Mechatronic Engineering program

# Python for machine learning and data science:
# 1: Introduction and data management

Ziemowit Dworakowski

*AGH University of Krakow*

# Before we begin…

Lecture presentations (and all other course materials) will be available on my webpage at least 2 days before given lecture or laboratory:

http://galaxy.agh.edu.pl/~zdw/students.html

I recommend making notes only for stuff that is **NOT** on the slides. During lectures just focus on understanding relations and thought process. Knowledge will be easier to memorize after you see ***the big picture***

**Your feedback is important – if you feel something is not going as it should, notify me as soon as possible, don't wait till the end of semester.**

# What this course is all about?

**We will be learning Python (a bit)**

**We will be learning Machine Learning (more than before)**

**We will be learning actual teamwork**

**We will be learning to think like data scientists**

# What should you know already?

**„AI stuff" from *„Intelligent methods"* courses:**

- Basics of optimization (gradient and 1+1 methods)
- Basics of classification (algorithms that we've used, basic vocabulary and rules)



If you attended the polish version of the course before or just want a quick reminder, download my lecture slides on optimisation and classification, read through them and you should be good to go:

http://galaxy.agh.edu.pl/~zdw/Materials/SP/2022_PSIWSUM_Notes_EN_01_Optimization.pdf
http://galaxy.agh.edu.pl/~zdw/Materials/SP/2022_PSIWSUM_Notes_EN_03_Classification.pdf

# Course team



Dr hab. inż.
**Ziemowit Dworakowski,**
*Course coordinator,*
*Lectures and projects*

**D3, 4.17** zdw@agh.edu.pl
Office hours:
Wednesday, 8:00 – 10:00



Dr inż.
**Adam Machynia**
*Laboratories and projects: IME*

**D3, 4.08** machynia@agh.edu.pl
Office hours:
Thursday, 10:00 – 11:30 *

*Please notify respective teacher via email at least a day before you plan to use office hours.*

# What about it being English-only course? (for IME students)

- Lectures will be supported by notes

- Most of the tasks will be performed in teams, make sure that in each team we have at least one person with decent English.

- All of the written work (reports) should be grammatically correct, without typos, etc. But you have **ChatGPT** and **Grammarly** so that's not an issue

- Your English skills will not affect your grades in tests – you just need to be communicative, that is enough.

# So what exactly are contents of this course?

**We have:**

**5 lectures**

5 project classes

8 laboratories

L1: Introduction & data management

L2: Basics of scientific method and data interpretation

L3:Regression & classification revisited – from a learner perspective

L4: Challenges and problems in data science, data leakage

L5: Interpratation of results, quality measures & model improvement

# So what exactly are contents of this course?

**We have:**

5 lectures

**5 project classes**

8 laboratories

P1: Teamwork organization, agile methodology

P2: Waterfall methodology, project planning and scheduling

P3: Business presentation, selling your ideas

P4: Problem session – how to deal with issues in team

P3: Final presentations and assessment, feedback session

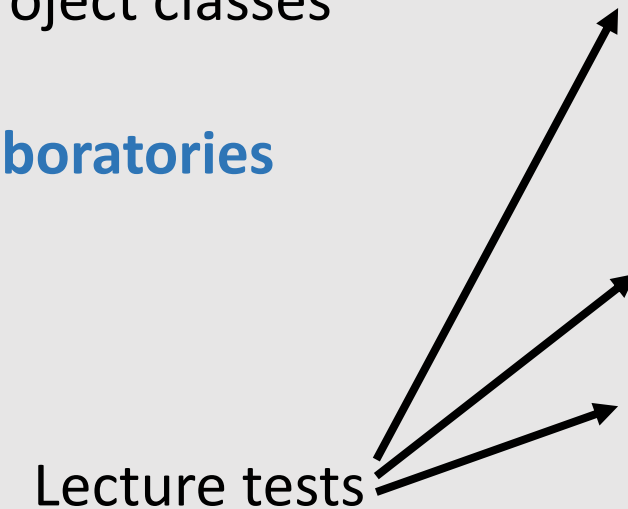
# So what exactly are contents of this course?

**We have:**

5 lectures

5 project classes

**8 laboratories**




L1: Python & Colab, loading and visualizing data

L2*: Drawing hypotheses and understanding datasets

L3: Feature selection

L4* & L5: Classification and regression

L6* & L7: Assessment and improvement of decision systems

L8: Conclusions and work evaluation

# OK, apart from the tests, how will we get a grade?

1) Passing all laboratories is required. Some will be graded, some (project-related) are just for a pass
2) All the tests are required (with a positive grade)
3) At least 4 out of 5 project classes are required, including the final one
4) You will prepare project report and do a project presentation/defense

Final grade = ( 2* Project grade + Average test grade + average lab grade) / 4

# How will projects be graded?

- Each project will include research of one particular dataset. The actual task will be formulated mid-semester based on your conclusions from initial research – and will be different for different teams working on the same dataset.

- Several of the laboratories will be devoted strictly to project-related tasks. During these laboratories you will be working on required project parts.

- Final project grade will be affected by:
  - **Presentation quality (team)**
  - **Report quality, both merit and form (team)**
  - **Project defense (individual)**

# Laboratories are again using „flipped classroom” approach…

So again we have 3.0, 4.0 and 5.0 tasks, you are supposed to work at home before classes, you (hopefully) don’t prepare any reports. Recommended scope includes doing at least tasks „for 3.0” on your own before classes.

„Project laboratories” will not be graded (they contribute to the final project report). They will, however, contain „tasks for 3.0” – which are required for project positive grade and „tasks for 5.0” which allow you to gain additional points for higher project grade. You are free to pick and choose which of these you find interesting and worth pursuing.

# What if something goes wrong?

If you don’t pass a laboratory (either due to lack of preparation or due to absence) – you prepare a standard report with one additional task selected by the LA. You may do so twice in total
(after that passing conditions will be set individually).

You can attempt any test up to three times.

Note, that right to 3 „attempts” extends also to project reports.

**Note also that my webpage contains a document on report writing (including laboratory and project reports)**

# What should you prepare in advance?

**Project classes:**

You should probably sign in to **Asana**
(We will use it to manage project work)

**Laboratories:**

On my webpage there is a **matlab-python handout** prepared by us to facilitate your initial steps in Python. Download it (possibly: print it as well?) and have it ready before the first laboratory.

# What is python?

**Easy for beginners**

**High-level programming language**

**Easily readable**

**Significant indentation**

**Dynamically typed**

**Functional**

**Object-oriented**

**Garbage collected**

**Modular**

**Free and open-source**

**Go-to option for machine learning**

# Where will we code?

**Standard (PC) approach:**

**1) Install Python**

**2) Configure environment**

**3) Install a preferred IDE**

**4) Add libraries as needed**

**5) Solve conflicts as needed**

**Colab (online) approach:**

**1) Have google account**

**2) Log in to colab, write your codes there**

**3) …  (that's it!)**

Feel free to configure your own PC environment and use it, you will probably learn much more this way. However, you are on your own in case of conflicts, problems or stability.

# How to share your codes in team?

**a.k.a: Why will encourage you to use GIT?**

- Version control
- Standard tool if you want to actually code in the future
- Possibility of easy cooperation
- Possibility of simultaneous off-line work



**On my webpage there is a supplementary handout for colab-GIT configuration**

# How will we approach python?

**Matlab is similar to Python**

**You all know Matlab**

**We'll learn via differences between Python and Matlab**

**You will be expected to work the differences on your own, using handout provided by us:**



https://galaxy.agh.edu.pl/~zdw/Materials/Python/Matlab%20-%20python%20handout.pdf

# What about GenAI?

**What is mandatory:**

- **Unprompted** acknowledgement of form of usage if it is used (e.g. *„report was rewritten for clarity using ChatGPT")*

- Source files (Full prompts) available on-demand

**What you can do:**

- Use it to correct codes (bug fixes)

- Use it to prepare fragments of reports from prompts or drafts (*note, you should store these prompts or drafts!*)

- Use ChatGPT to translate reports from polish to english

**What you should do:**

- Use ChatGPT to correct your texts (maintain consistent and easy-to-read style, correct typos, grammatical errors, etc.)

