COM SCI M148, Lecture 2:
Data collection, cleaning, and bias

Prof. Nathan Szymanski
nszymanski@seas.ucla.edu
Department of Materials Science

UCLA

Samueli
School of Engineering

Recap: The data science pipeline

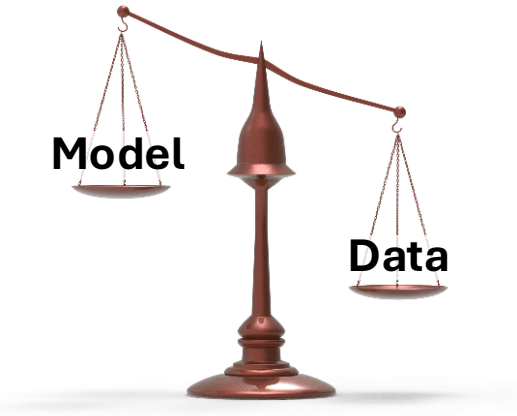
Ask an interesting question

Get the Data

Clean the Data

Model the Data

Interpret the Results



Today we are going to focus on **the data!**



We will learn how to avoid garbage data ☺

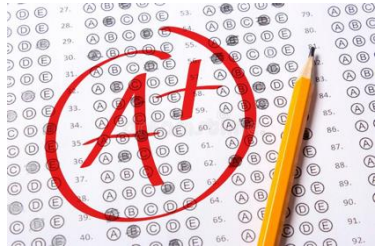
Step 1: Clearly define your variables

Question: How does house size affect the selling price of a home?



Variables:
- House size (square feet) → input variable
- Selling price ($) → output variable

Question: Does studying improve exam performance?



Variables:
- Hours studied (h) → input variable
- Exam grade (%) → output variable



Question: Under what conditions does my battery perform best?

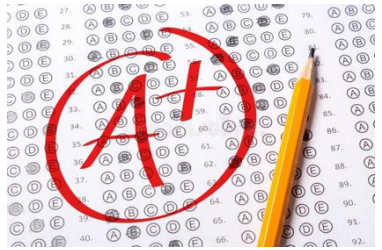
Variables:
- Temperature, humidity, charge/discharge rates → input variables
- Lifetime of the battery, energy delivered → output variables

Step 2: Decide *how to measure* those variables

Question: Does studying improve exam performance?

Variables:

- Hours studied (h) → input variable
- Exam grade (%) → output variable





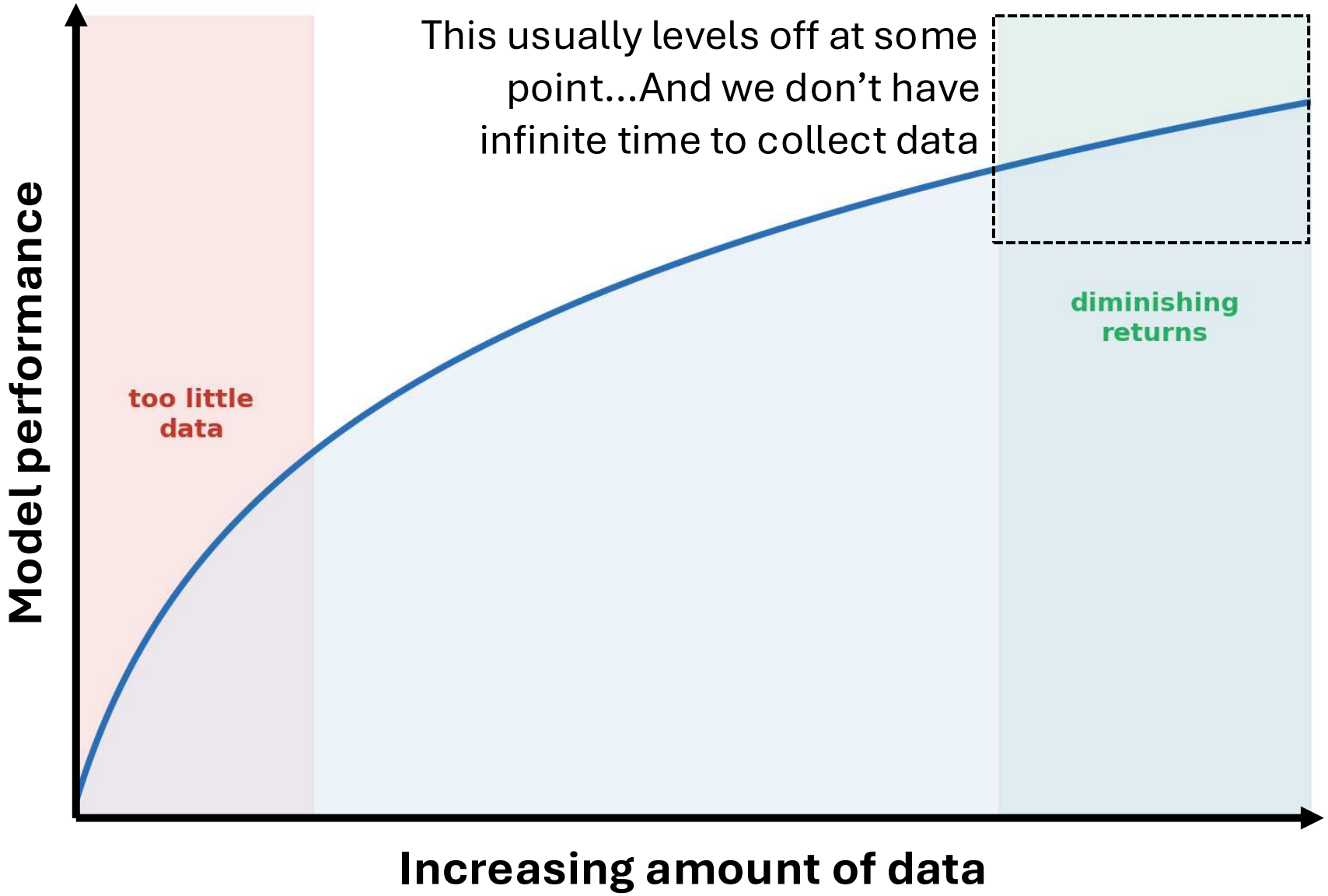
"I will do a survey on some of the class, asking how long they studied, and what grade they got."



How many students do you need to sample?

Which ones?

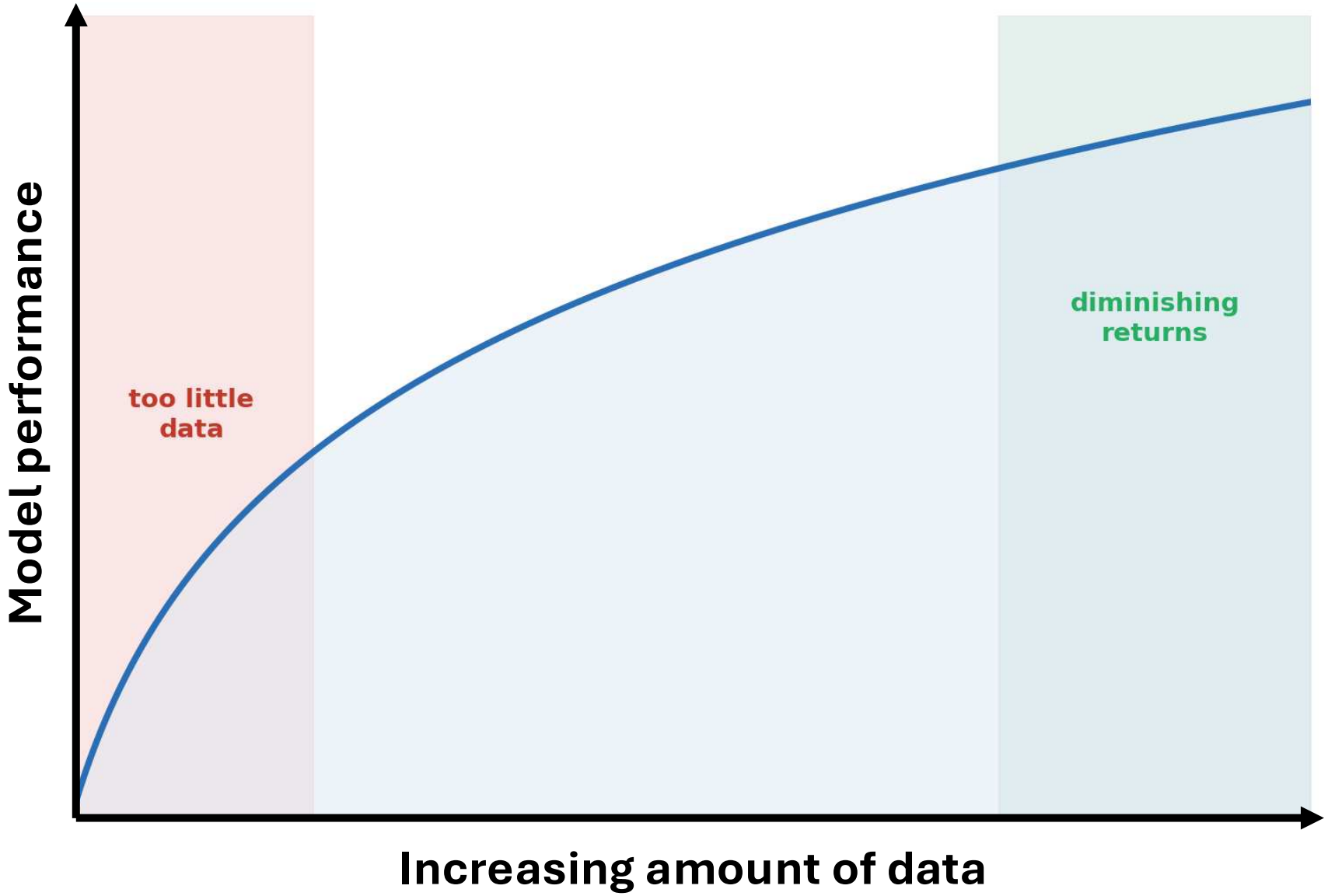
How much data is enough?



"How many samples is enough?" is a tricky question that requires a nuanced answer…

We'll answer this in more detail later when discussing confidence, but for for now, let's just cover some reasonable rules of thumb

How much data is enough?



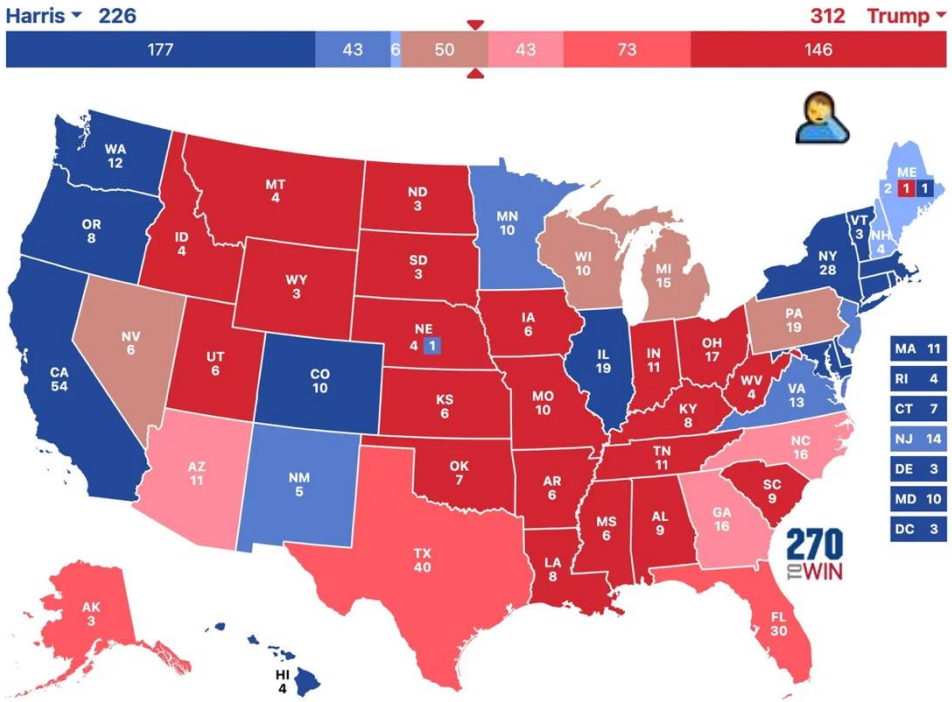
Variability:
More noisy data
→ more data needed

Effect size:
Small effects
(weak trends)
→ more data needed

Confidence:
To get higher certainty
→ more data needed

Complexity:
With more variables
→ more data needed

This also depends on total population size



For a very large population (the nation), polling a few thousand can be enough, **assuming this sample is representative of the population(!!!)**

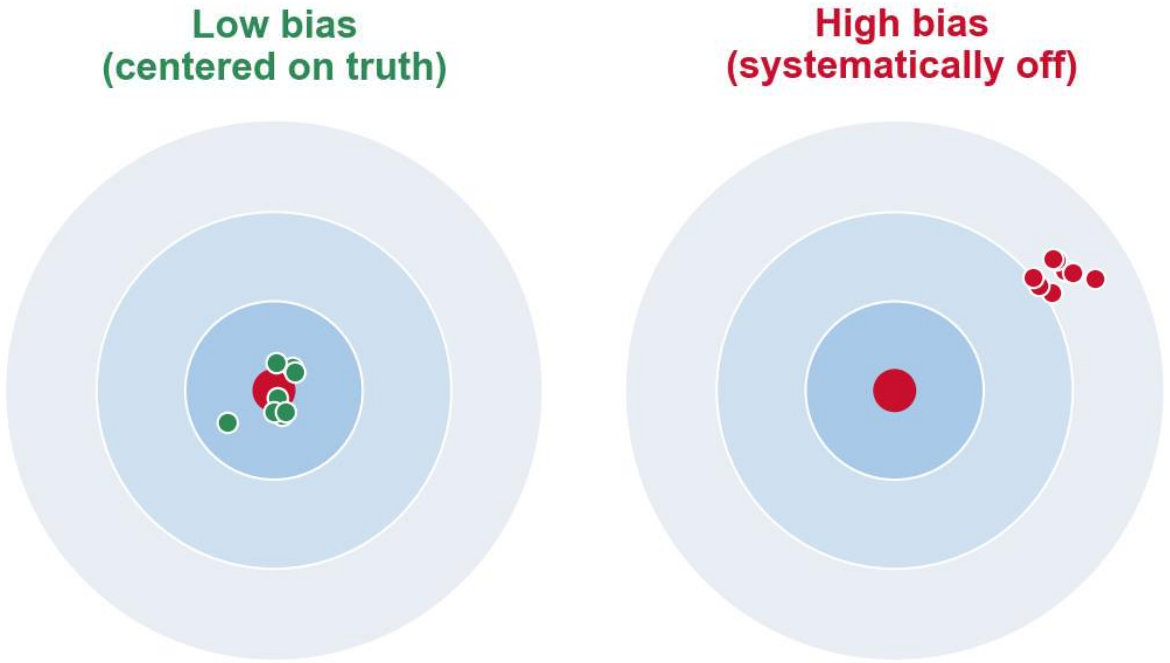
(a big assumption…remember *bias*…)

(more data may not fix bias)

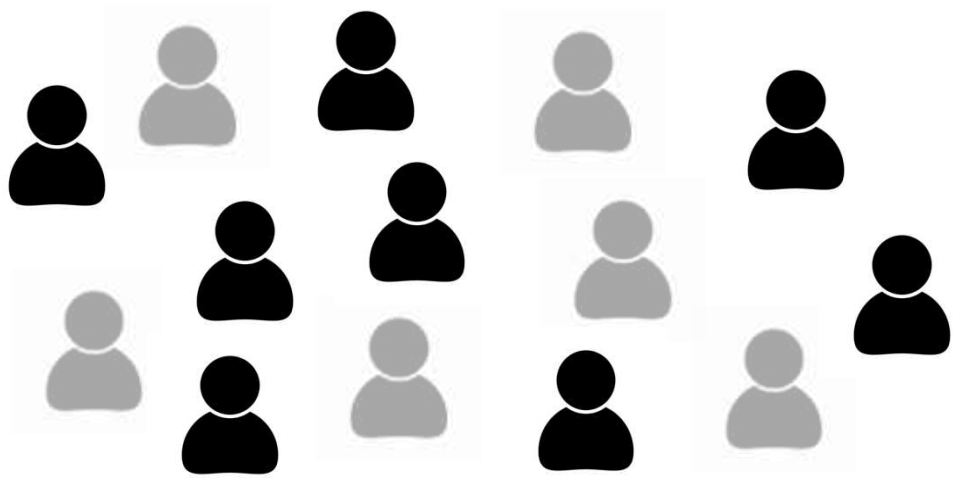


But for our smaller population, a **classroom of 100 people**, we need a *larger percentage* (say, **50-60 students**)

Which subset to sample?



