DataSci 400

lesson 10: simple statistics

Seth Mottaghinejad

today's agenda

- **univariate** measures
- **bivariate** measures
- statistical and ML pitfalls
    - **simpson's paradox**
    - **anscombe's quartet**
    - **spurious** correlations
    - fishing for **significance**
    - data **leakage**

measures of central tendency

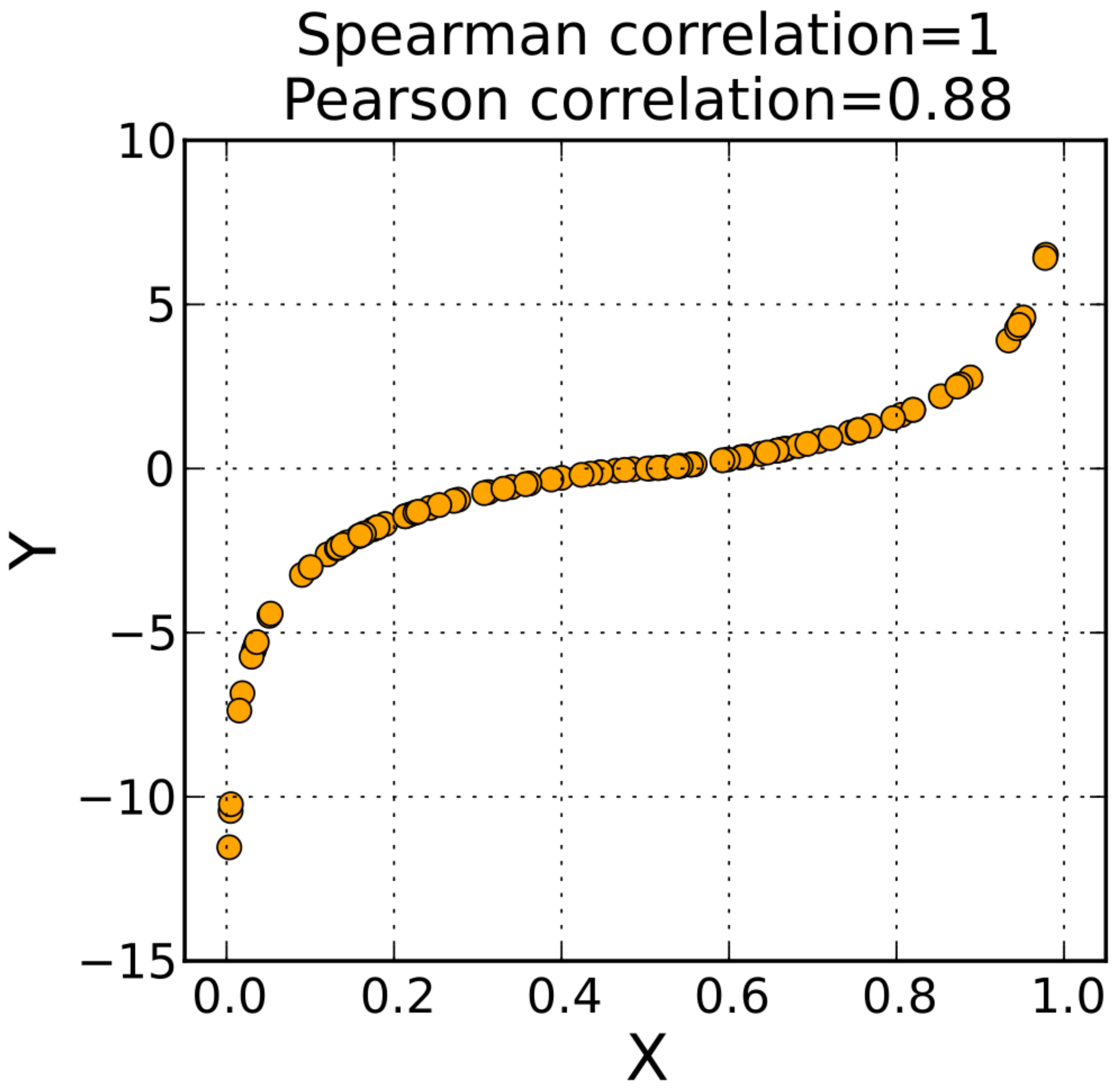
- what's the "average" value of my data
- **mean:** $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$
- **weighted mean:** $\bar{y} = \sum_{i=1}^{n} w_i y_i$ where $\sum_{i=1}^{n} w_i = 1$
- **median:** sort the data, find the middle value (if there are two middle value just take their average)
  - the mean is pulled toward **outliers**, the median is not
- **mode:** most common category in categorical data