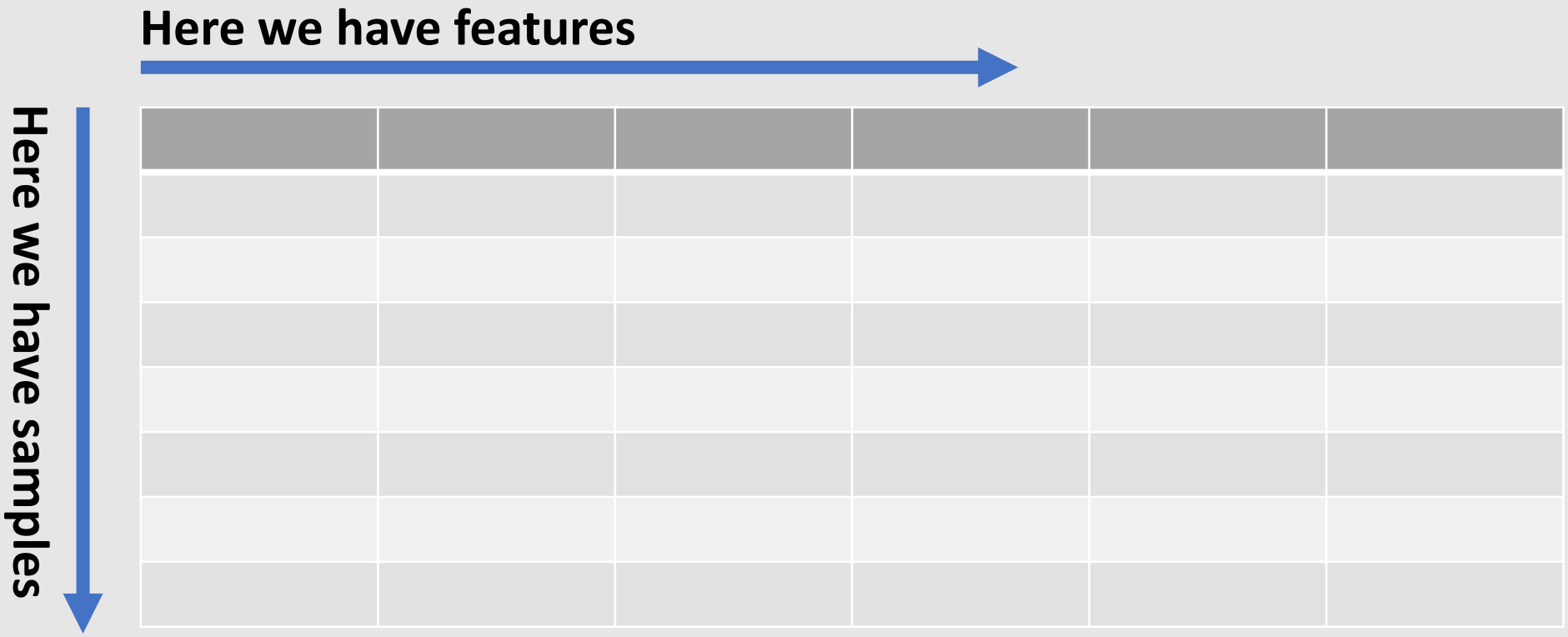
**What might get you into trouble:**

- Use ChatGPT to prepare complete codes

- Use ChatGPT to gain new knowledge (merit questions regarding e.g. machine learning)

- Use ChatGPT to do interpretation tasks for you

# Lets look at a typical dataset:



# Data types

Numerical

Categorical

Floating point

Integer

Binary (Dichotomus)

Polytomous

| Temperature |
|---|
| 34.31 |
| 24.51 |
| 23.23 |
| 31.19 |
| 33.10 |
| 32.54 |

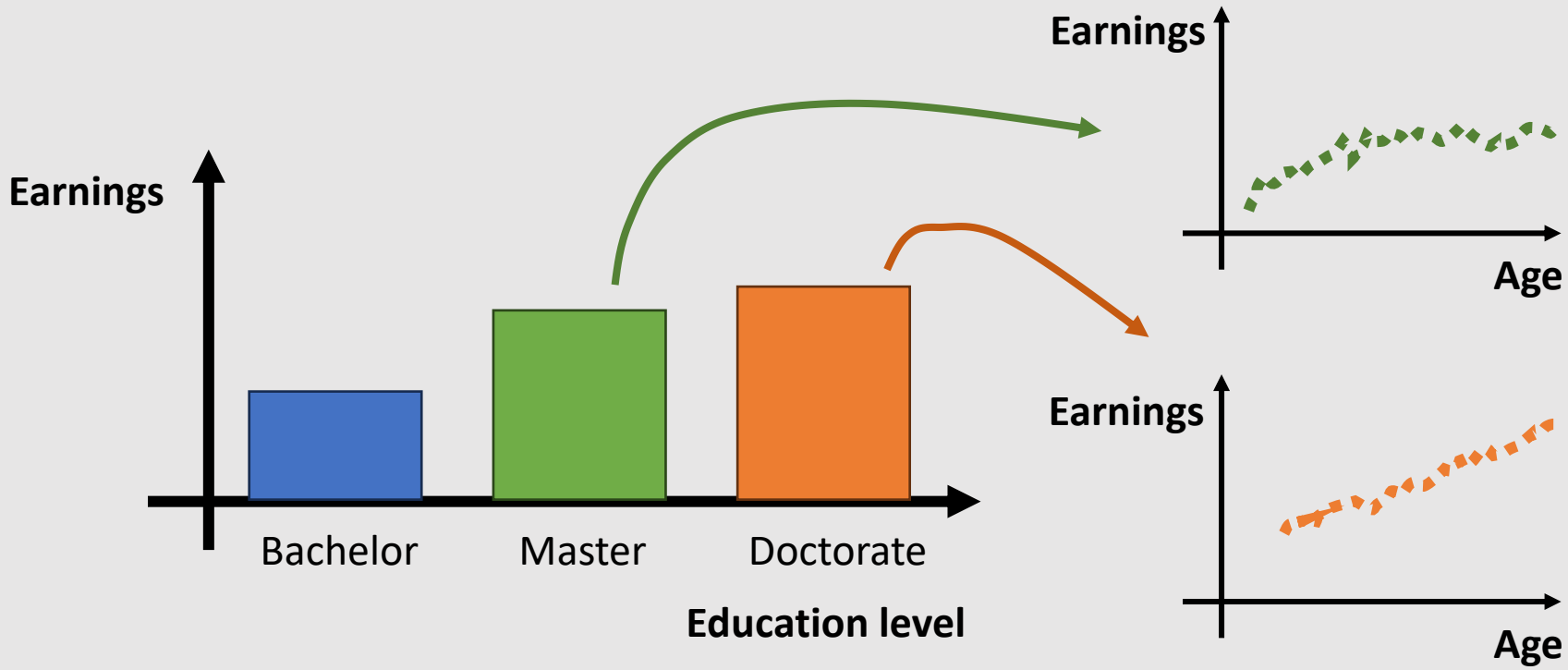
| Price |
|---|
| 1400 $ |
| 1200 $ |
| 700 $ |
| 800 $ |
| 1100 $ |
| 1000 $ |

| Employment |
|---|
| Yes (1) |
| No (0) |
| No (0) |
| Yes (1) |
| No (0) |
| No (0) |

| Education level |
|---|
| Bachelor |
| Bachelor |
| Master |
| Doctorate |
| Bachelor |
| Master |

# Data types

There are two ways of approaching categorical variables:
- Use them to divide and analyze dataset




Categorical

Polytomous

| Education level |
| --- |
| Bachelor |
| Bachelor |
| Master |
| Doctorate |
| Bachelor |
| Master |

# Data types

Categorical

There are two ways of approaching categorical variables:

- Use them to divide and analyze dataset
- Map them into numerical variables
  - a) Use **dummy variables**
  - b) Use integers

Polytomous

| Bachelor | Master | Doctorate |
|---|---|---|
| 1 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |

| Education |
|---|
| 1 |
| 1 |
| 2 |
| 3 |
| 1 |
| 2 |

**Only if a variable has „natural progression”!**

**We can do this, because „Master” is between „Bachelor” and „Doctorate” in education level**

| Education level |
|---|
| Bachelor |
| Bachelor |
| Master |
| Doctorate |
| Bachelor |
| Master |

# Data types

Categorical

There are two ways of approaching categorical variables:

- Use them to divide and analyze dataset
- Map them into numerical variables
  - a) Use **dummy variables**
  - b) Use integers

Polytomous

| Food | Clothing | Leisure |
|---|---|---|
| 1 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |

| Industry |
|---|
| 1 |
| 1 |
| 2 |
| 3 |
| 1 |
| 2 |

**Only if a variable has „natural progression”!**

**We can’t do this, because „Clothing” is not higher in any sense than „Food” and lower than „Leisure”**

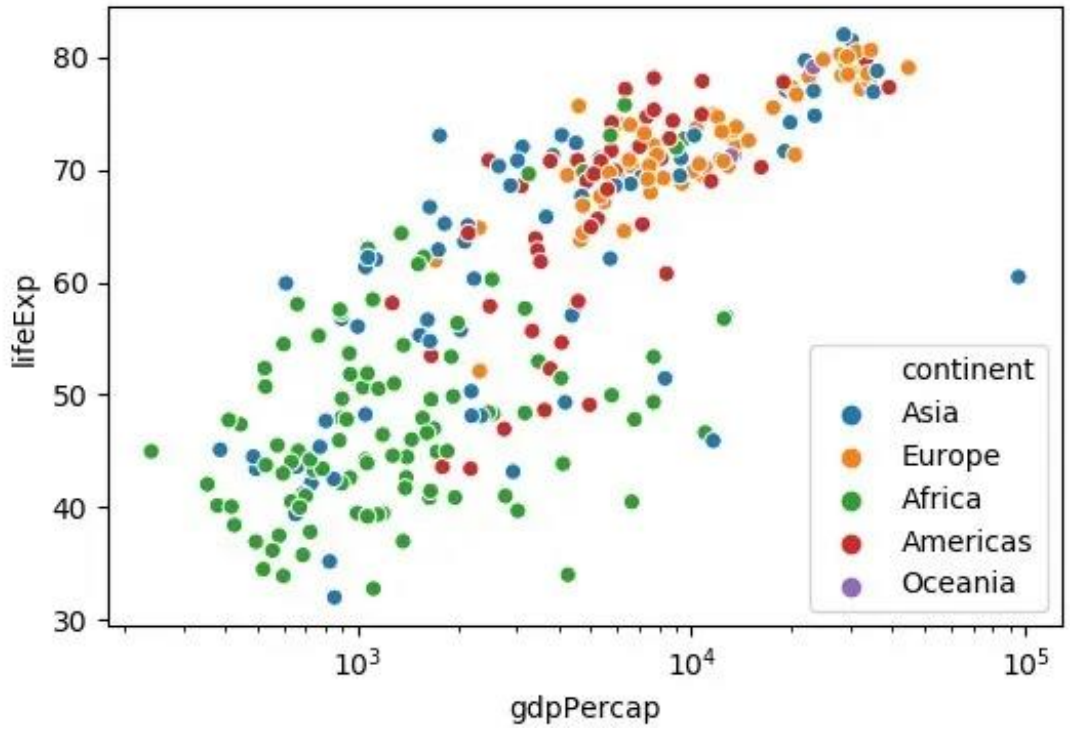
| Industry |
|---|
| Food |
| Food |
| Clothing |
| Leisure |
| Food |
| Clothing |