More data does *not* fix this!



Our goal is to **choose a subset** whose average properties closely match those of the entire population. In other words: **minimize bias (systematic error).**



Which subset to sample?

1) Simple random sampling

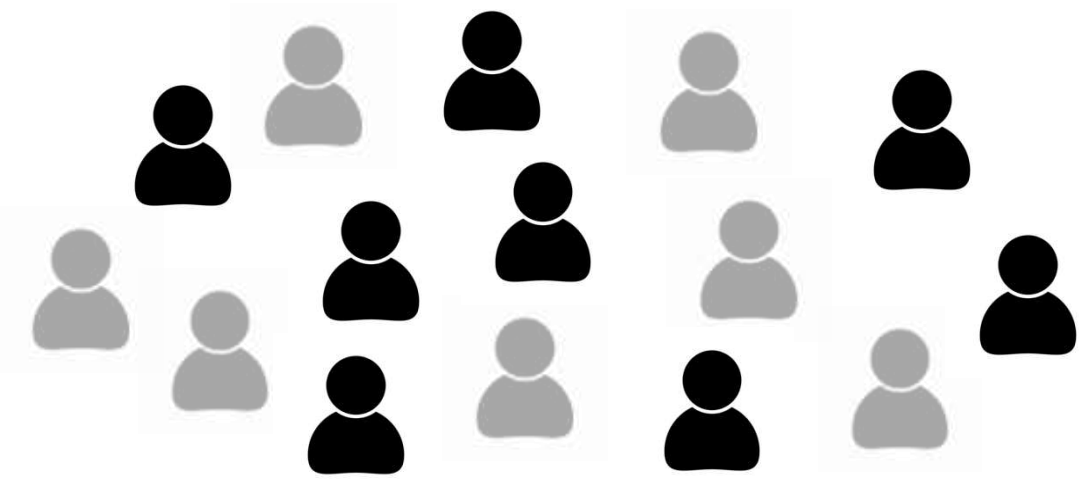
2) Stratified sampling

3) Cluster sampling

Probability sampling

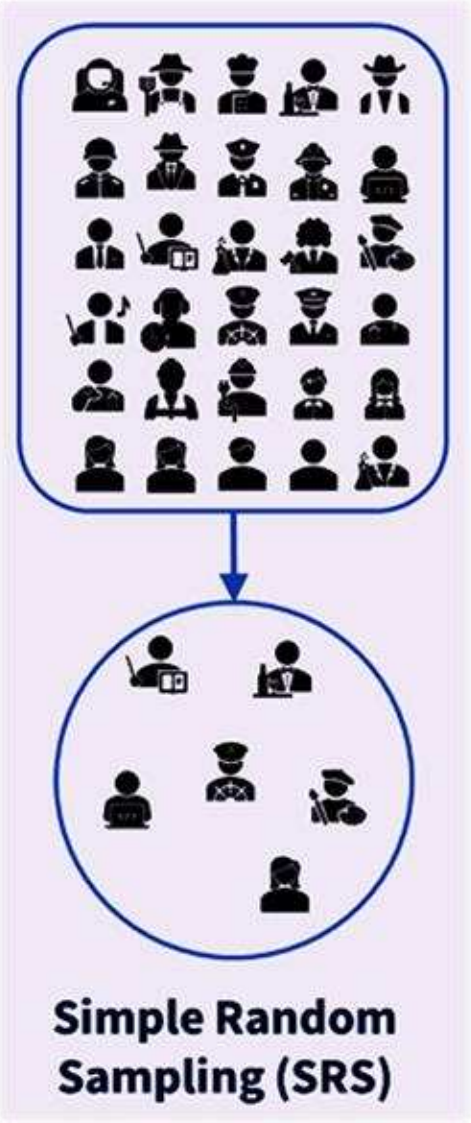
4) Systematic sampling

5) Convenience sampling





Simple random sampling



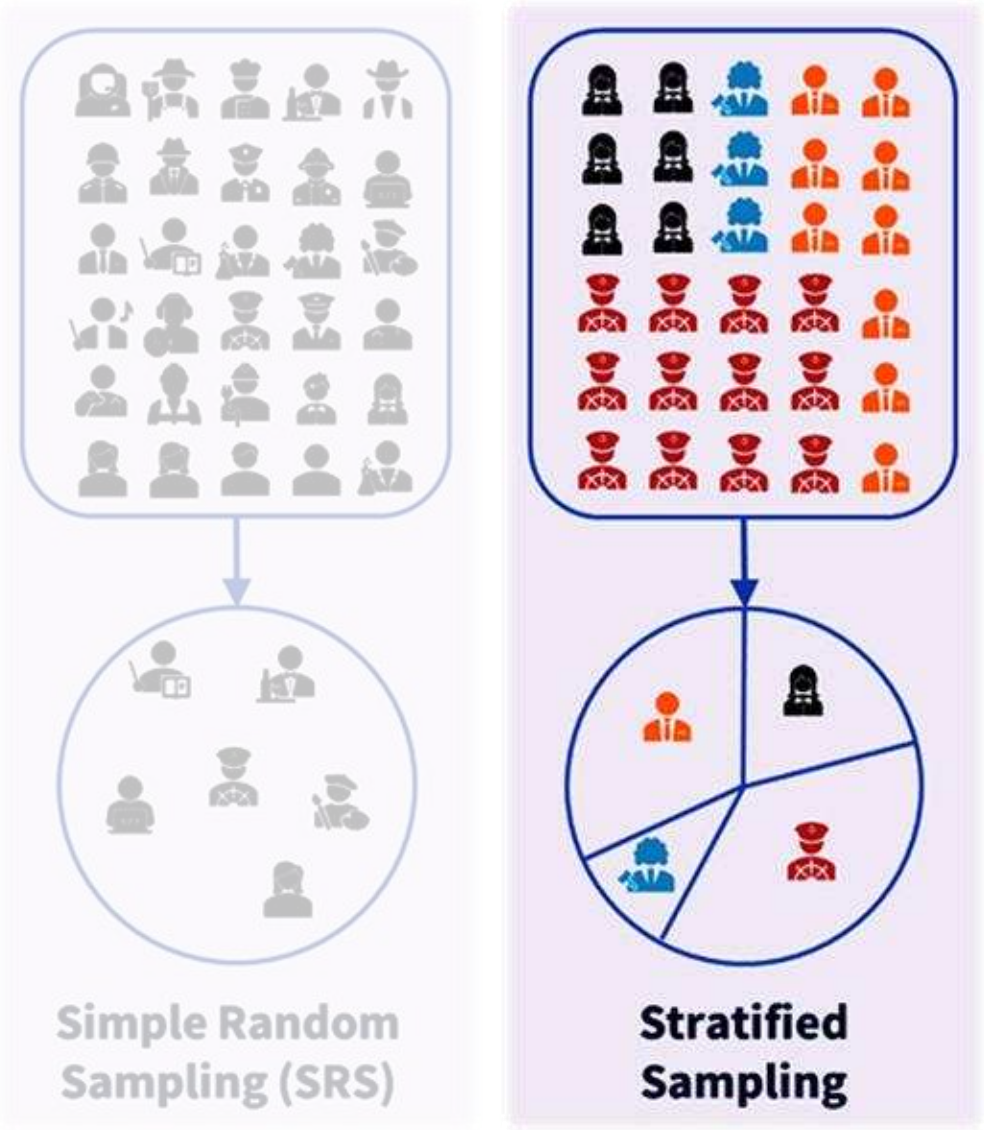
In our classroom example:
Randomly draw student names from a hat



Simple random sampling is the "gold standard" because every member of the population has an equal chance of being selected ***(if sampling is truly random)***

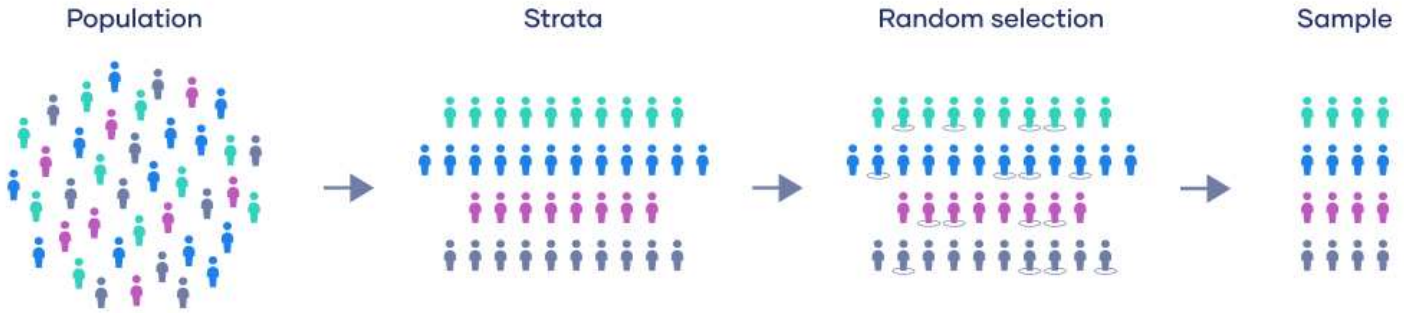
But in larger case studies like national case studies, it can be difficult to do in an unbiased way (some populations are easier to reach than others…)

Stratified sampling



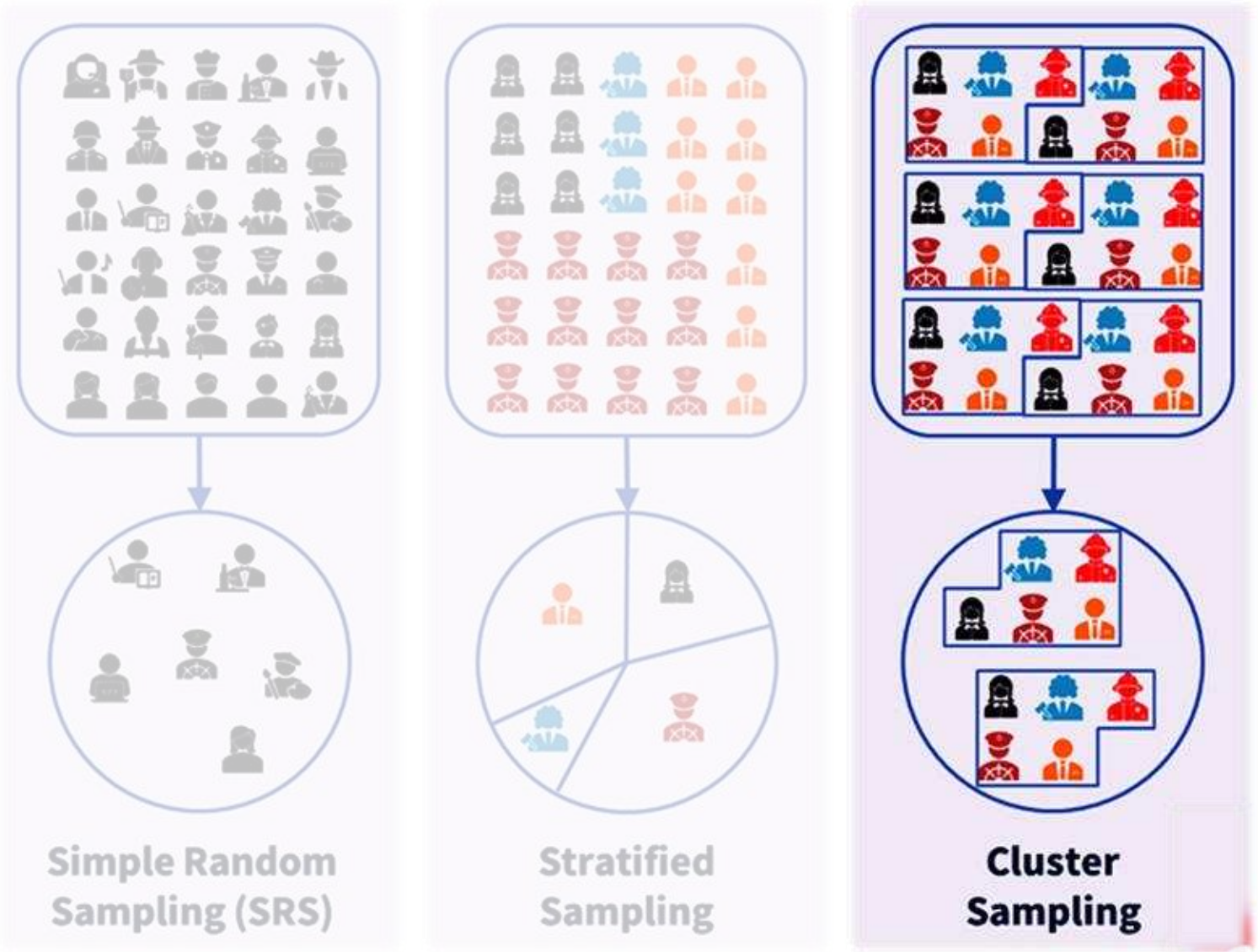
In our classroom example:
Divide students into logical groups (freshman, sophomores, juniors, seniors)

Then randomly sample from each group

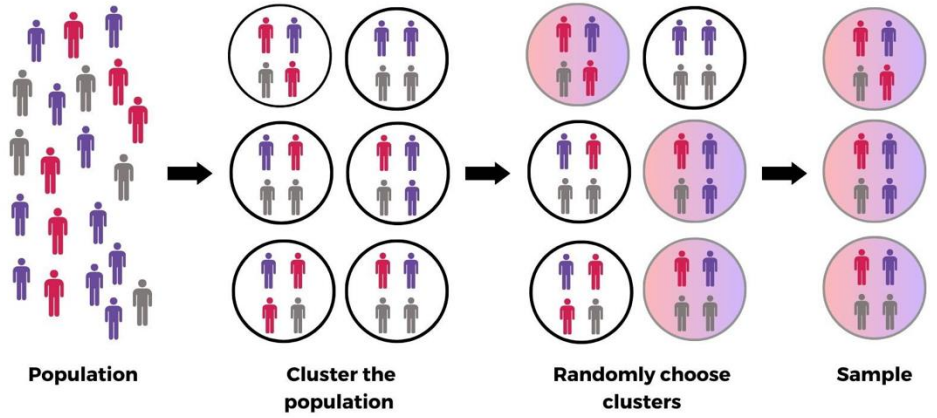


This is useful because it **guarantees all subgroups are represented** in the sample

Cluster sampling



In our classroom example: Randomly sample a subset of naturally occurring groups or "clusters" which each are representative of the overall population. For example: **Sample *every* student in *just one* discussion section**



Which subset to sample?