measures of spread

- **spread** is also called **dispersion**, **variability**, **uncertainty**
- **variance:** $\mathrm{var}(y) = \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2$
- sometimes we divide by n instead of $n - 1$
- the summation term is called **total sum of squares (SST)**, think of it as **total variability**
- a model's job is to **explain away** as much of that variability as possible, leaving us with $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$, or **SSE**
  - **the coefficient of determination:** $R^2 = 1 - \frac{\mathrm{SSE}}{\mathrm{SST}}$

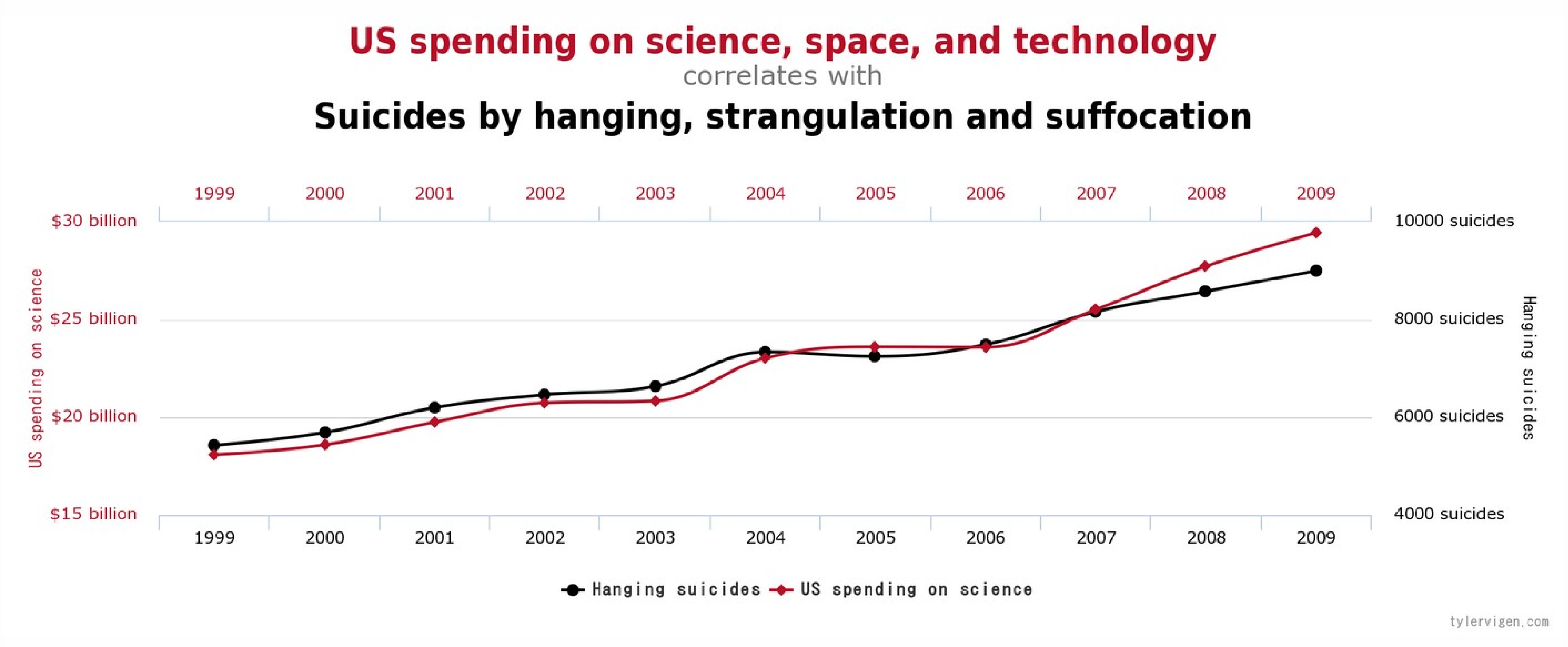
bivariate measures

- **Pearson's correlation** looks for a **linear** relationship
- **Spearman's rank correlation** is Pearson's correlation applied to **ranks** (min = rank 1, max = rank n)
- image source: www.wikipedia.com