# How to visualize data?

**A time series plot**

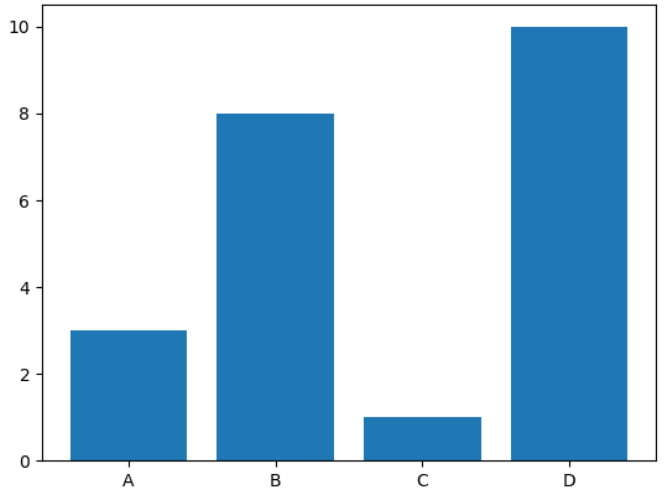



**A scatterplot**

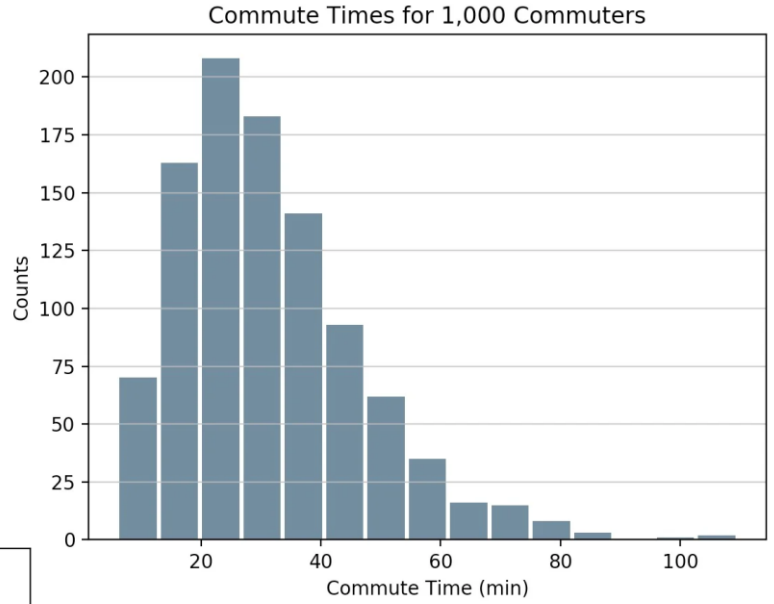


# How to visualize data?

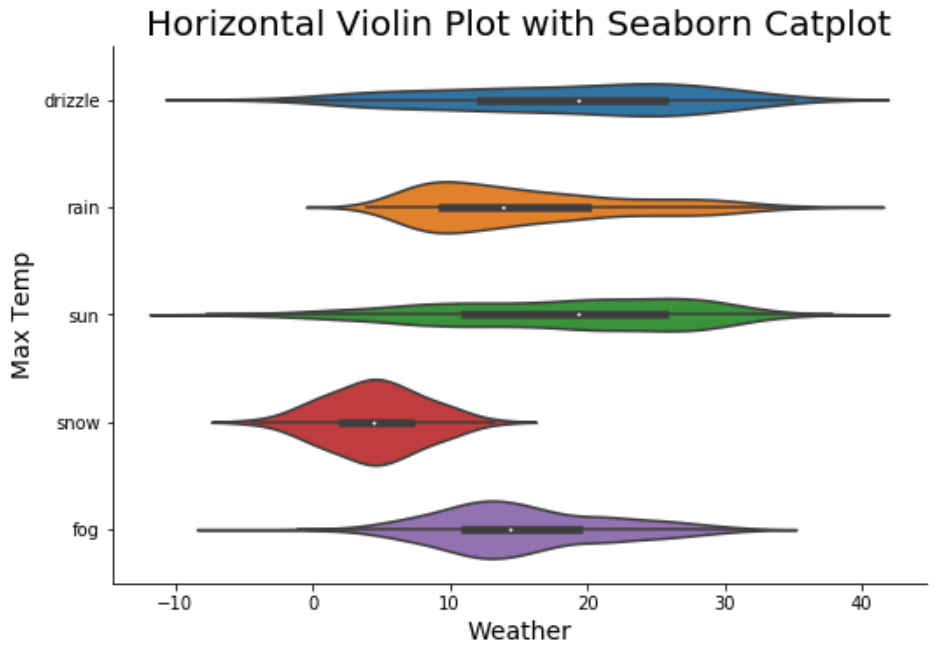
**A bar graph**



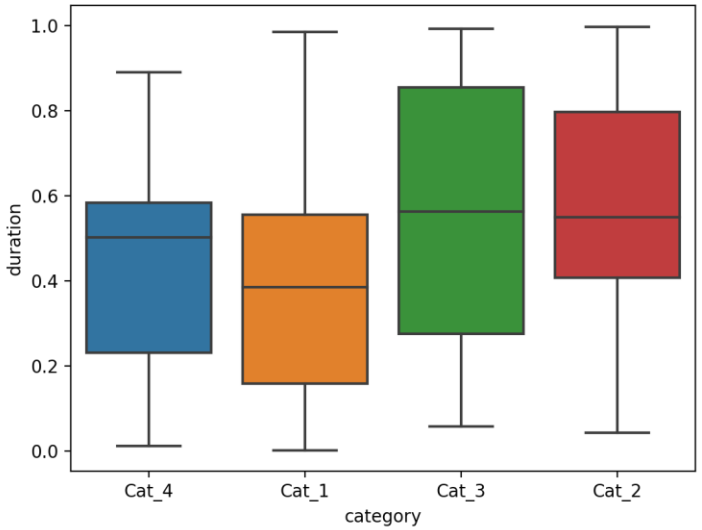

**A histogram**




**A violin plot**

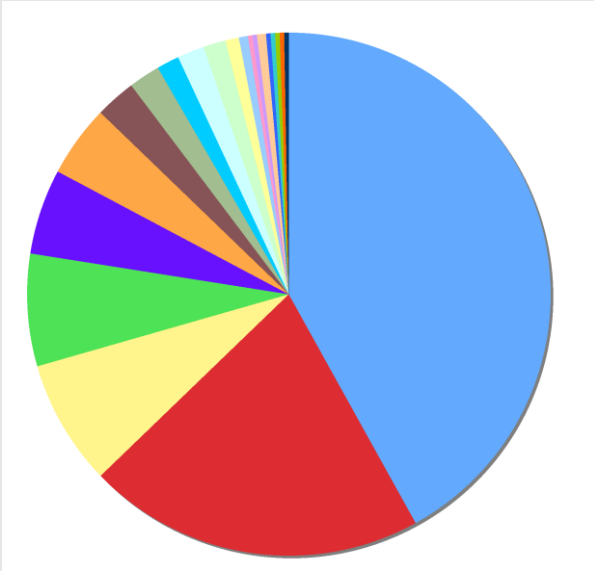



**A boxplot**



# Data visualization errors

1. **Using pie charts**



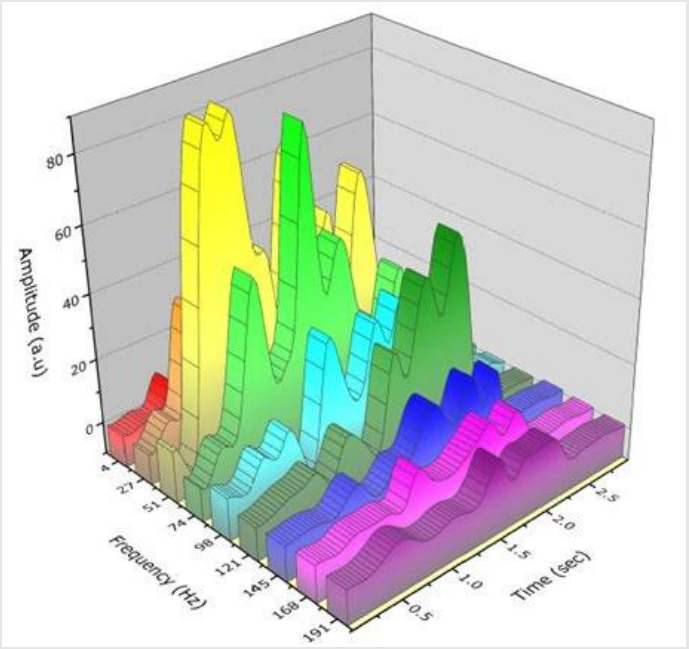
- You look at volumes but you should look at angles
- Very small values tend to be perceived similarily
- It is hard to compare categories