1) Simple random sampling

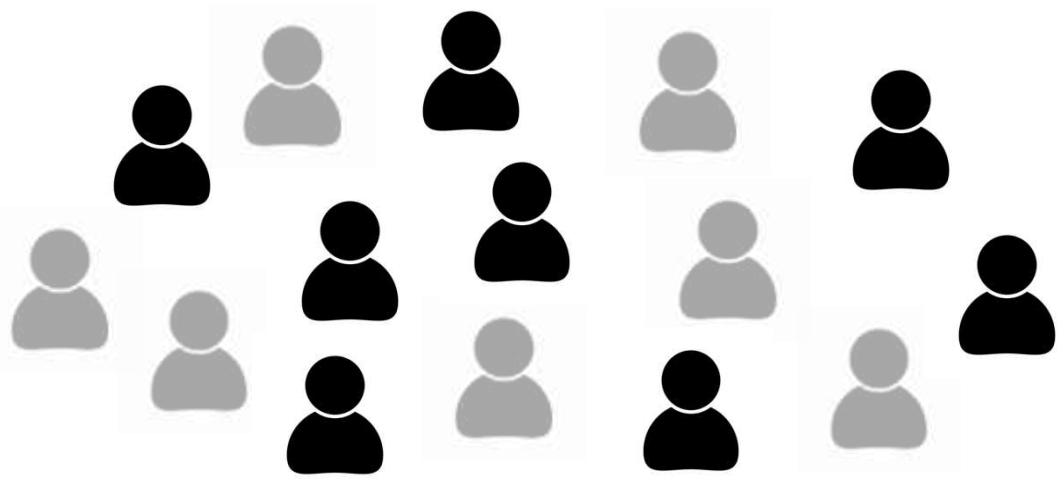
2) Stratified sampling

3) Cluster sampling

4) Systematic sampling

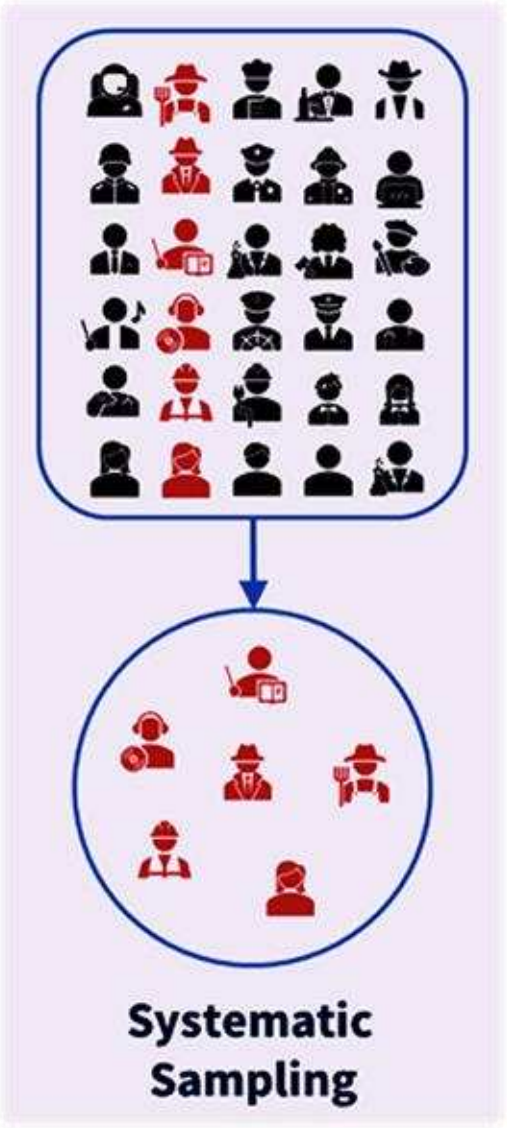
5) Convenience sampling

Nonprobability sampling

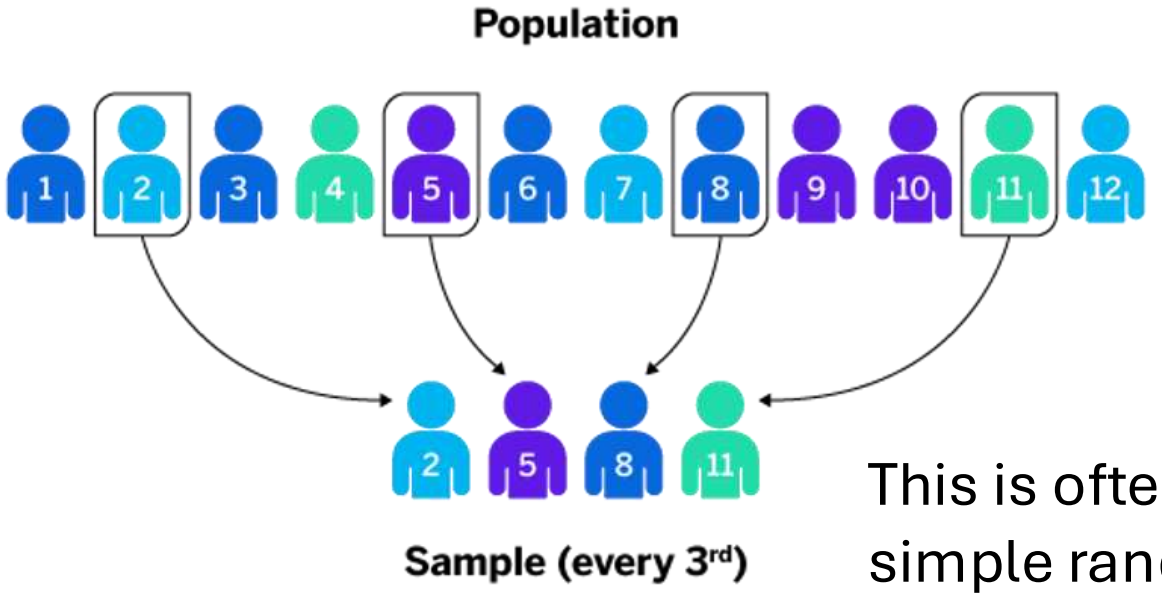




Systematic sampling

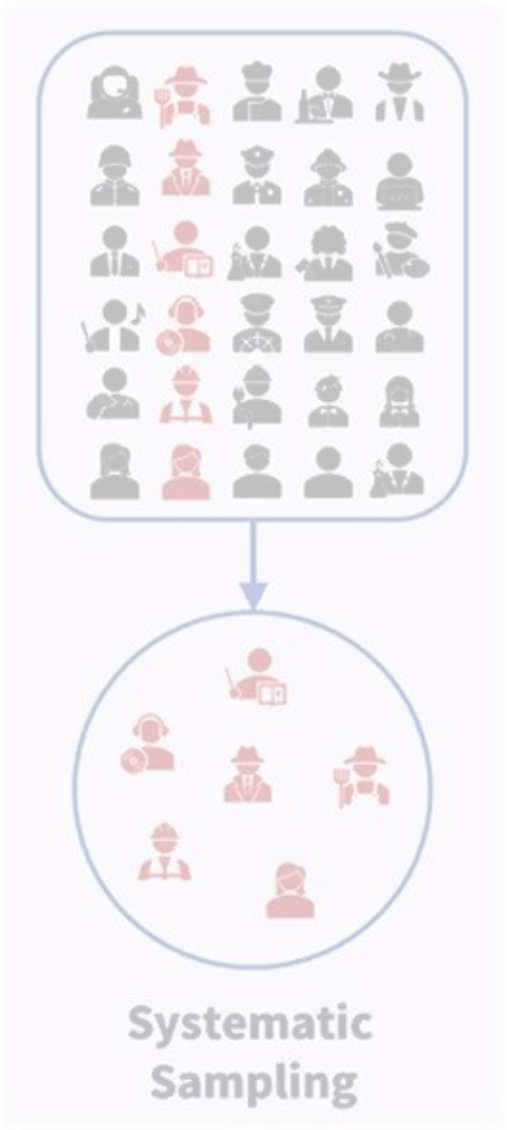


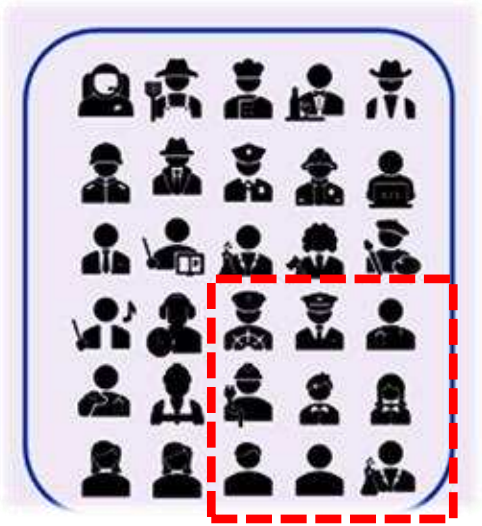
In our classroom example:
Go down each row in the classroom and **select every 3rd student**



This is often more practical than simple random sampling, but it can introduce bias if the ordering of students is not random.

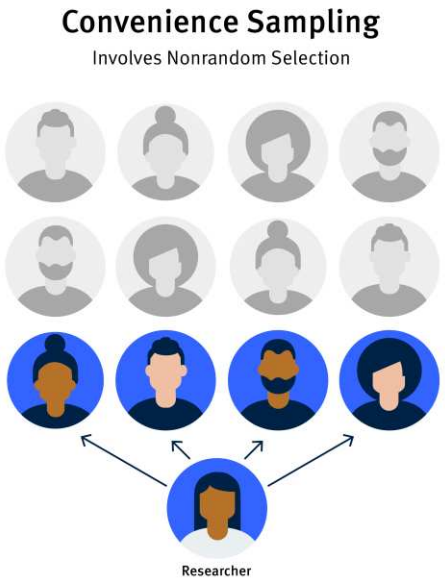
Convenience sampling





In our classroom example:
Sample all students from the **first three rows**

This is often the worst method, *not* being representative of the overall population





But as the name implies…It's convenient!

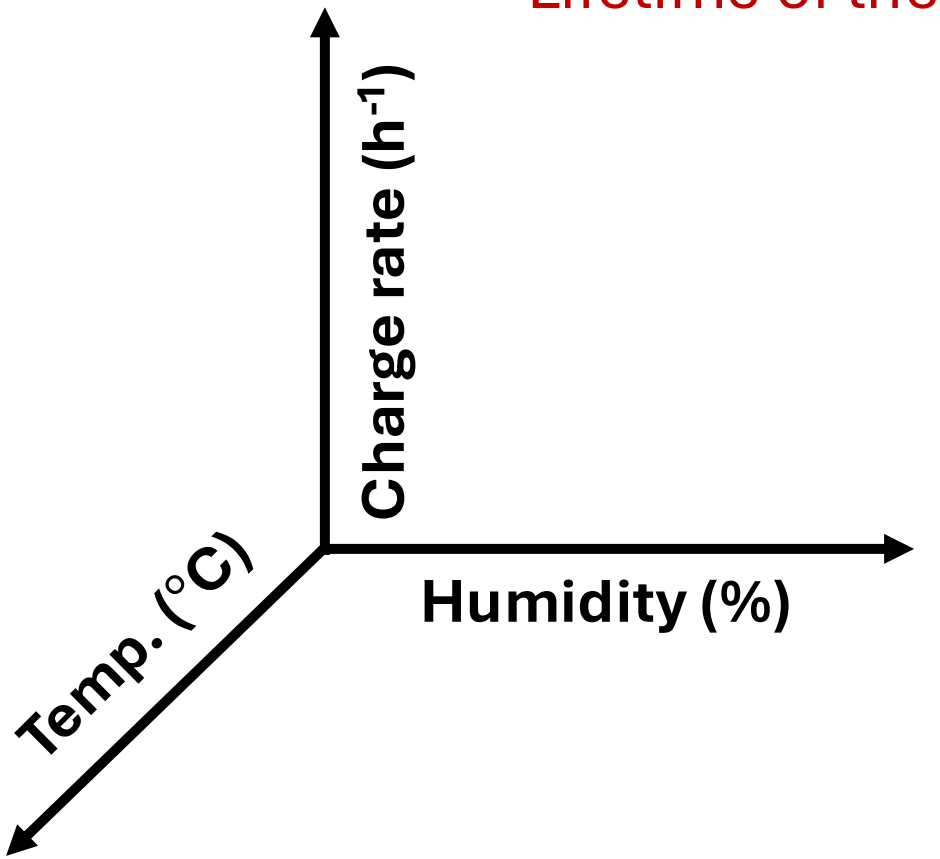
Similar methods apply to *continuous* spaces

Question: Under what conditions does my battery perform best?



Variables:

- Temperature, humidity, charge/discharge rates → input variables
- Lifetime of the battery, energy delivered → output variables

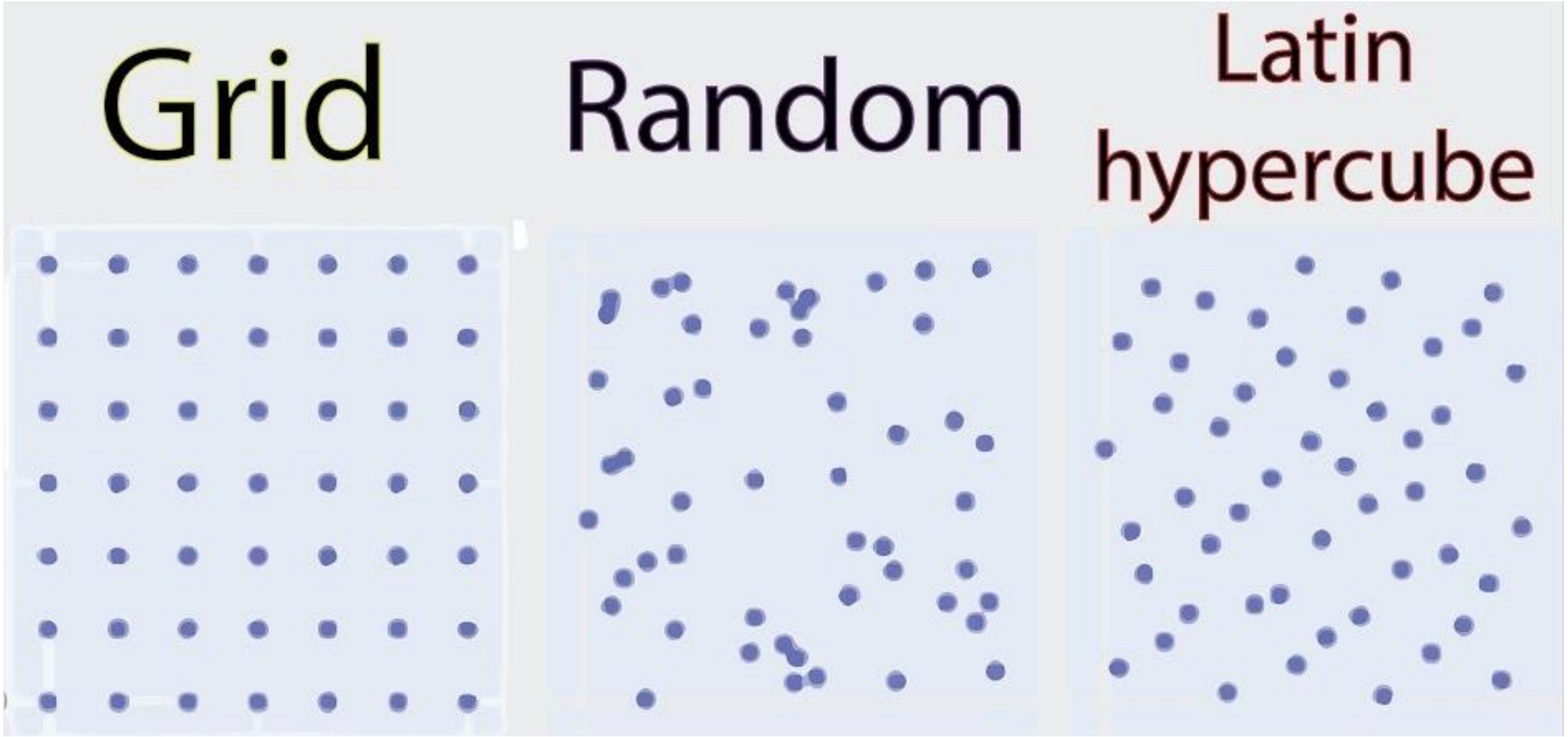


"I will run several experiments **under different conditions** and monitor battery health."