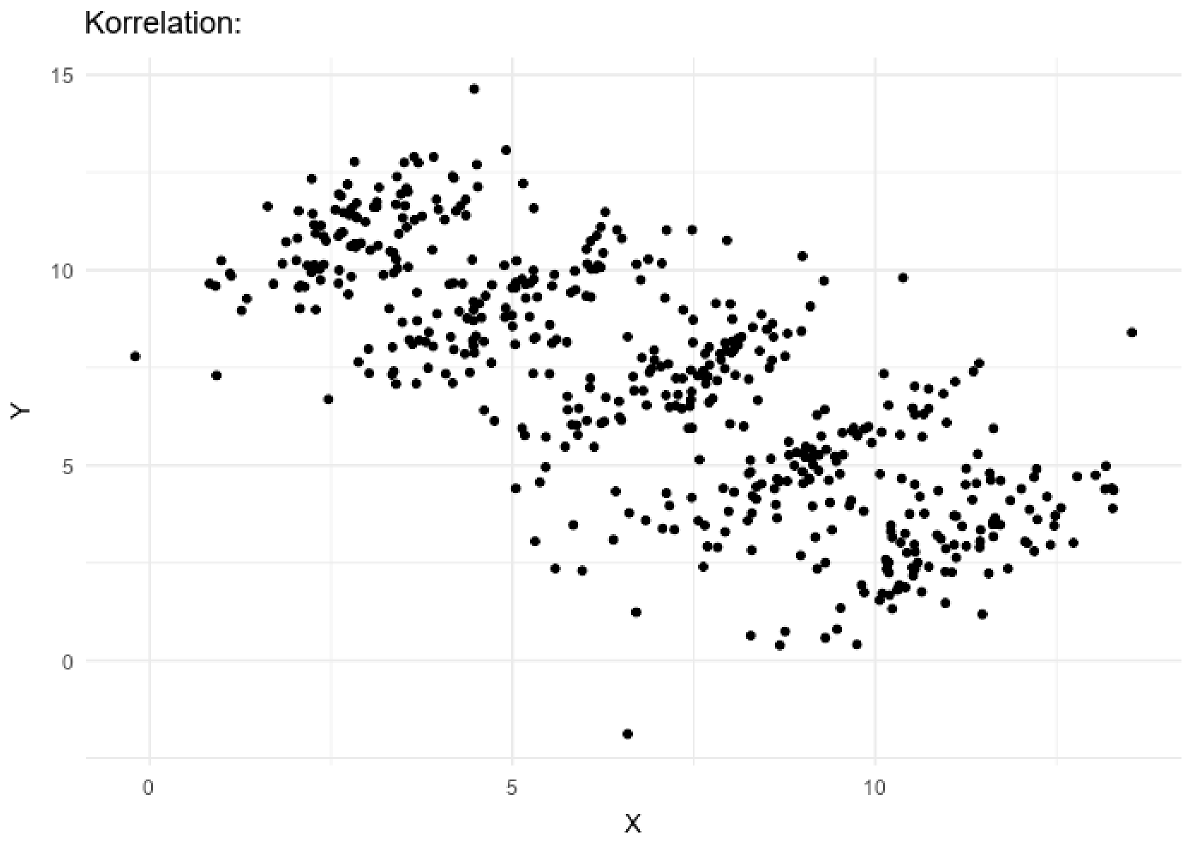
break time

Spurious correlations



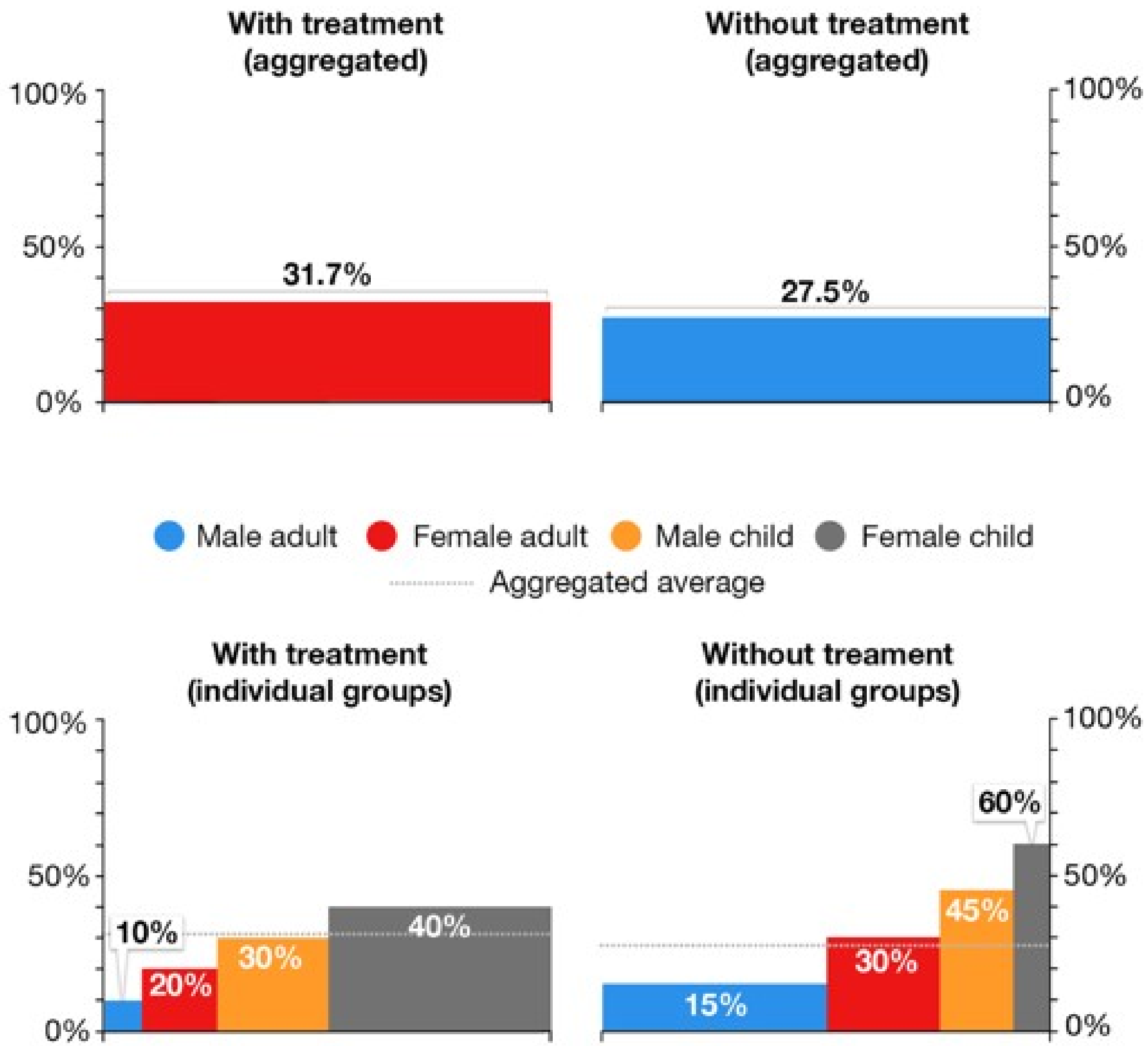
Simpson's paradox

- a trend appears to go in one direction when data is broken into groups, but the other direction when combined
- image source: www.wikipedia.com