# Data visualization errors

1. Using pie charts
2. **Using fancy 3D visualizations**





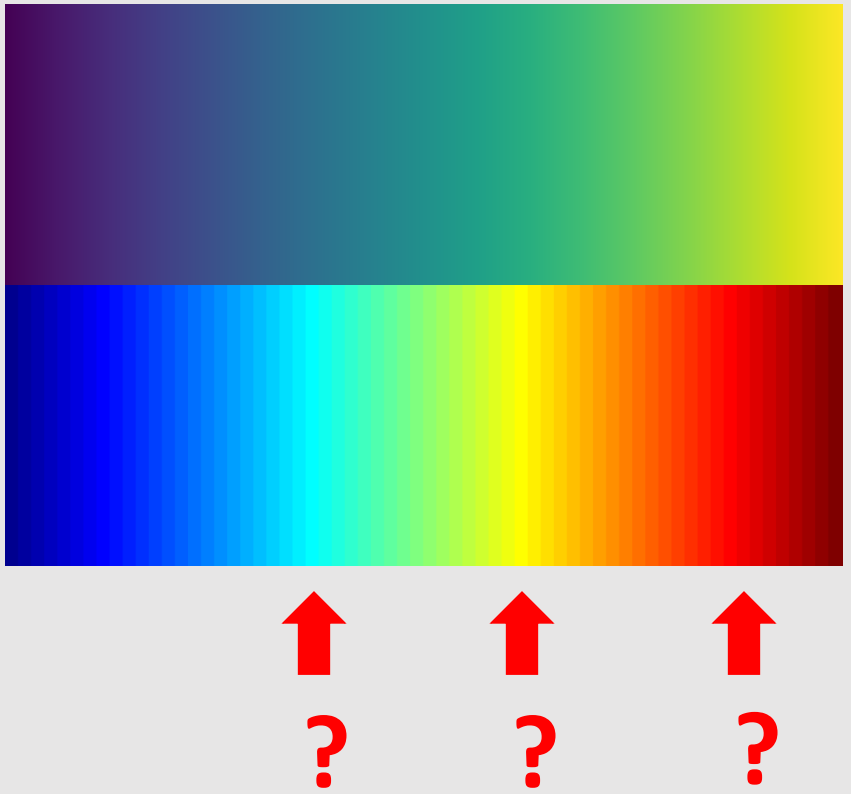
# Data visualization errors

1. Using pie charts
2. Using fancy 3D visualizations
3. **Using unclear colormaps**



**Viridis**

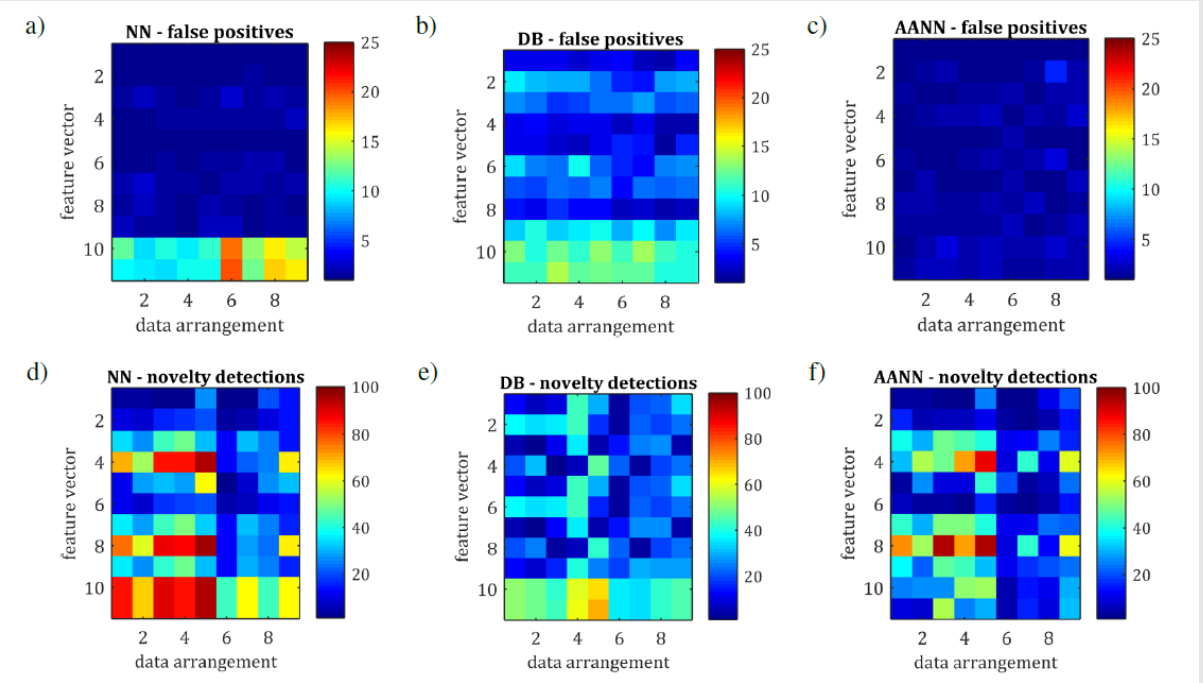
**Jet**




If you can name more than 3 colors of a colormap, it is probably not good. If you can see any sharp change in colors, it is bad as well.

# Data visualization errors

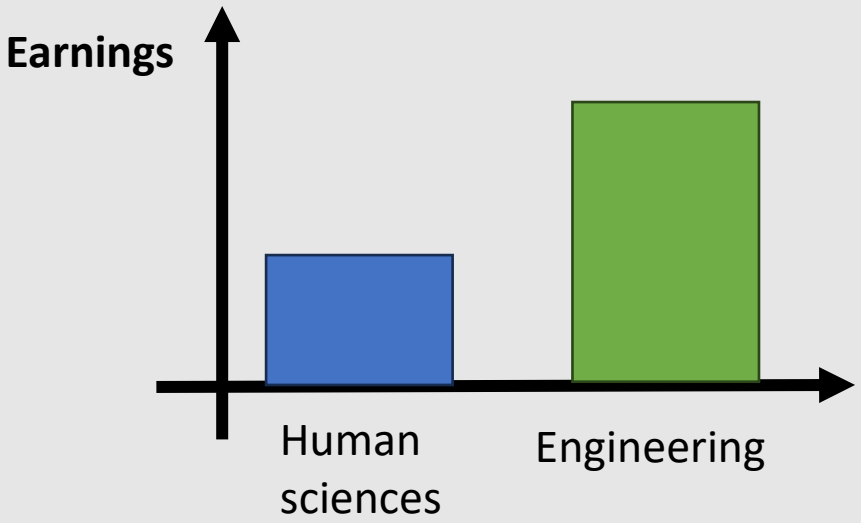
1. Using pie charts
2. Using fancy 3D visualizations
3. Using unclear colormaps
4. **Using too complex visualizations**




***Z. Dworakowski**, K. Dziedziech, A. Jabłoński „A novelty detection approach to monitoring of epicyclic gearbox health" Metrology and Measurement systems, Vol. 25 (2018), No. 3, pp.459-473*

# Data visualization errors

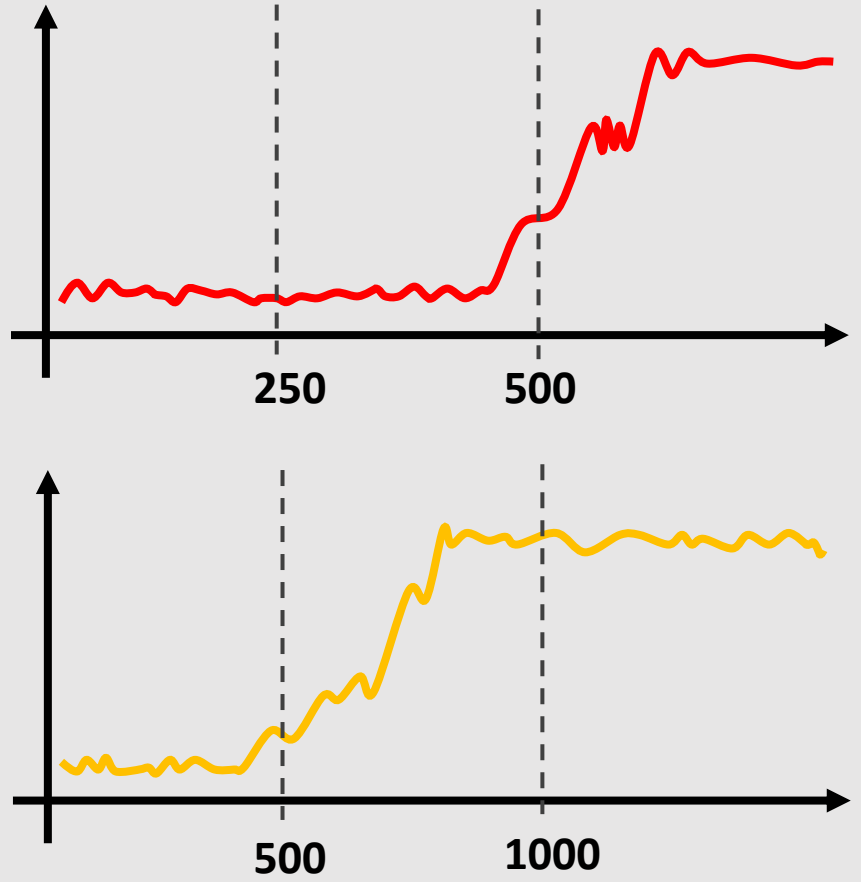
1. Using pie charts
2. Using fancy 3D visualizations
3. Using unclear colormaps
4. Using too complex visualizations
5. **Overcomplication of simple relations**