What conditions to test?

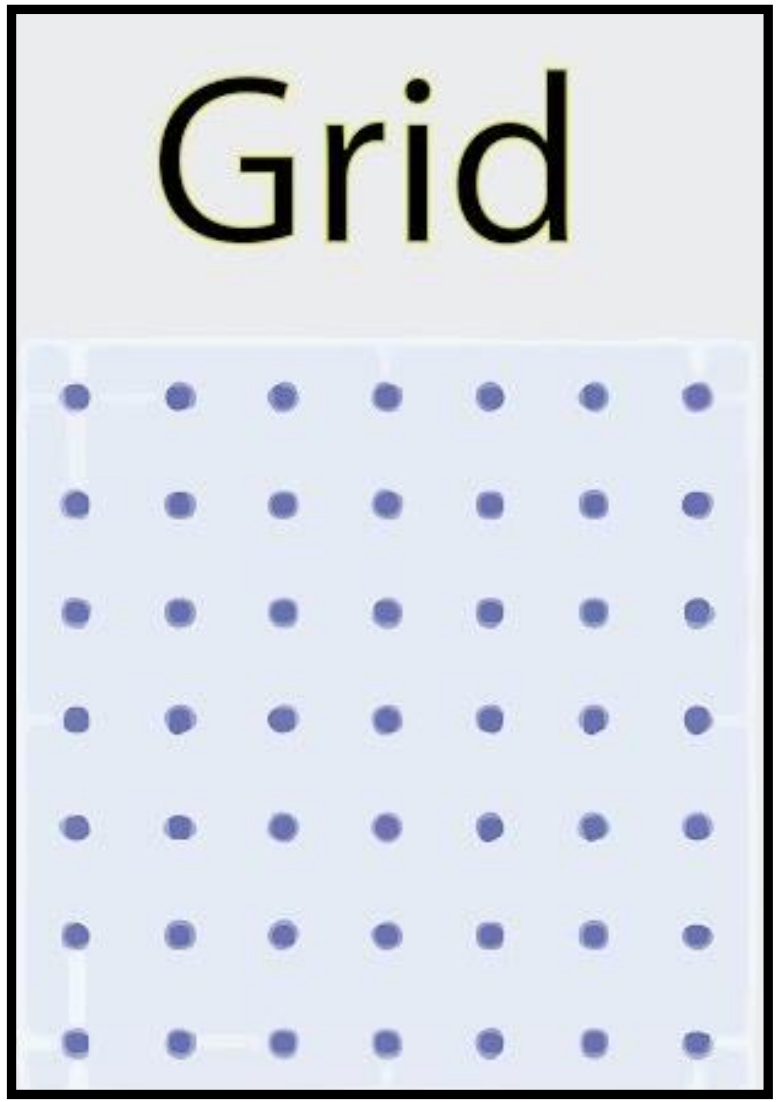
This is sometimes called "Design of experiments"

Sampling methods for continuous spaces



^Taken from a thumbnail of a YouTube video, though I forget which one…

Sampling methods for continuous spaces



This is analogous to **systematic sampling**

We uniformly sample across each variable

EX: 20 °C, 40 °C, 60 °C, …

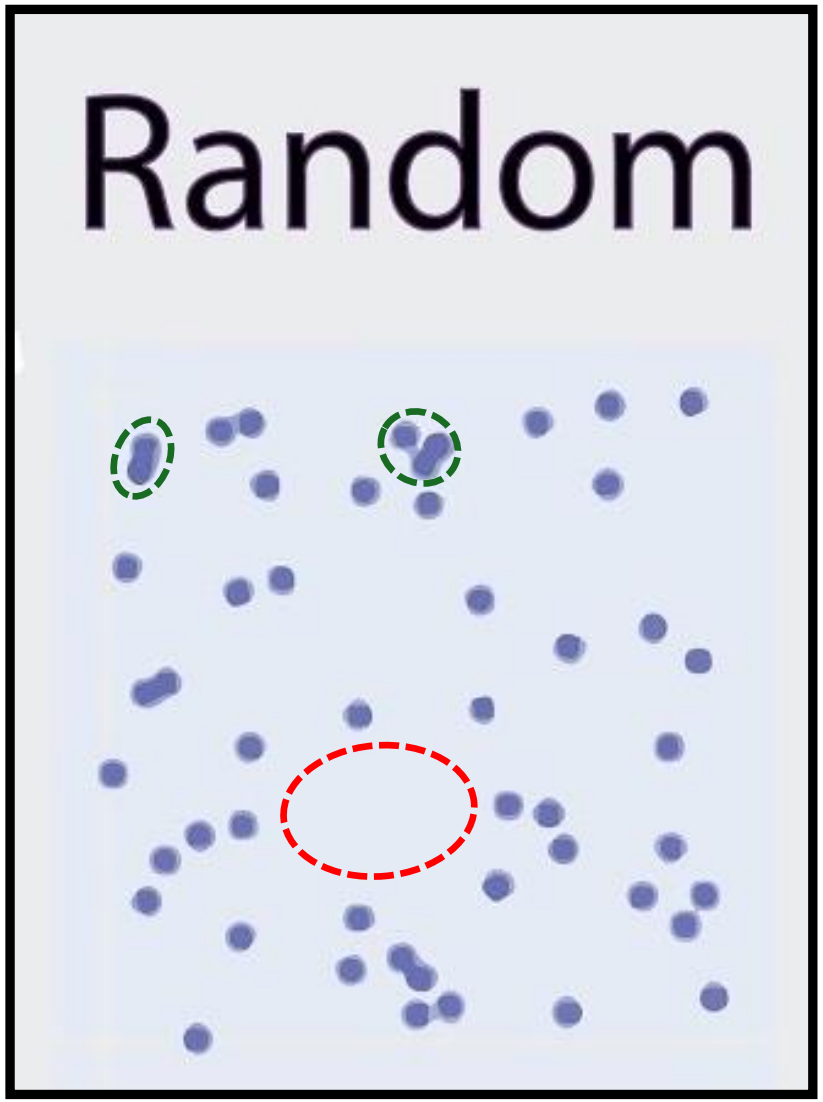
This ensures the **design space is well sampled**

But it can **become expensive** for very large spaces involving many variables

Sampling methods for continuous spaces

This is usually the method of choice for systems involving many variables

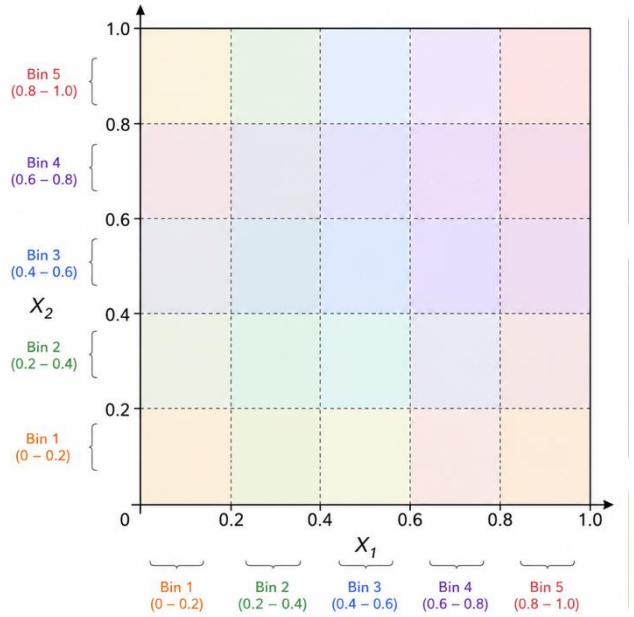
However, it can lead to "gaps" in the space, and overlapping points



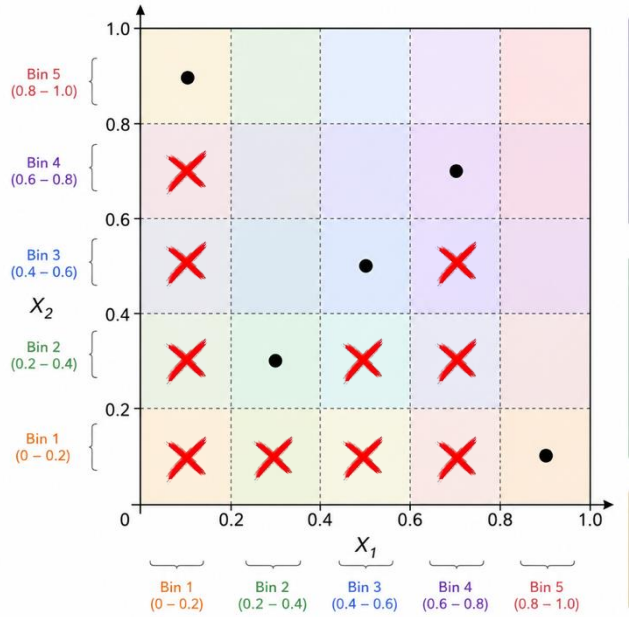
Sampling methods for continuous spaces

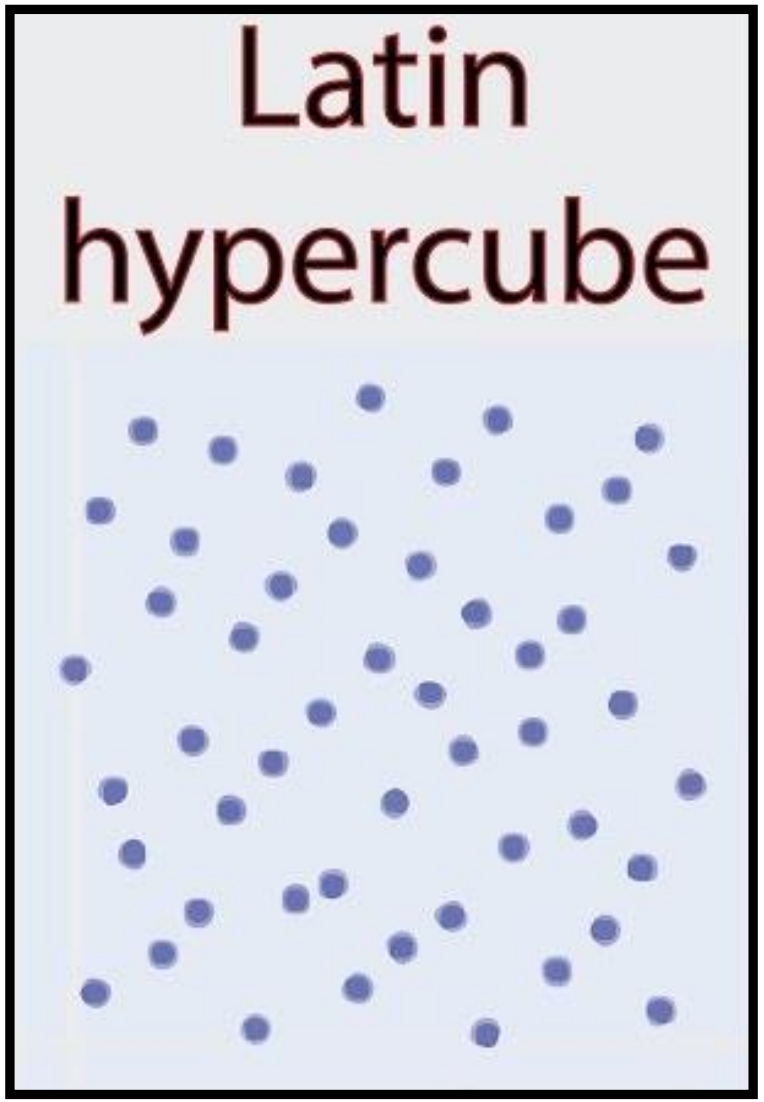
This provides a **nice balance** "in between" random and uniform sampling. Notice the points *look* random but are more evenly spread out.

1) Divide each variable into bins, **forming a grid**



2) Place ***exactly one point*** in each row and column





Questions?

Cleaning & Curating Data

Real-world data is imperfect and messy!

Missing values Wrong types/values Duplicates

Inconsistent labels Bad dates

name	age	income	city	signup
Ana	24	55000	LA	2023-01-04
BEN	-1	62k	los angeles	01/05/2023
ana	24	55000	LA	2023-01-04
Carlos	thirty		NYC	2023/02/30
Dee	41	NaN	New York	

Real-world data is imperfect and messy!

Missing values Wrong types/values Duplicates

Inconsistent labels Bad dates

name	age	income	city	signup
Ana	24	55000	LA	2023-01-04
BEN	-1	62k	los angeles	01/05/2023
ana	24	55000	LA	2023-01-04
Carlos	thirty		NYC	2023/02/30
Dee	41	NaN	New York	

Real-world data is imperfect and messy!

Missing values | Wrong types/values | Duplicates | Inconsistent labels | Bad dates

name	age	income	city	signup
Ana	24	55000	LA	2023-01-04
BEN	-1	62k	los angeles	01/05/2023
ana	24	55000	LA	2023-01-04
Carlos	thirty		NYC	2023/02/30
Dee	41	NaN	New York	

Real-world data is imperfect and messy!

Missing values Wrong types/values Duplicates

Inconsistent labels Bad dates

name	age	income	city	signup
Ana	24	55000	LA	2023-01-04
BEN	-1	62k	los angeles	01/05/2023
ana	24	55000	LA	2023-01-04
Carlos	thirty		NYC	2023/02/30
Dee	41	NaN	New York	

Real-world data is imperfect and messy!

Missing values Wrong types/values Duplicates

Inconsistent labels Bad dates

name	age	income	city	signup
Ana	24	55000	LA	2023-01-04
BEN	-1	62k	los angeles	01/05/2023
ana	24	55000	LA	2023-01-04
Carlos	thirty		NYC	2023/02/30
Dee	41	NaN	New York	

Real-world data is imperfect and messy!

Missing values Wrong types/values Duplicates

Inconsistent labels Bad dates

name	age	income	city	signup
Ana	24	55000	LA	2023-01-04
BEN	-1	62k	los angeles	01/05/2023
ana	24	55000	LA	2023-01-04
Carlos	thirty		NYC	2023/02/30
Dee	41	NaN	New York	

inconsistent

Real-world data is imperfect and messy!

