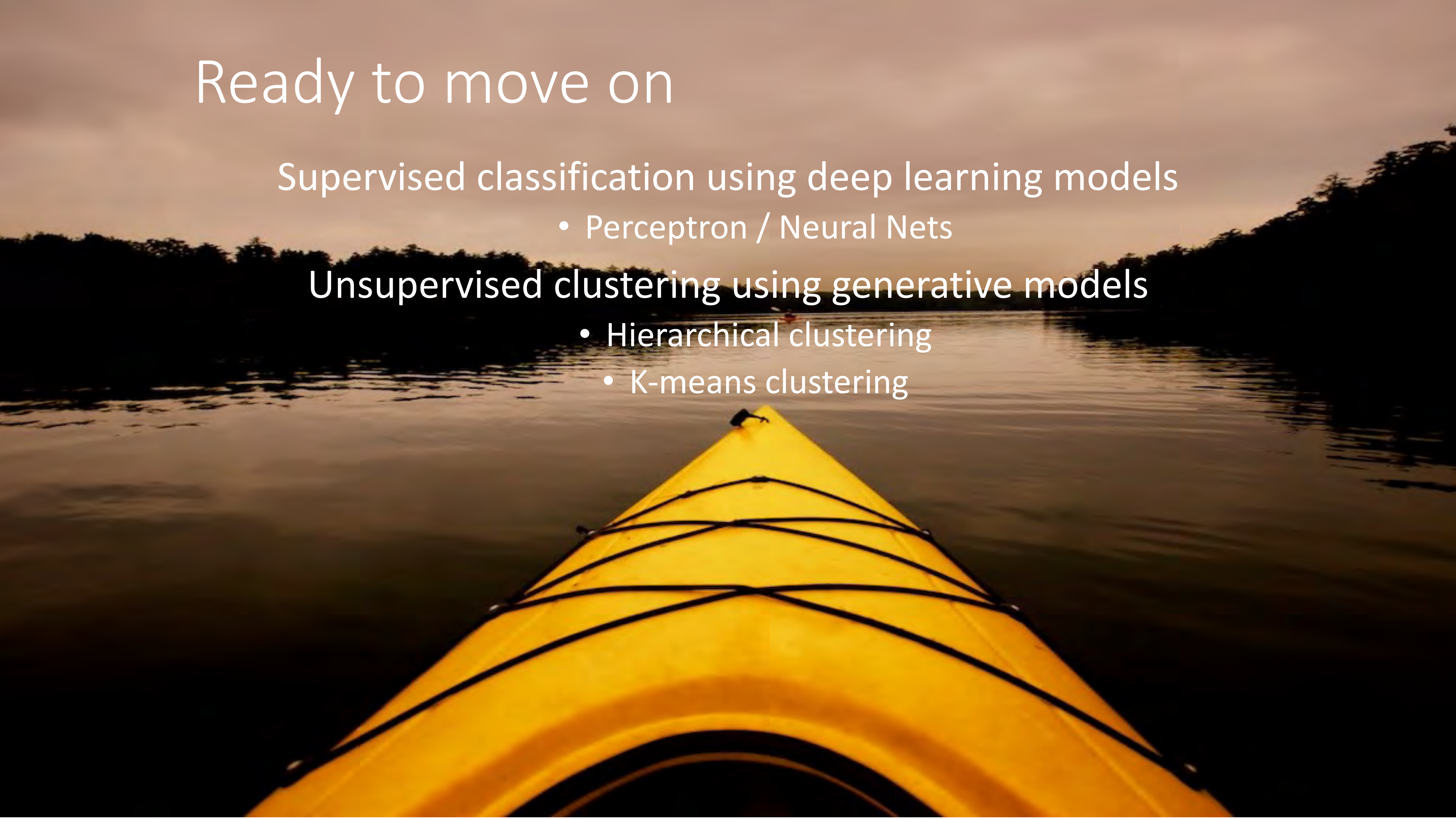


Hands-on Example: PCA

Homework Assignment #2
Due Tuesday (July 6), 11:59 pm (Central)

Your assignment is to create a Jupyter notebook that demonstrates how to do the following (use methods discussed in the class materials shared so far):

- Load the dataset in the file named winequality_white.csv and produce at least one table and one graph that summarize the dataset statistics. Separate the data into training and testing datasets and set up a classification problem: predicting the quality value (variable with seven classes labeled 3, 4, 5, …, 9) based on the values of all the other variables (acidity, alcohol, pH, etc.). (2 points)
- Train and tune (via cross-validation) at least two different models based on Decision Trees (e.g., DecisionTreeClassifier, RandomForestClassifier); Consider at least two different hyperparameter options (e.g., tree depth). (5 points)
- Train and tune (via cross-validation) at least two different SVM models based on different kernel options (e.g., linear and sigmoid) and regularization parameters (different values of C). (5 points)
- Use the make_pipeline() method to study and describe the impact of feature selection (using different n_component values for PCA) and data scaling (e.g., MinMaxScaler) on the performance of the tuned SVM from Step 3. (5 points)
- Test the performance of the best method you found using the test set you created. Discuss your overall results. (3 points)



Ready to move on

- Supervised classification using deep learning models
  - Perceptron / Neural Nets
- Unsupervised clustering using generative models
  - Hierarchical clustering
  - K-means clustering