**Earnings in the field of Engineering are 120% higher than in the field of Human sciences.**

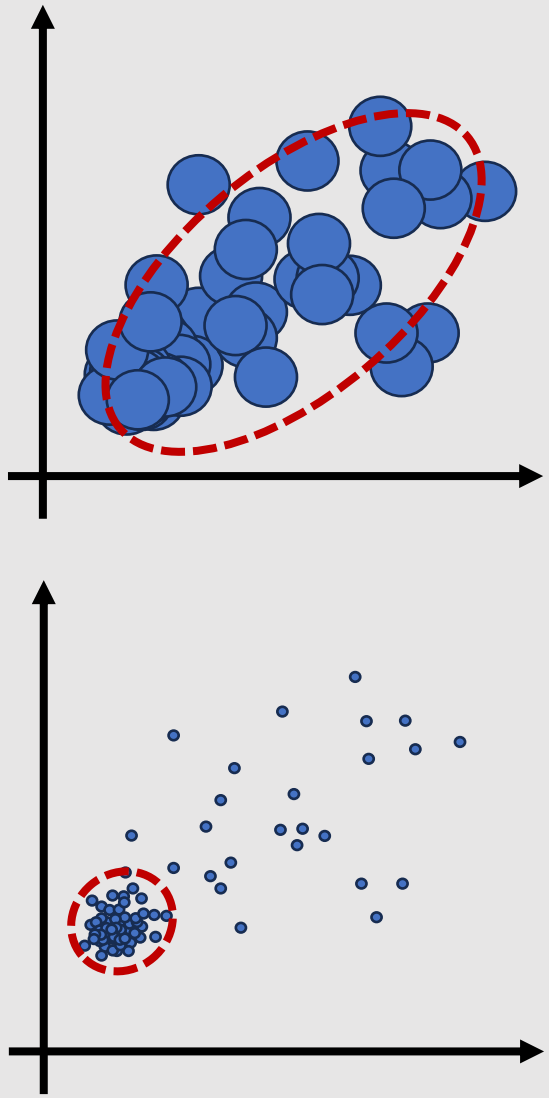
# Data visualization errors

1. Using pie charts
2. Using fancy 3D visualizations
3. Using unclear colormaps
4. Using too complex visualizations
5. Overcomplication of simple relations
6. **Inconsistent or unclear axes choices**



# Data visualization errors

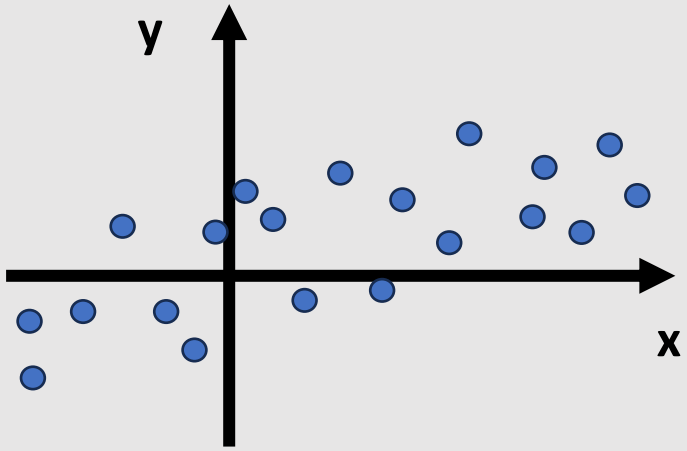
1. Using pie charts
2. Using fancy 3D visualizations
3. Using unclear colormaps
4. Using too complex visualizations
5. Overcomplication of simple relations
6. Inconsistent or unclear axes choices
7. **Occlusion in scatterplots**

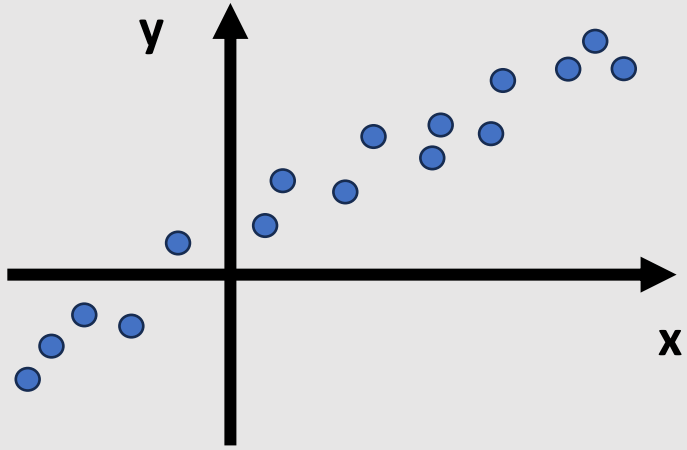


„If you torture the data long enough, it will tell you anything”