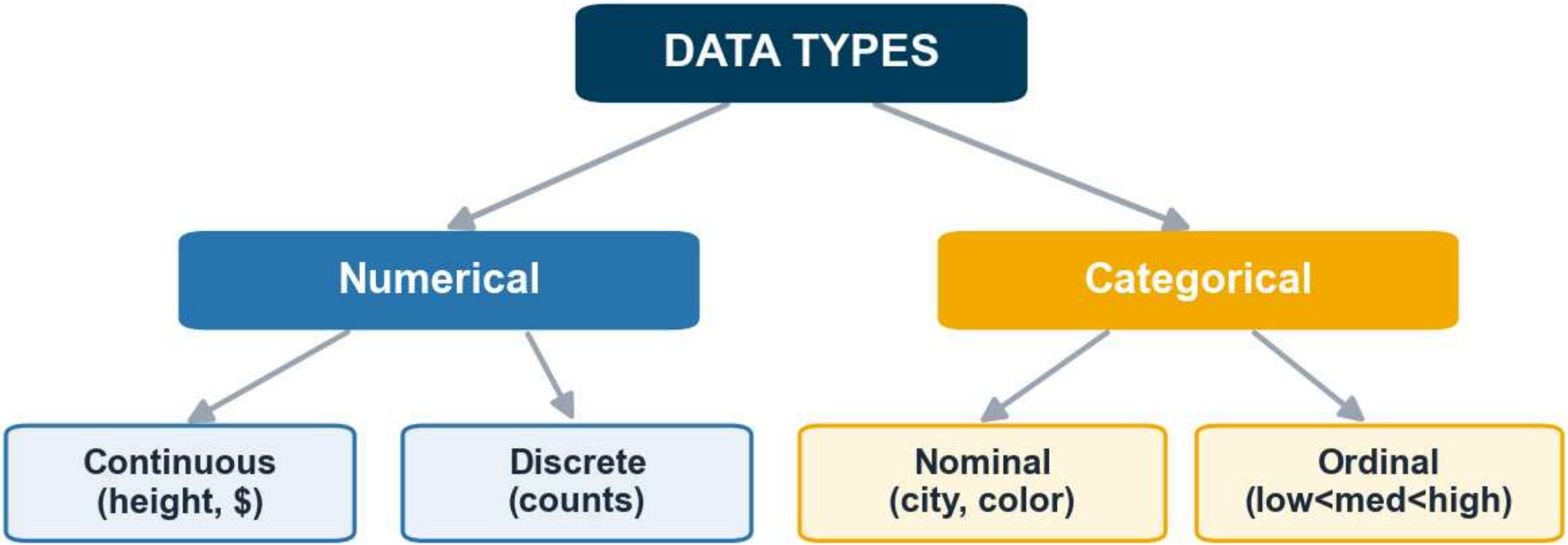
Missing values | Wrong types/values | Duplicates | Inconsistent labels | Bad dates

name	age	income	city	signup
Ana	24	55000	LA	2023-01-04
BEN	-1	62k	los angeles	01/05/2023
ana	24	55000	LA	2023-01-04
Carlos	thirty		NYC	2023/02/30
Dee	41	NaN	New York	

As data scientists, we should **“tidy this up”** *before* passing it to a model

How exactly we do this will depend on the **data type**

Know your data types



“Is a number”

“Is *not* a number”

(slightly oversimplified, but this is usually an okay distinction)

Inconsistency in categorical data

name	age	income	city	signup
Ana	24	55000	LA	2023-01-04
BEN	-1	62k	los angeles	01/05/2023
ana	24	55000	LA	2023-01-04
Carlos	thirty		NYC	2023/02/30
Dee	41	NaN	New York	

Duplicates

Inconsistent labels

Categorical variables should be **standardized** in their naming

CA

Calif.

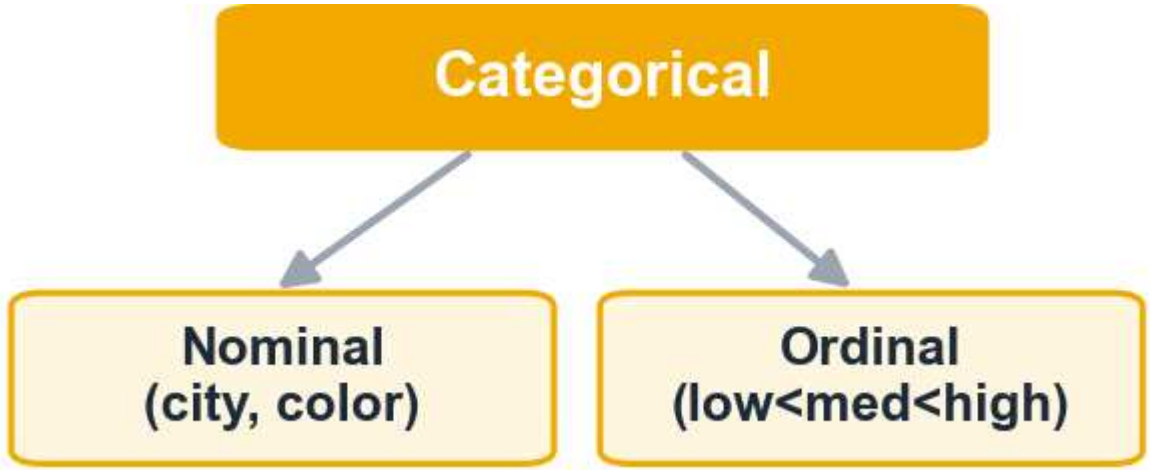
california

C.A.

Cali

Choose one and stick with it!

standardize → "CA"



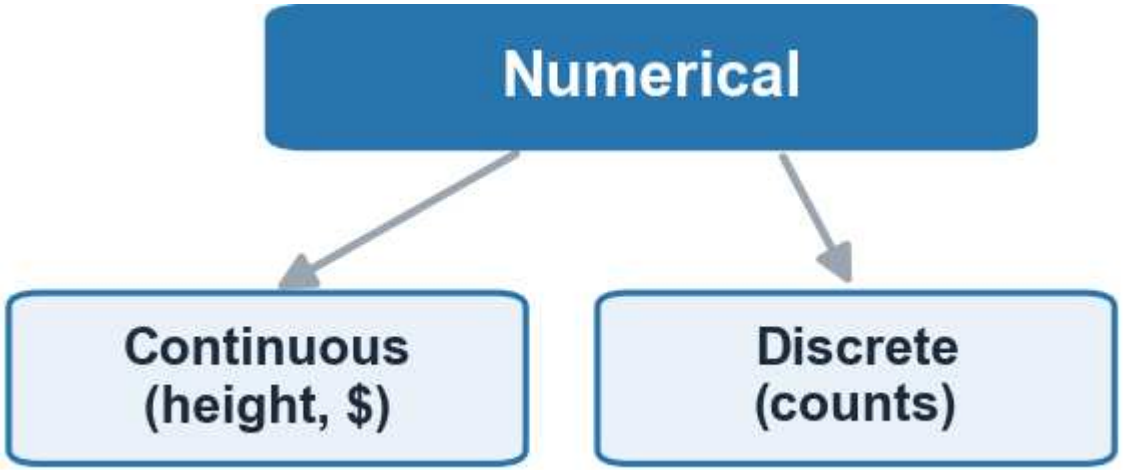
Sometimes requires domain expertise (*e.g.*, scientific terms)

Missing or strange numerical data

Missing values

Wrong types/values

name	age	income	city	signup
Ana	24	55000	LA	2023-01-04
BEN	-1	62k	los angeles	01/05/2023
ana	24	55000	LA	2023-01-04
Carlos	thirty		NYC	2023/02/30
Dee	41	NaN	New York	



First, let's deal with **missing values**

Should we just **drop** the rows associated with Carlo and Dee?

Maybe…But then we lose other (maybe *valuable*) data (age, city) and we may introduce bias (what if it's missing for a reason?)

Why is data missing?

MCAR

Missing Completely at Random

Gaps are pure luck.
No pattern.

Safe to drop / impute.

MAR

Missing at Random

Missingness depends on OTHER observed columns.

Impute using them.

MNAR

Missing Not at Random

Missingness depends on the MISSING value itself.

Dangerous — can bias!

Why is data missing: MCAR

MCAR

Missing Completely at Random

Gaps are pure luck.
No pattern.

Safe to drop / impute.

Example:

You survey 100 students about study hours.
A coffee spill destroys 10 survey forms at random.



If you simply **remove the incomplete records**, your estimates **remain unbiased** ✓ (though you lose statistical power).

Why is data missing: MAR

Example:

You survey 100 students about study hours. **Male students are less likely to report exam grade.**

So missingness depends on gender, **which you can observe** and therefore statistically account for the missing data.



But the missingness does *not* depend on the missing value itself!

Imputation

Imputation methods to replace missing data

Gender	Hours studied	Exam grade
Male	3.5	88%
Male	2	**84.6%**
Female	4	93%
Male	0.5	72%
Female	1.5	76%
Female	2	85%
Female	0.5	61%
Male	4.5	94%

Imputation fills in the gaps in an **informed way**

Mean imputation does this by averaging over known values…

Average over gender, only if you suspect it plays a role:

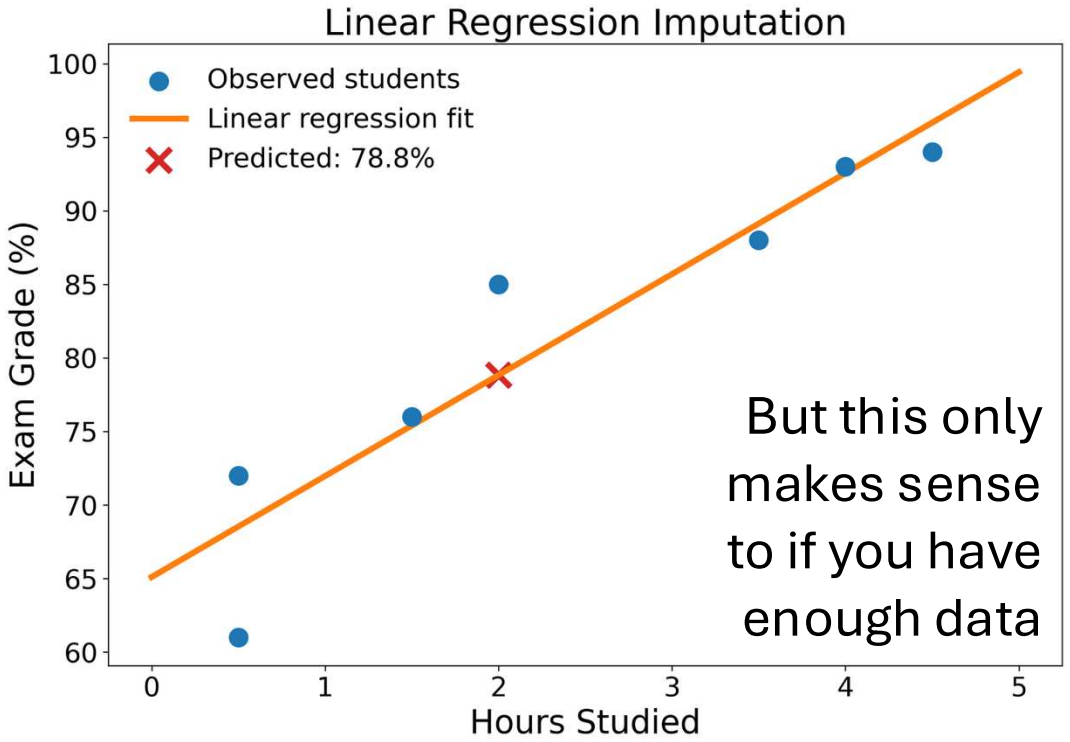
$$\text{Gap} = (88\% + 72\% + 94\%)/3$$

Average over everyone, or remove the row, if you do not expect gender to have an effect

Imputation methods to replace missing data

Gender	Hours studied	Exam grade
Male	3.5	88%
Male	2	76.4%
Female	4	93%
Male	0.5	72%
Female	1.5	76%
Female	2	85%
Female	0.5	61%
Male	4.5	94%

Regression imputation does this by fitting a model over known values and predicting the missing one



But this only makes sense to if you have enough data

Why is data missing: MNAR

Example:

You survey 100 students about study hours. **Students with low exam grades are less likely to report than those with high grades.**

In this case, there is no easy solution… **You *cannot* impute or remove the data without introducing bias!** The only viable solution is to come up with a better data collection method!

MNAR

Missing Not at Random

Missingness depends on the MISSING value itself.

Dangerous — can bias!

Why you can’t impute MNAR cases

Gender	Hours studied	Exam grade
Male	3.5	88%
Male	2	
Female	4	93%
Male	0.5	72%
Female	1.5	76%
Female	2	85%
Female	0.5	61%
Male	4.5	94%

Suppose this guy didn’t report because he got a 20% ☹

Then mean imputation would yield a grade that is very incorrect!

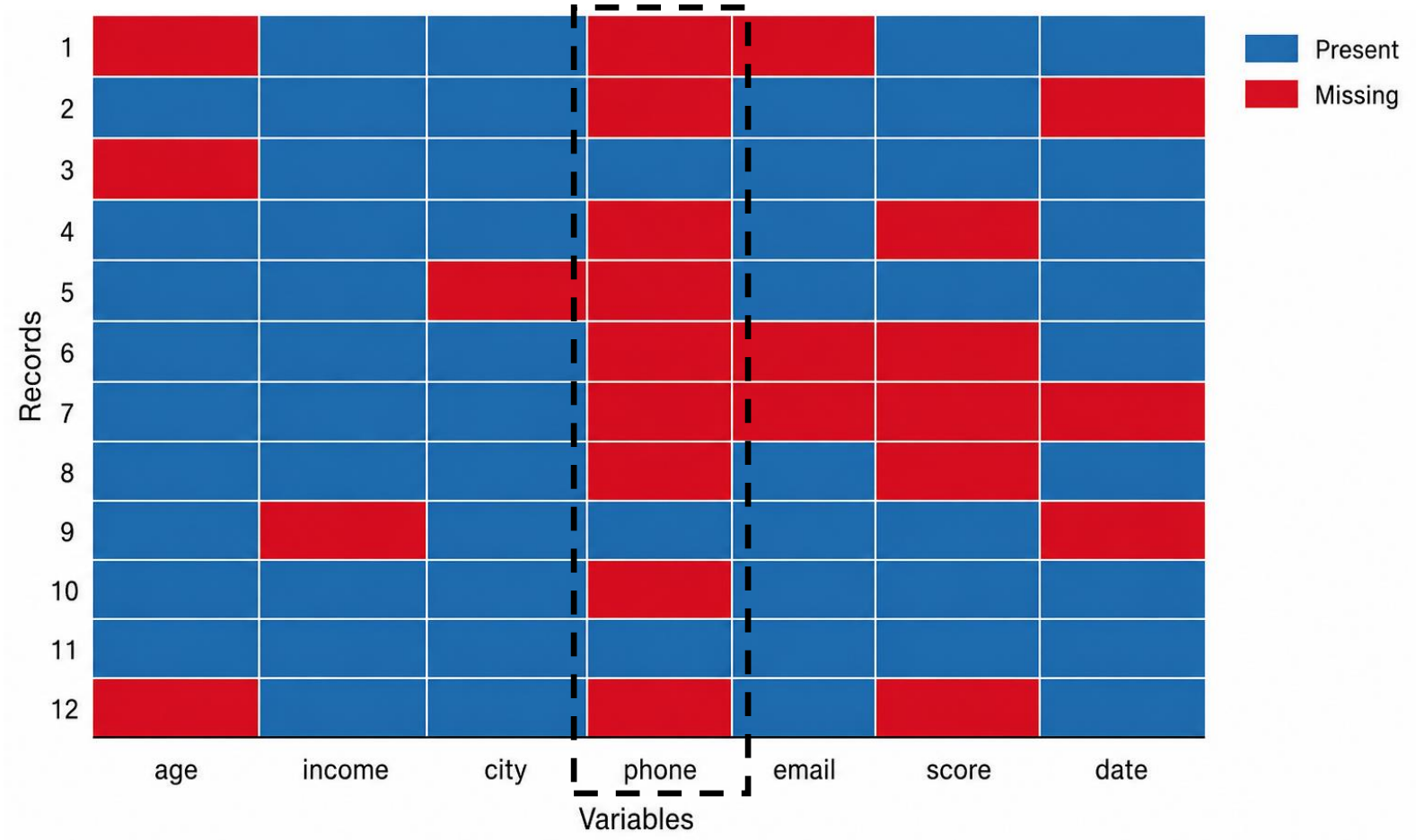
Visualize the missingness

In some cases, it can make sense to **remove a feature** instead of a sample (remove the columns, not the rows)