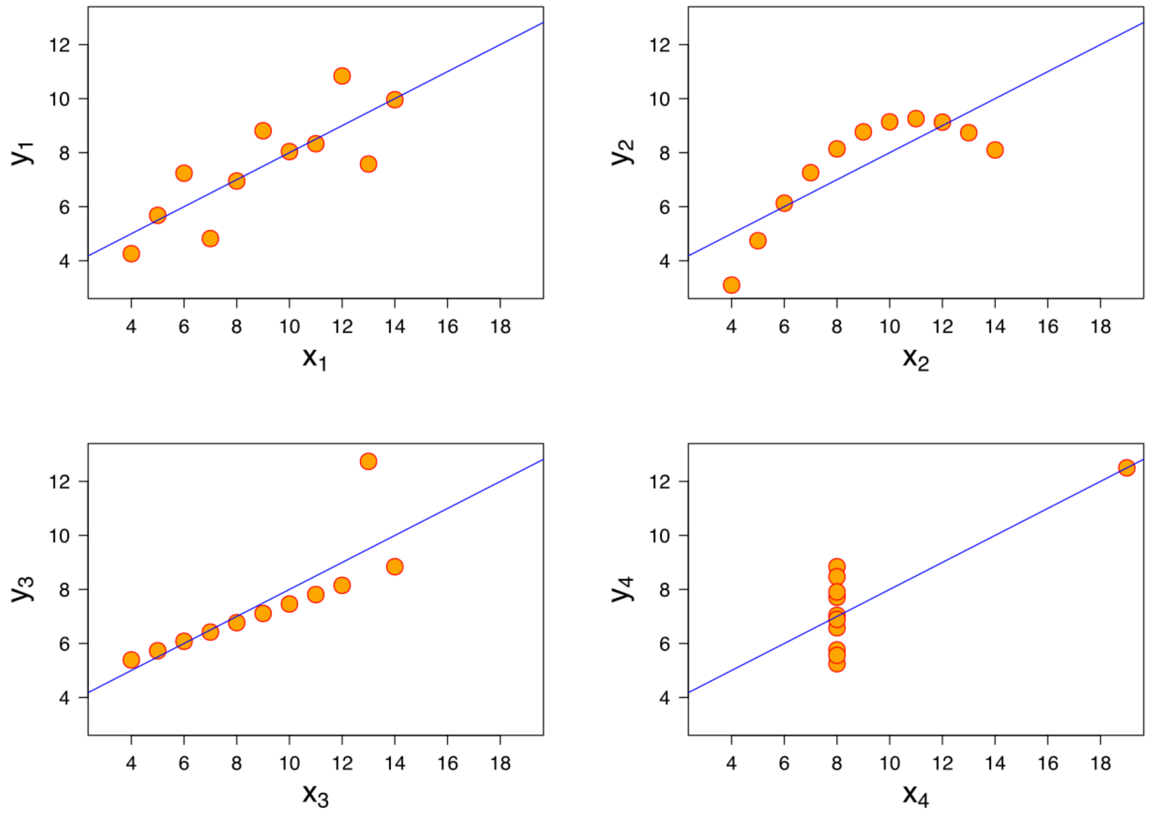
Simpson's paradox

- with categorical data, Simpson's paradox can occur when the relative size of the groups is different between the control and treatment
- image source: theconversation.com



Anscombe's quartet

- x and y in the four data sets have the same **mean**, **variance**, **correlation** and **trend line** $y = a + bx$ if we use linear regression to find a and b
- image source: www.wikipedia.com



fishing for significance

- in research, you can **fiddle** with your statistics until your tests show **significance**, e.g. by increasing your sample size
- this **publication bias** research findings hard to **reproduce** and slows the advancement of science
- in ML you can also **fiddle** with your hyper-parameters and evaluating model accuracy on **test data** over and over again
- but **hyper-parameter tuning** is an essential ML task: only catch is we must use **validation data** for **model selection**
- **test data** is only used **once** to evaluate **final model**

lab time

you need to Z-normalize your features as part of your pre-processing: which one of these is the right way of doing it?

1. use mean and std dev of the **whole data** (prior to splitting) to normalize
2. use mean and std dev **of the training data** to normalize training data, and mean and std dev **of the test data** to normalize test data
3. use mean and std dev of the training data to normalize **both** the training data and the test data

data leak in machine learning

- we must train models with an eye toward **scoring**
- **information** from training data can (and sometimes needs to) flow to test data, but not the other way around
  - to be **consistent**, test data must be pre-processed in the same way training data was pre-processed
- if a feature is not available at scoring time, it cannot be available at training time
- any information from the test data **leaked** to the training data may **inflate** test performance

the end