*John W. Tukey*

# Data visualizations are often not enough

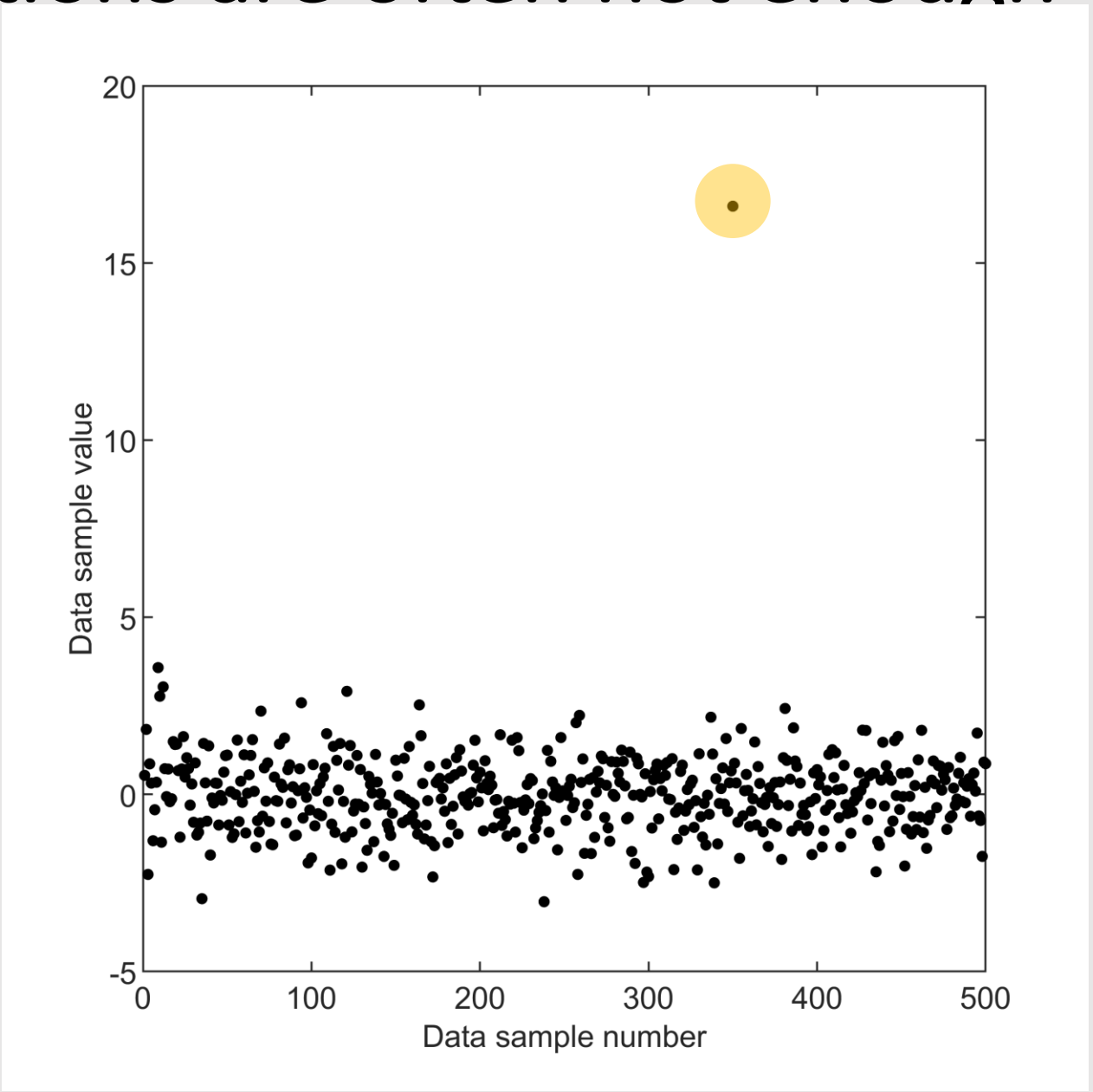


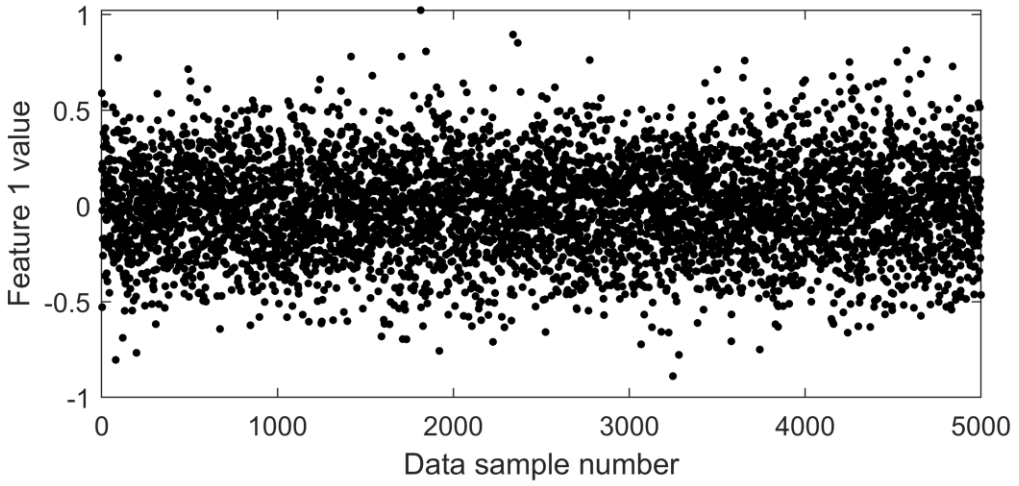

We can often look at two different datasets and roughly estimate the relations in data.

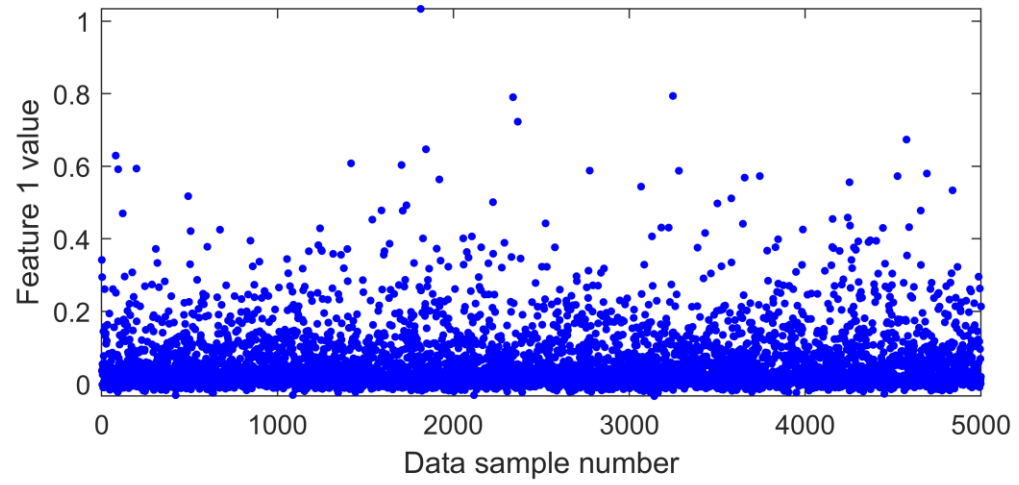
Is it true for high-dimensional datasets?

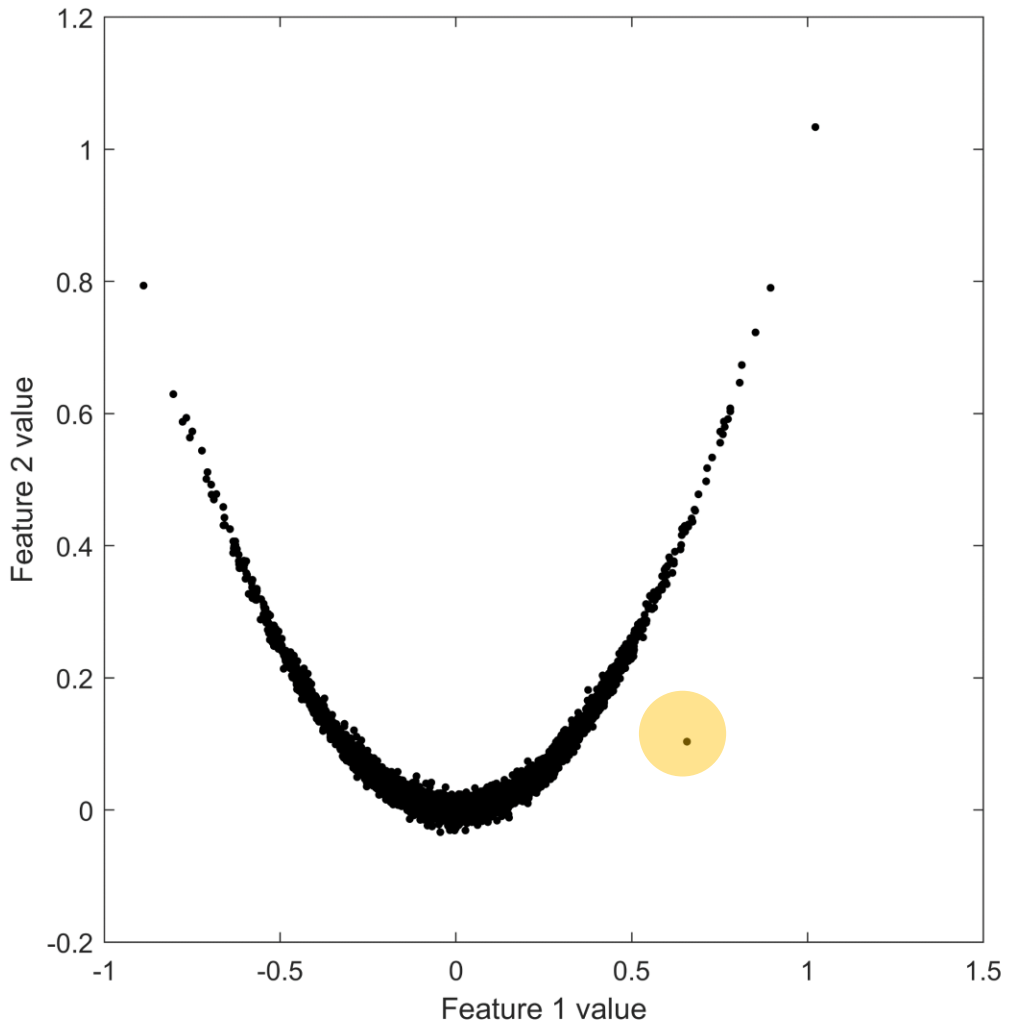
# Data visualizations are often not enough



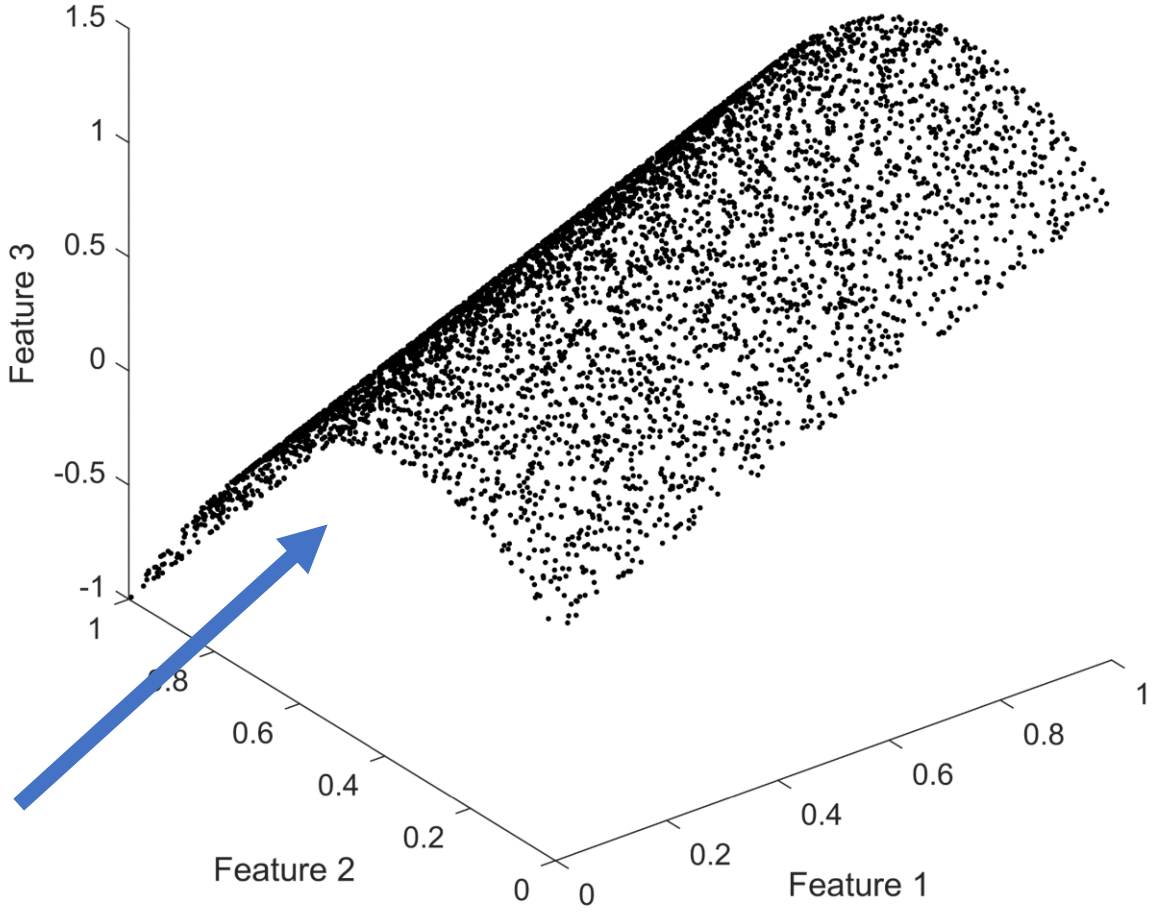
# Data visualizations are often not enough