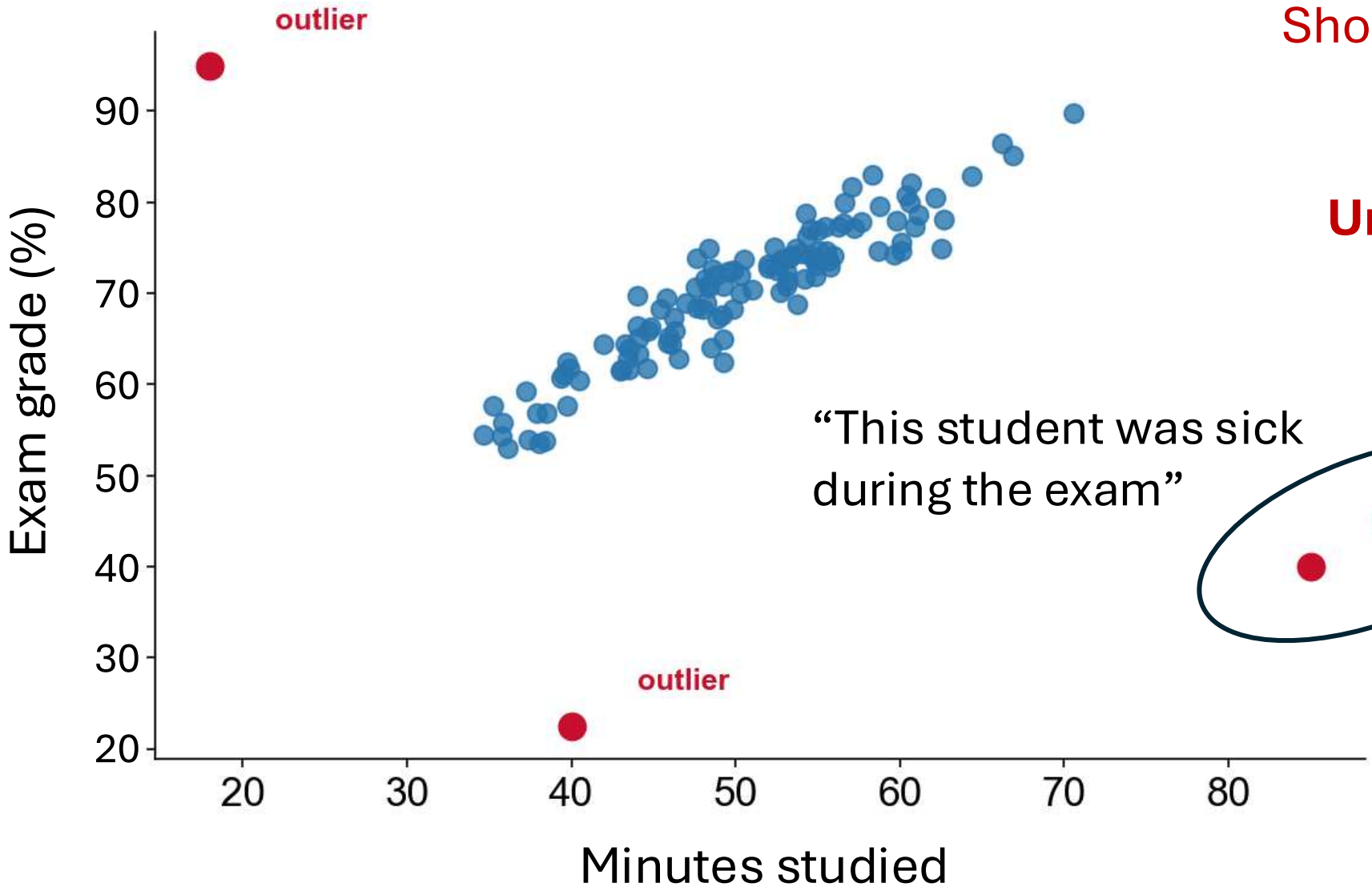
Looking at this "missingness matrix" and notice that many phone numbers are missing… Probably best to remove it

You should always visualize your data *before* modeling!

Another reason:
Outlier detection

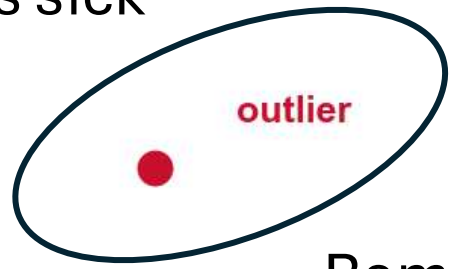


Outliers: points far from the rest



Should you remove these outliers? Probably not… **Unless you have good reason!**

"This student was sick during the exam"



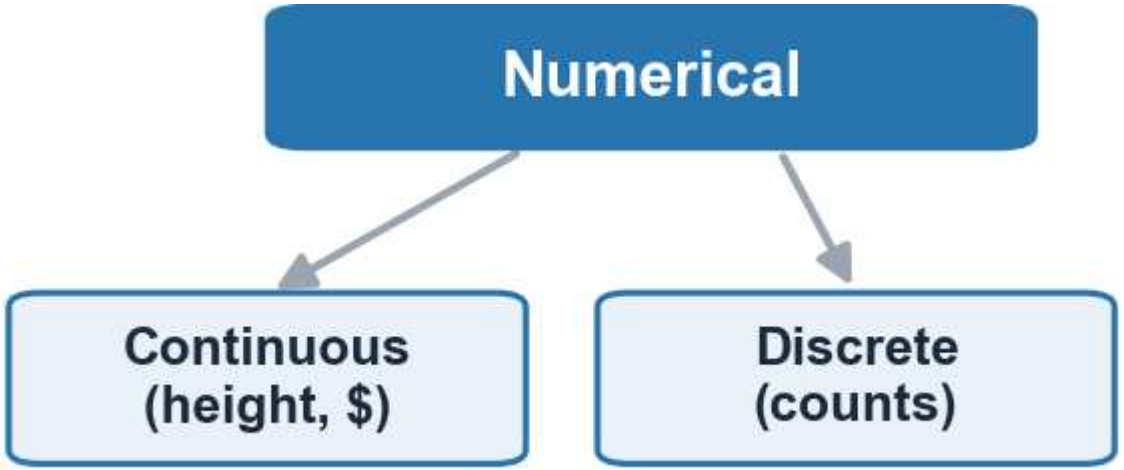
Remove, or better yet, relabel

Outliers: points far from the rest

Missing values

Wrong types/values

name	age	income	city	signup
Ana	24	55000	LA	2023-01-04
BEN	-1	62k	los angeles	01/05/2023
ana	24	55000	LA	2023-01-04
Carlos	thirty		NYC	2023/02/30
Dee	41	NaN	New York	



In some cases, the outliers should more obviously be removed (*e.g.*, a negative age)

Visualization helps us find these red flags when there is lots of data

Detecting outliers: box plots



Outliers lie outside the whiskers.
Investigate them!

Boxplot:

- The "box" ranges from the lower to upper quartile

75% of data lies below Q_3

25% of data lies below Q_1

- Interquartile range (IQR):

$$IQR = Q_3 - Q_1$$

(middle 50% of the data)

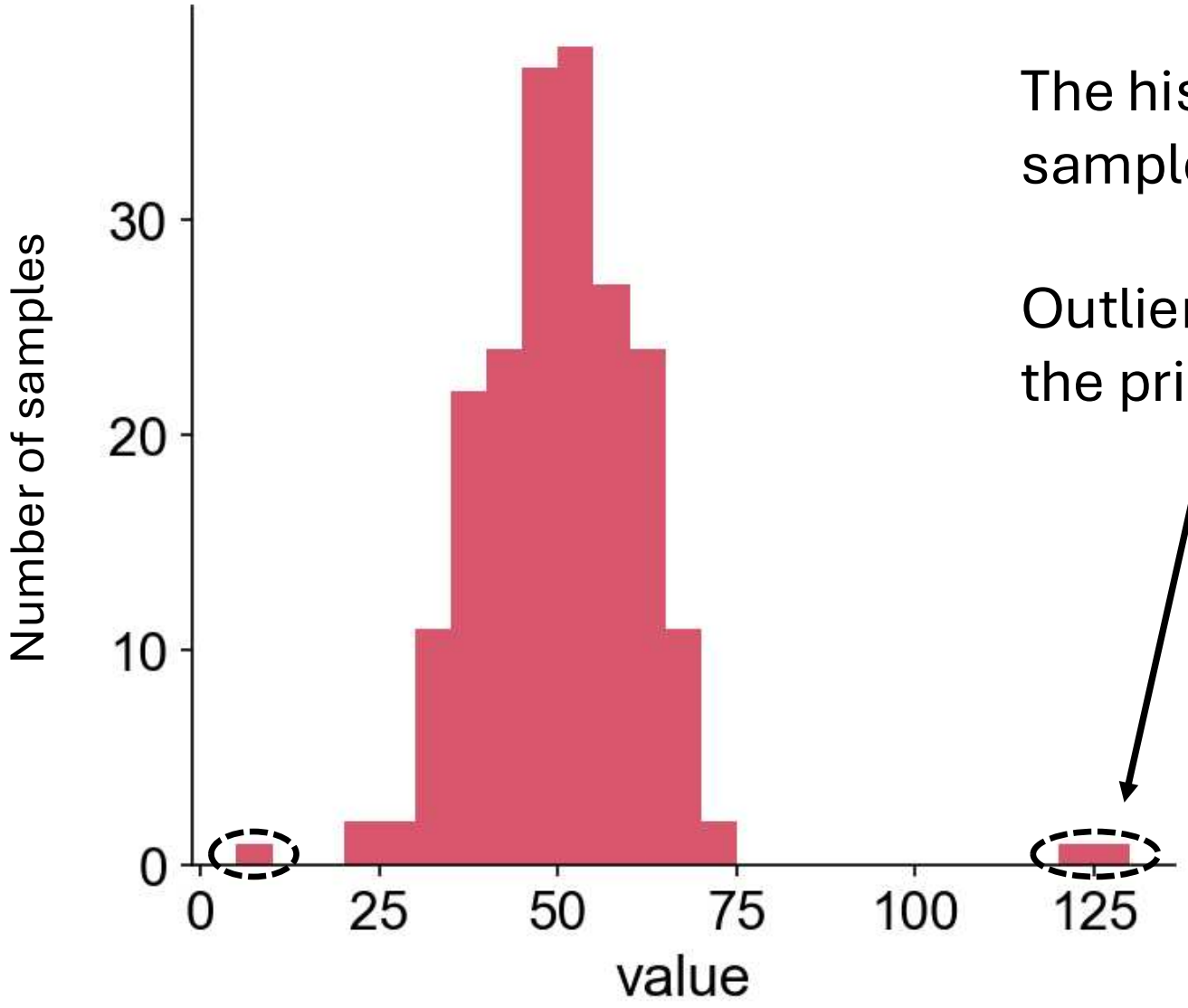
- Whiskers extend from:

$$Q_1 - 1.5 \cdot IQR$$

...to...

$$Q_3 + 1.5 \cdot IQR$$

Detecting outliers: histograms

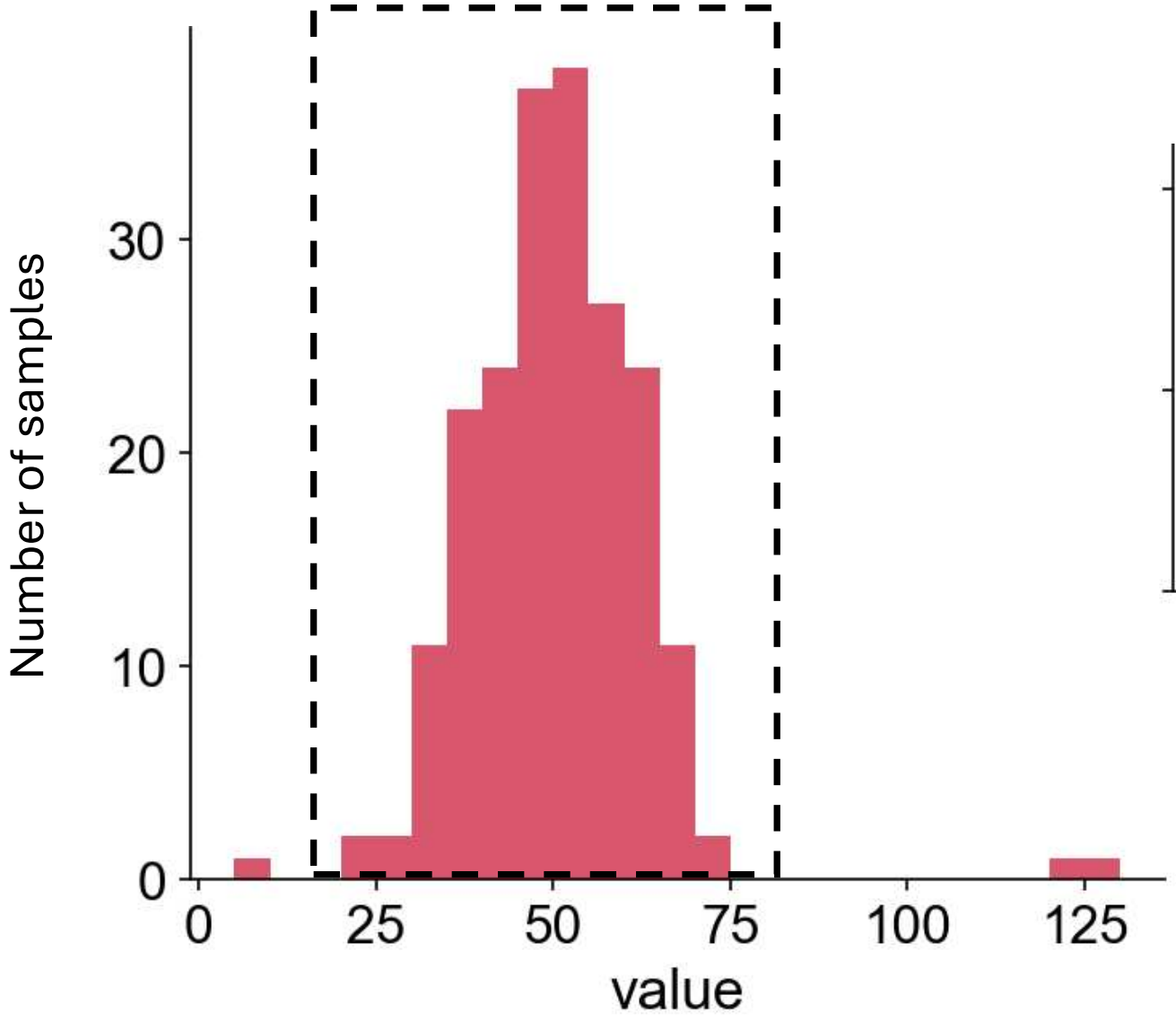


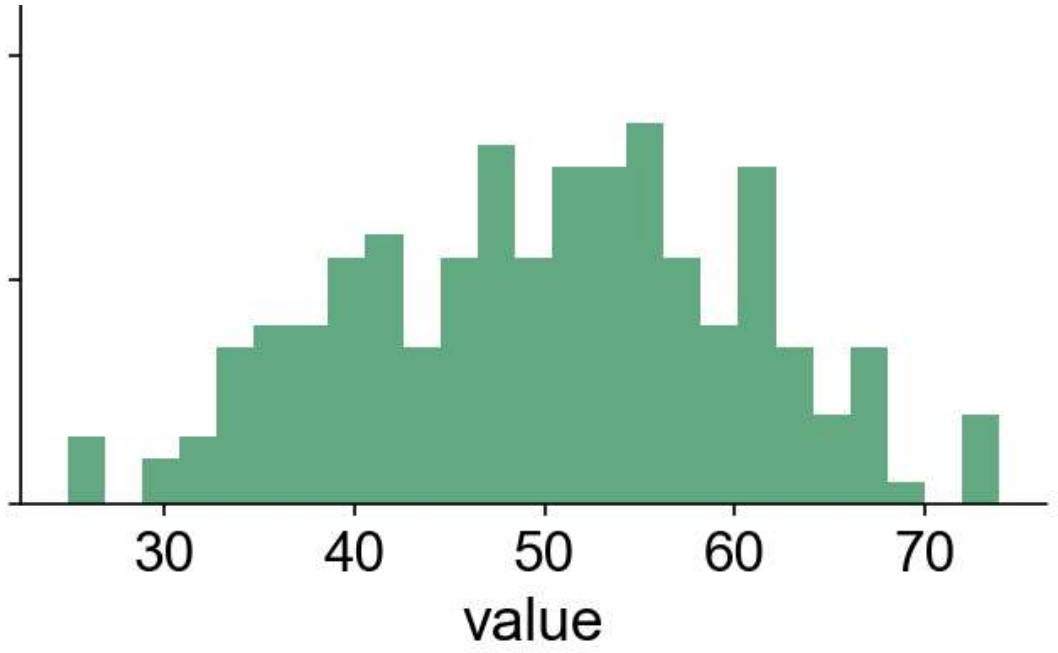
The histogram shows you how many samples lie within specifical intervals

Outliers are usually quite distinct from the primary distribution of data

It is usually **worth re-plotting without the outliers** (even if you don't remove them from the dataset)

Detecting outliers: box plots and histograms



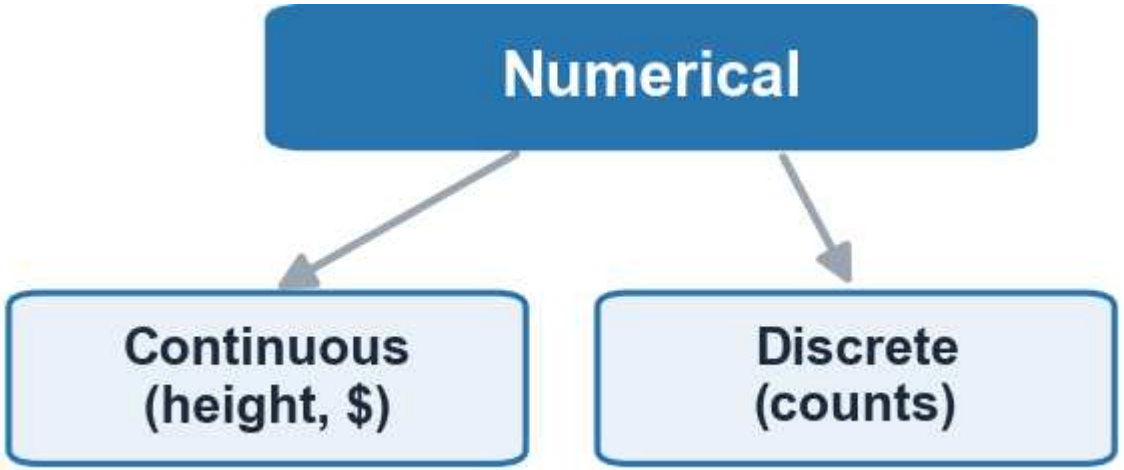


We can see more subtle details/variations

Duplicate records

Duplicates

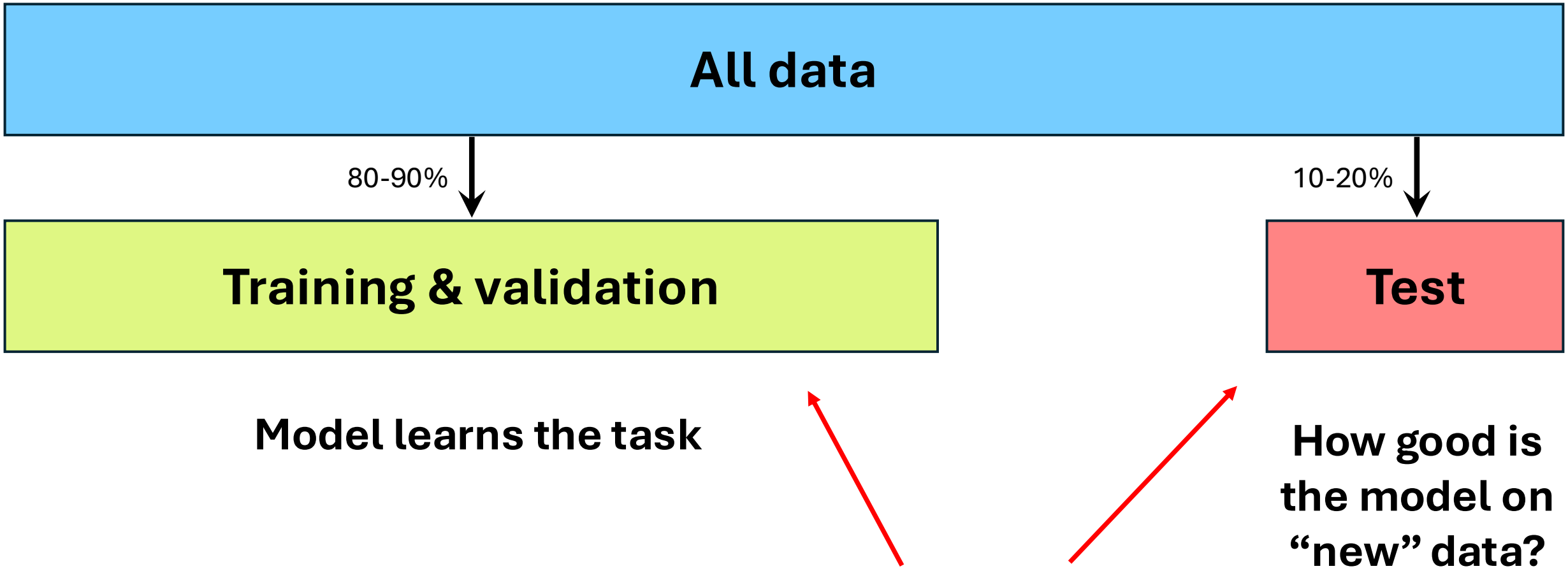
name	age	income	city	signup
Ana	24	55000	LA	2023-01-04
BEN	-1	62k	los angeles	01/05/2023
ana	24	55000	LA	2023-01-04
Carlos	thirty		NYC	2023/02/30
Dee	41	NaN	New York	



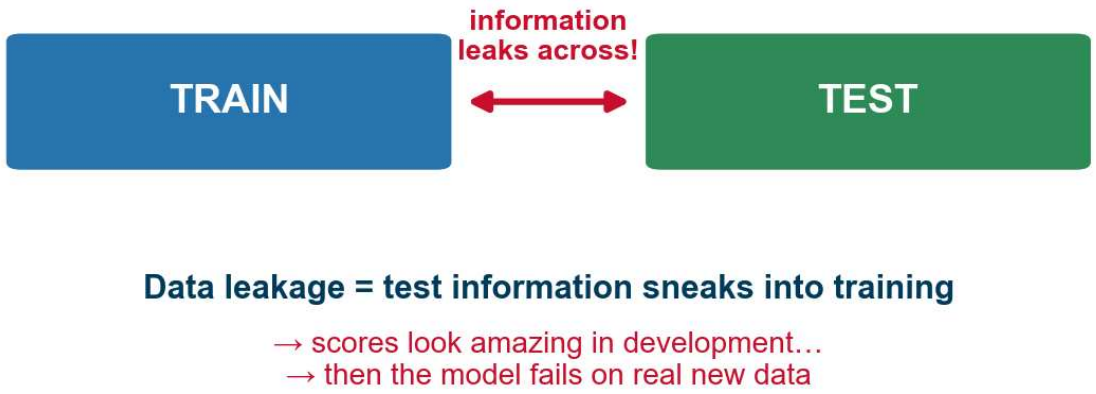
Notice the data from “Ana” and “ana” are the same…
We should remove one

Why? To avoid ***data leakage***

Recall: training *vs.* test datasets



Examples of data leakage



name	age	income	city	signup
Ana	24	55000	LA	2023-01-04
BEN	-1	62k	los angeles	01/05/2023
ana	24	55000	LA	2023-01-04
Carlos	thirty		NYC	2023/02/30
Dee	41	NaN	New York	

- Suppose we train a model to **predict income based on age and city**
- “Ana” data (row 1) gets placed in the training set
- “ana” data (row 3) gets placed in the test set
- Our model will *very* accurately predict the income of “ana”
- But this is ***not* real accuracy!** “ana” is not a new sample…

Examples of data leakage

From the training set

$Fe_{70}Cr_{19}Ni_9Mn_2$ → It's a good electrical conductor!



From the test set

$Fe_{0.69}Cr_{0.21}Ni_{0.08}Mn_{0.03}$ → It's a good electrical conductor!



"Wow, my model works great!"

(unfortunately, quite common)

Four classic ways data leakage occurs

Scaling on ALL data

computed mean/σ using test rows

Duplicates across split

same row in train AND test

Target leakage

a feature secretly contains the answer

Temporal leakage

using the future to predict the past

An example of target & temporal leakage

Patient ID	Age (years)	Blood Pressure (mmHg)	ICU Admission	Discharge Date (known only after outcome)	Survived? (Target)
P001	42	118	No	2026-05-08	Yes
P002	71	165	Yes	2026-05-21	Yes
P003	59	152	Yes	—	No
P004	33	122	No	2026-05-06	Yes
P005	84	178	Yes	—	No
P006	65	140	Yes	2026-05-18	Yes
P007	28	110	No	2026-05-02	Yes
P008	77	170	Yes	—	No
P009	50	130	No	2026-05-14	Yes
P010	63	155	Yes	2026-05-10	Yes

Our model learns that if a patient was discharged, they must not have died in the hospital…

Accurate but trivial – it uses no other information!

Questions?

Four classic ways data leakage occurs

Scaling on ALL data

computed mean/σ using test rows

Why scale in the first place?

Duplicates across split

same row in train AND test

Target leakage

a feature secretly contains the answer

Temporal leakage

using the future to predict the past

Features sometimes live on different scales

Suppose we are building a model to predict car price based on its **mileage** and **number of previous owners**



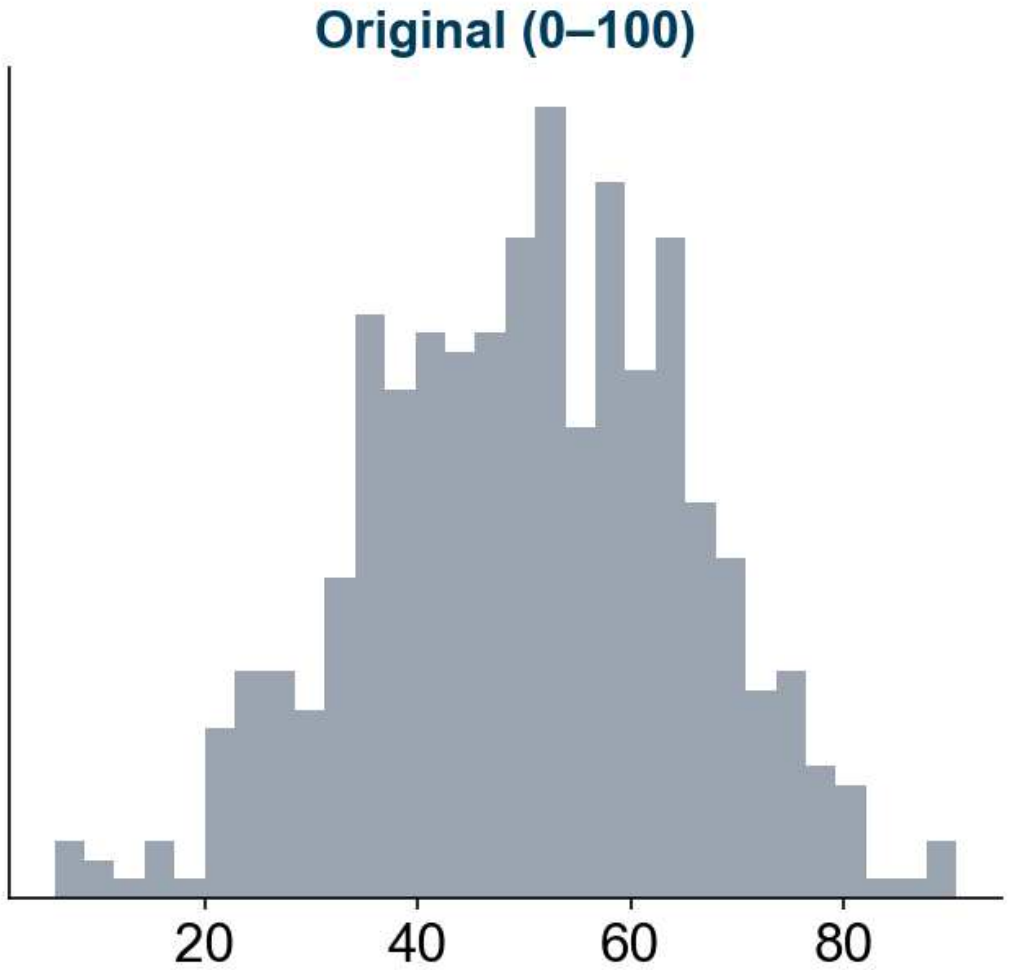
Number of previous owners: 1-5

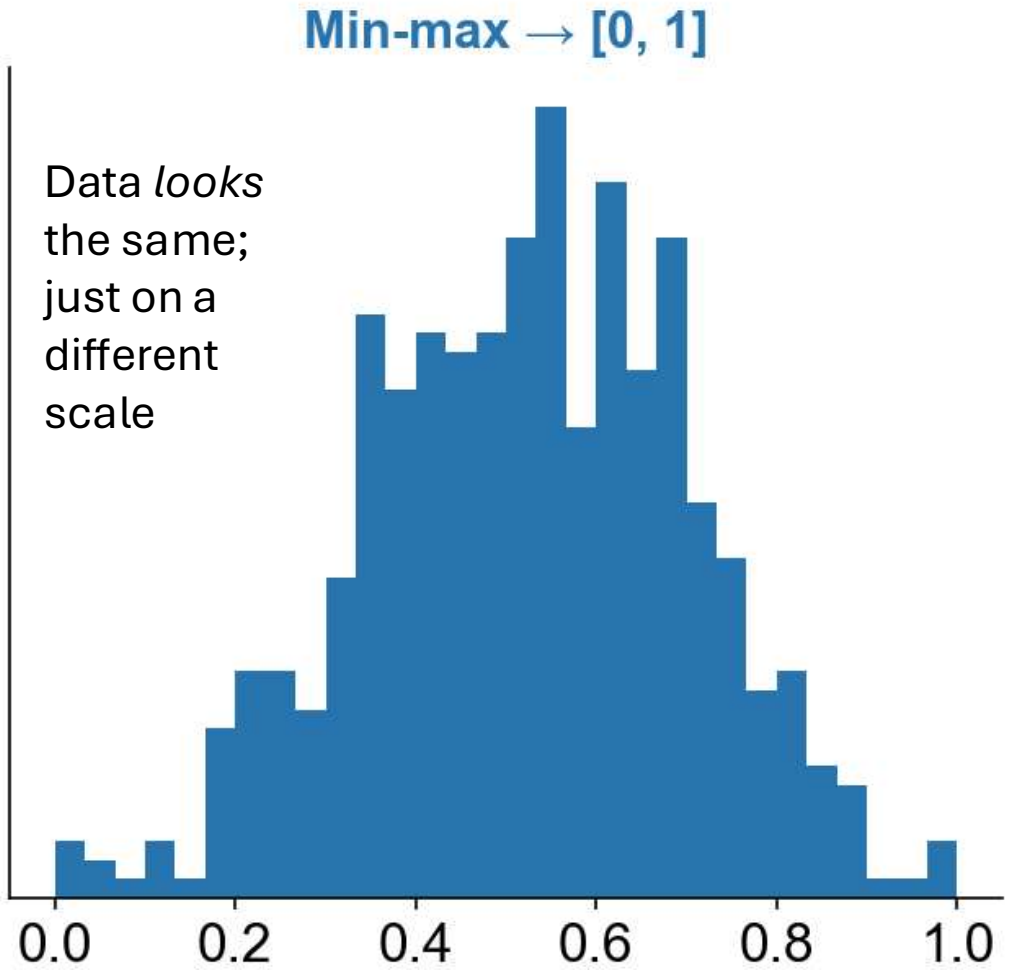


Mileage: anywhere from 0 to 300,000 miles

Without scaling, **mileage values are thousands of times larger** and can dominate distance-based analyses, even though previous owners may be just as influential.