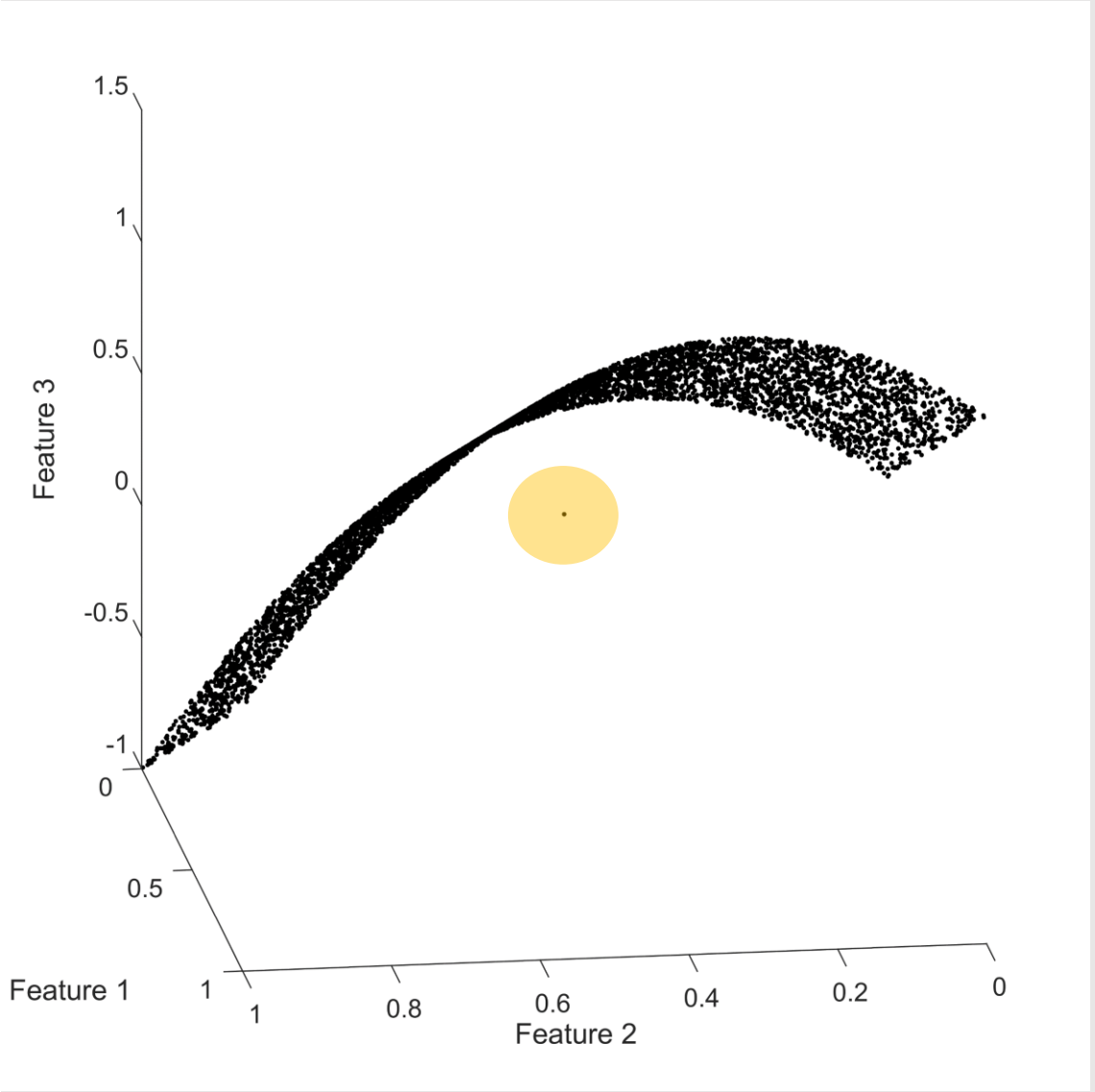


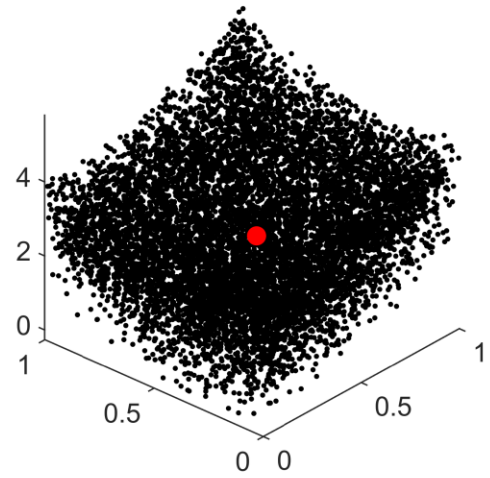



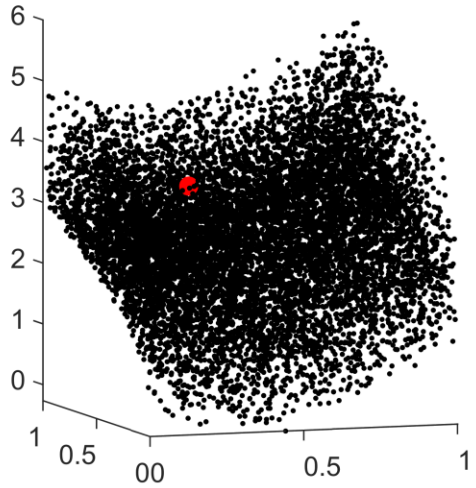
# Data visualizations are often not enough

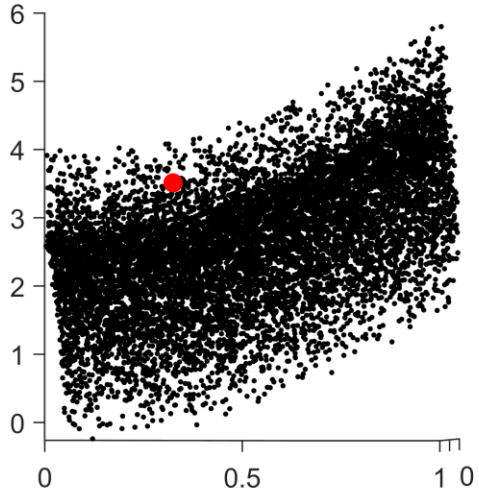


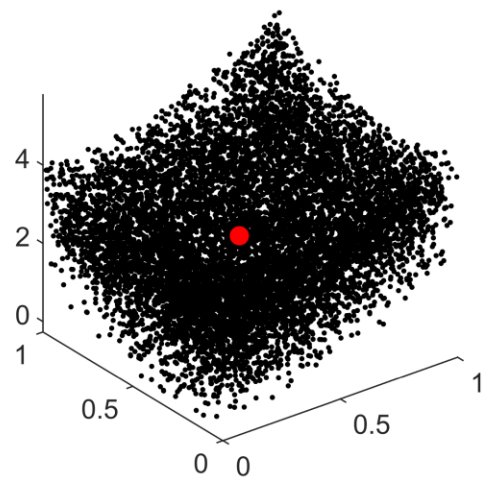



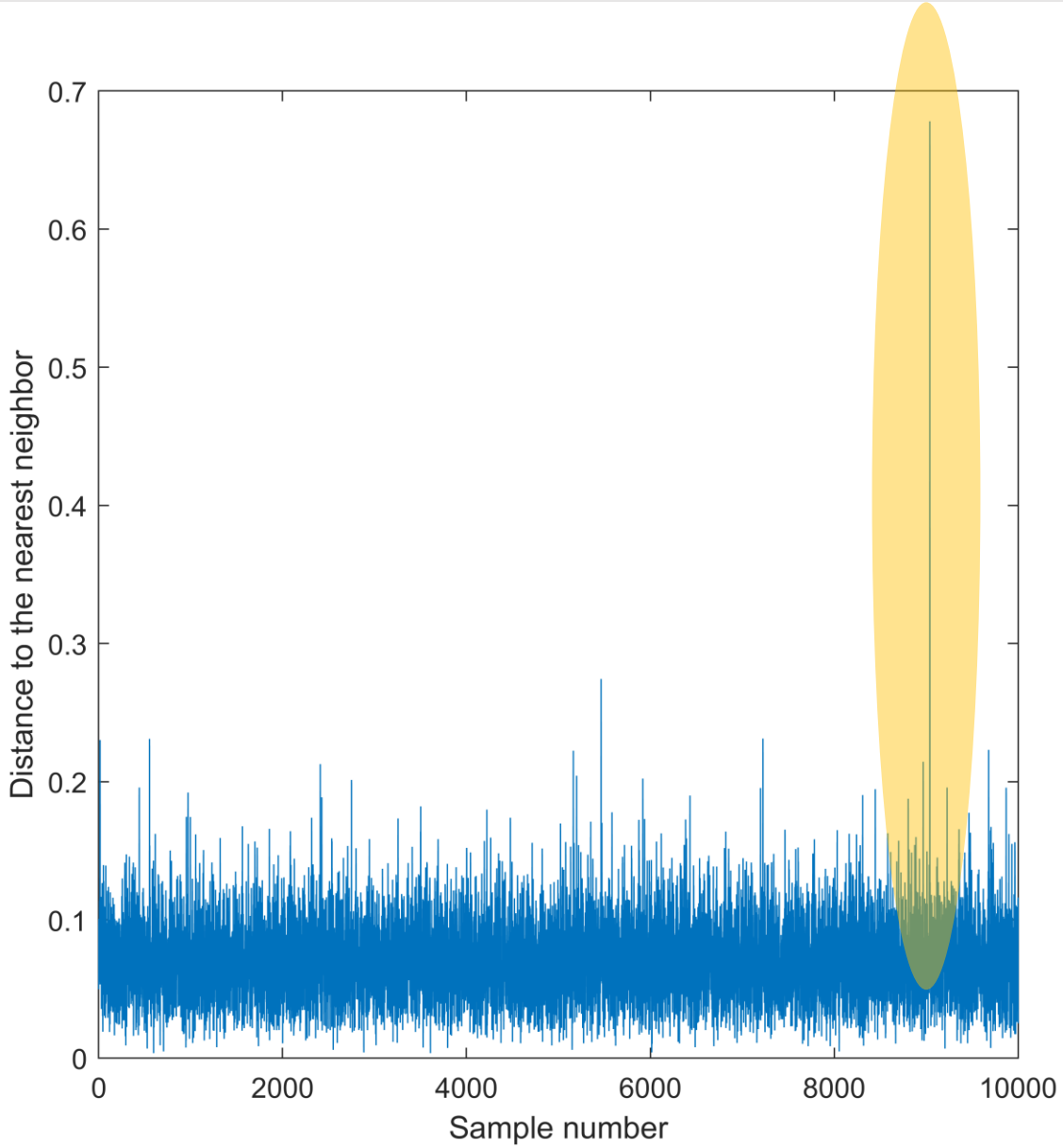
# Data visualizations are often not enough



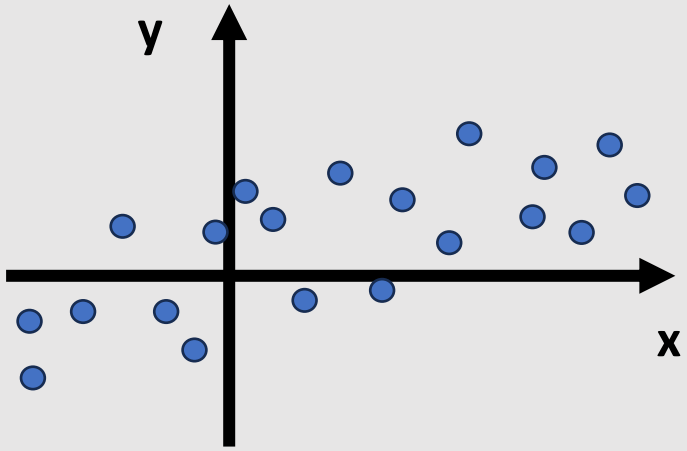


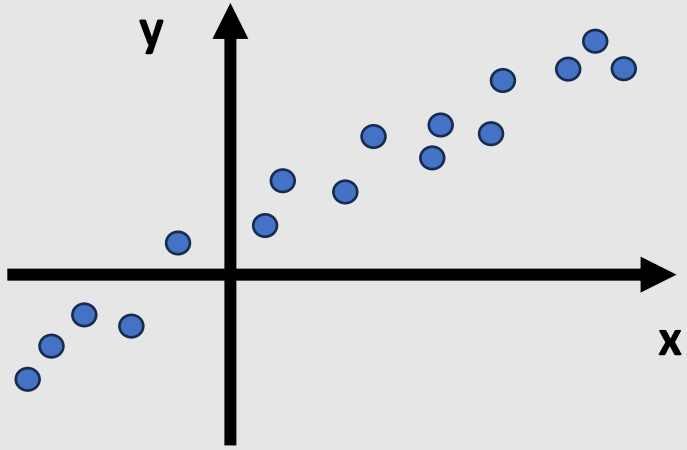






# Data visualizations are often not enough



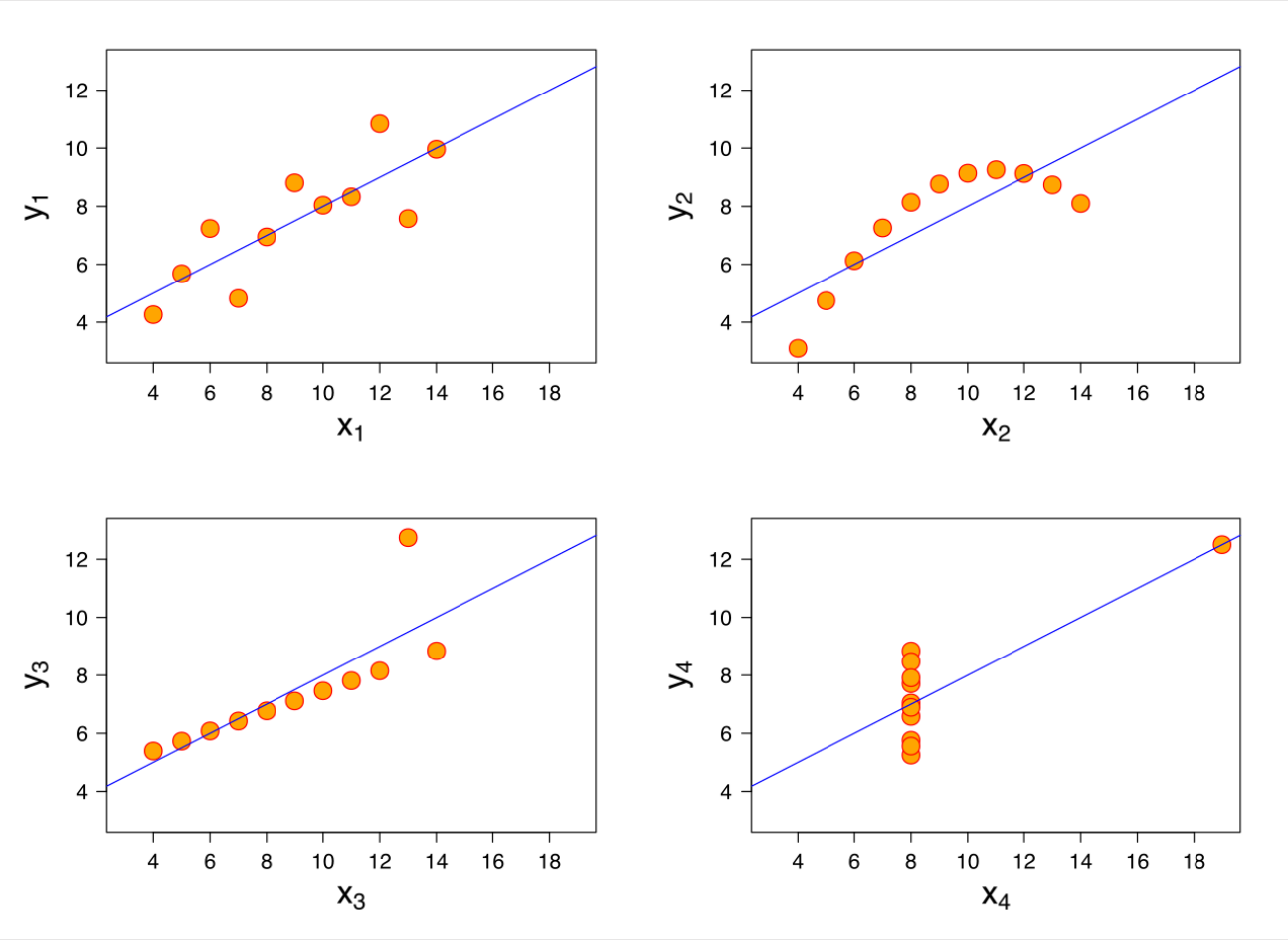

We can often look at two different datasets and roughly estimate the relations in data.

Is it true for high-dimensional datasets?

We actually need a better way of comparing datasets together…

Statistical properties come in mind here: mean values, variances, regression lines, etc.

# Statistical measures are often not enough




**Anscombe's quartet**

While this is a hand-crafted example, it shows the problem behind looking only at statistical data

**The same:**

Mean of x
Variance of x

**Almost the same:**

Mean of y
Variance of y
Correlation
Regression line
Regression error

# Statistical measures are often not enough

So how to actually look at datasets?

**Holistically!**



- Look at statistical measures
- Look at raw and processed data
- **Look at outcomes of various models**
- **Draw hypothesis about data and test them**
- Think and experiment!

# Things to remember:

1. What are the features of Python language?
2. What types of data are there? How can we approach categorical data?
3. Enumerate and explain different visualization techniques (axes descriptions, usage)
4. What are the most common data visualization errors?
5. Why 1D representations are often not enough to draw conclusions?
6. Why basic statistical estimates of dataset are often not enough to draw conclusions?