Normalization (min-max)





Subtract the minimum and divide by the range

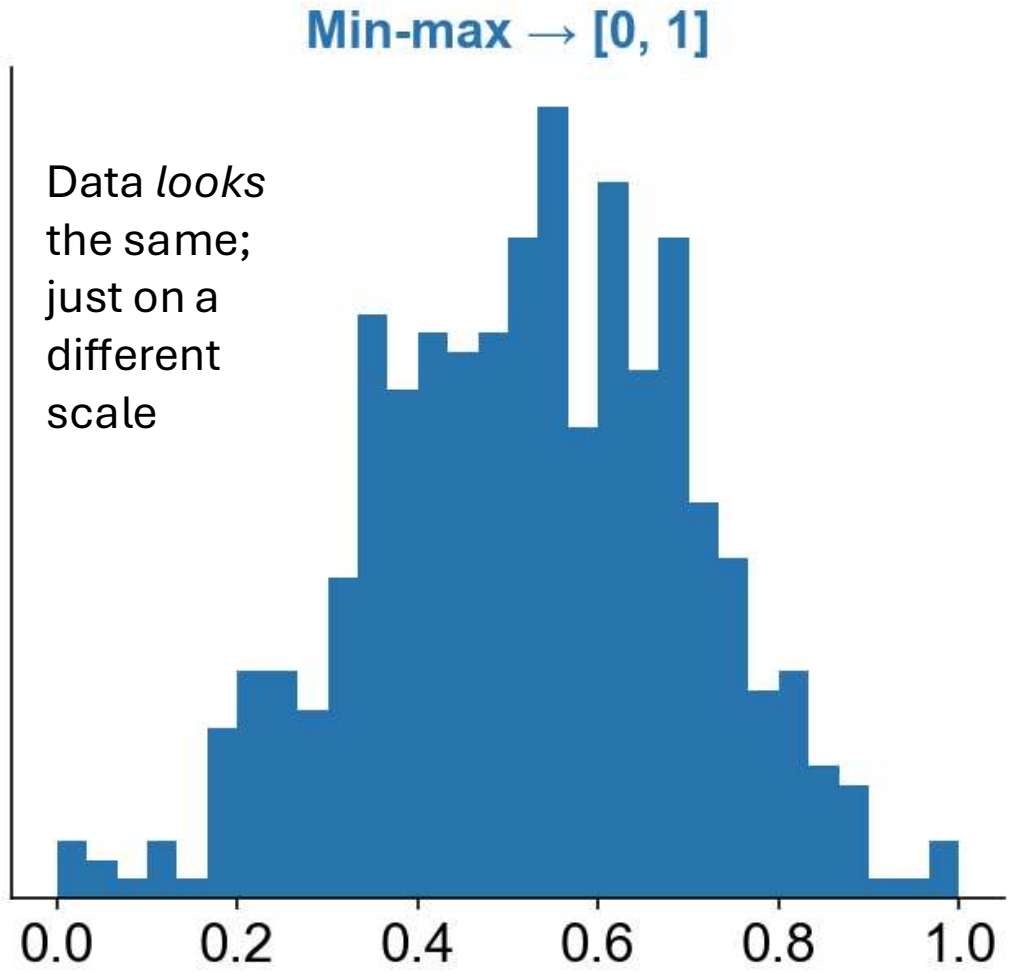
$$x' = \frac{x - \min}{\max - \min}$$

Do this for all variables to be consistent!

Normalization (min-max)

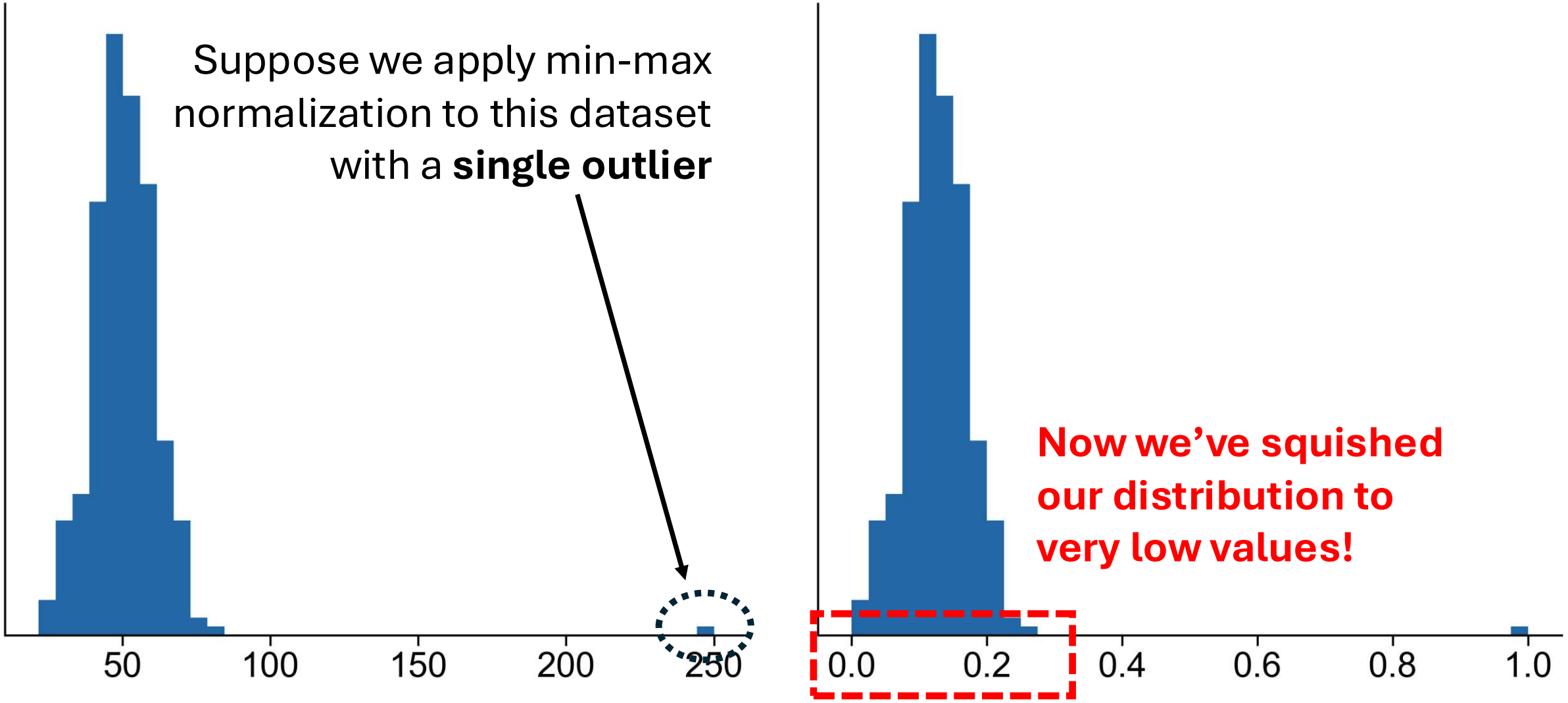
Problem: this min-max approach is highly sensitive to outliers!

Suppose we have one outlier with a huge value – all other "normal" values (the ones we care about) will get scaled to small numbers

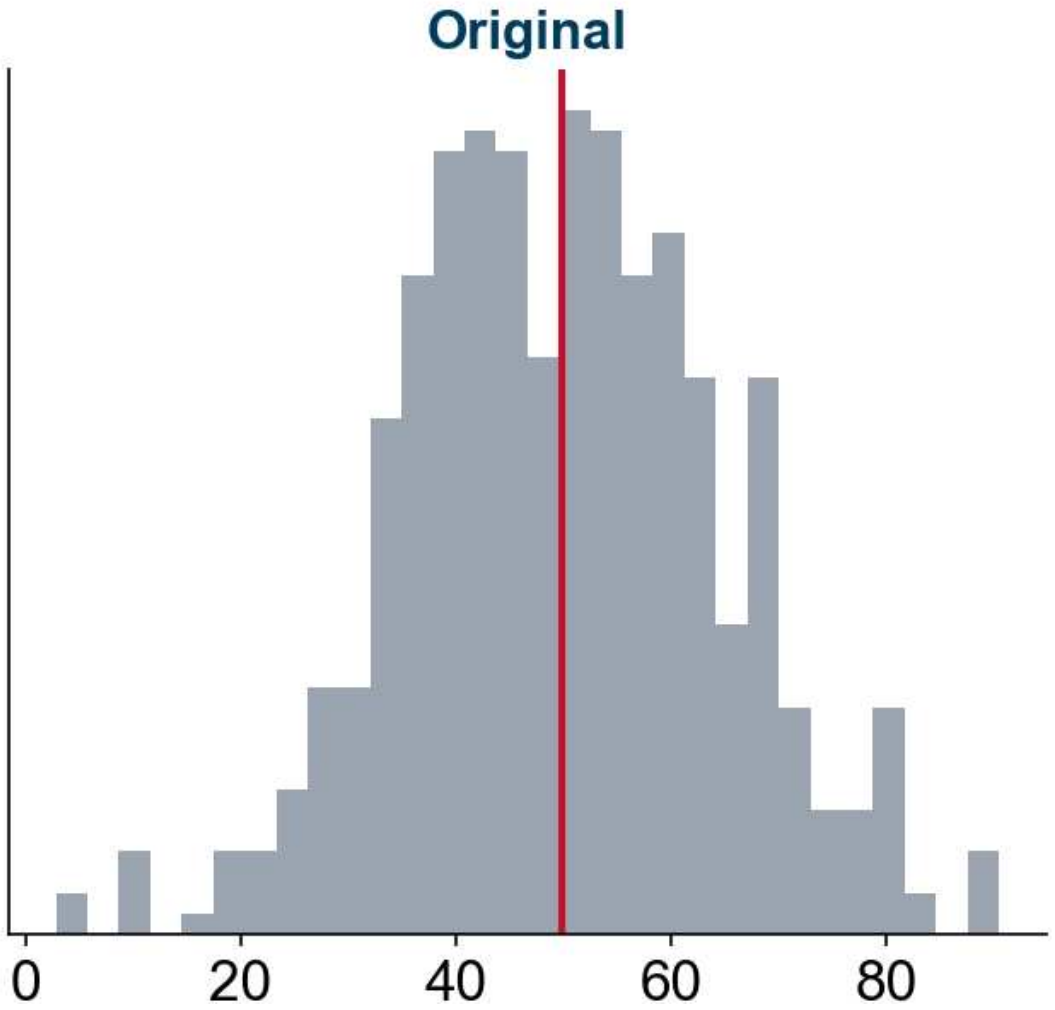


Do this for all variables to be consistent!

The problem with min-max normalization



Standardization (z-score)



Each sample

Mean of the data

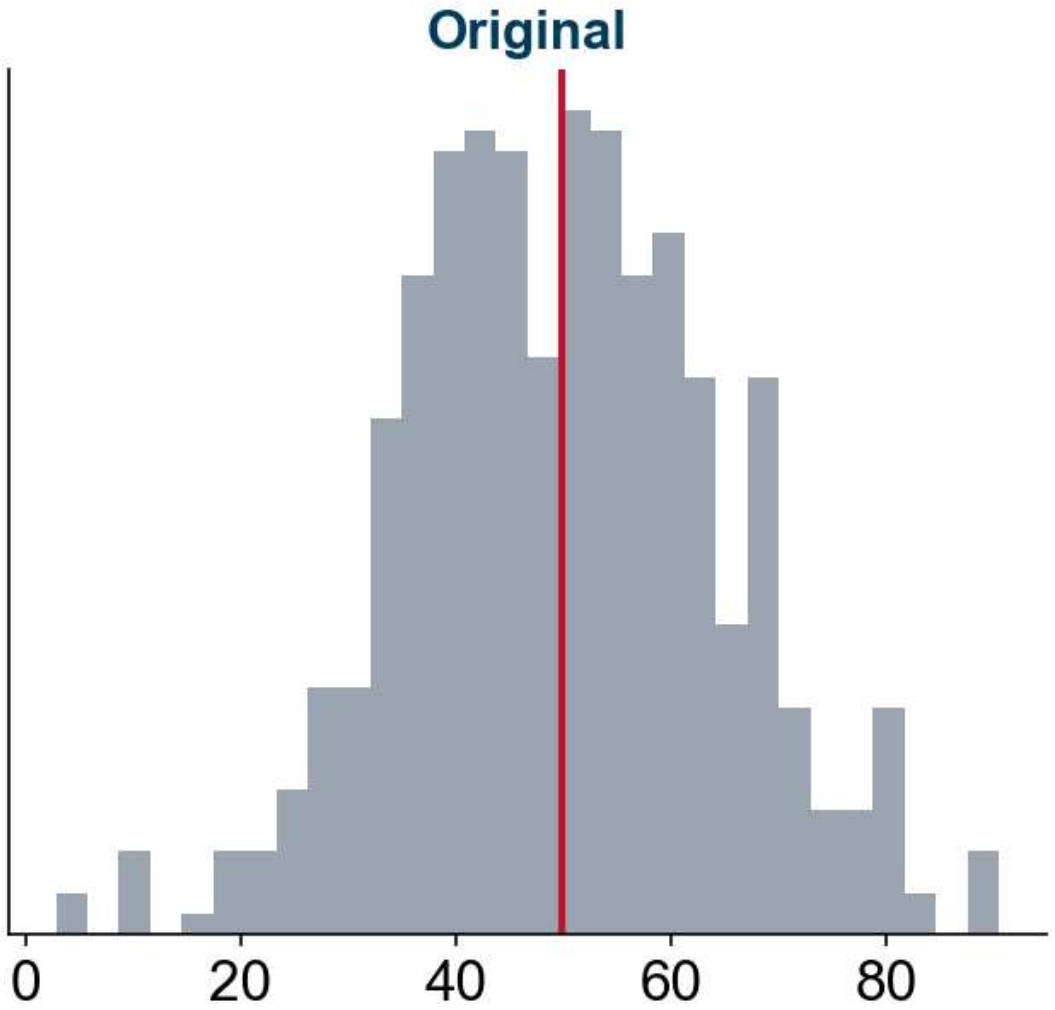
$$Z = \frac{x - \mu}{\sigma}$$

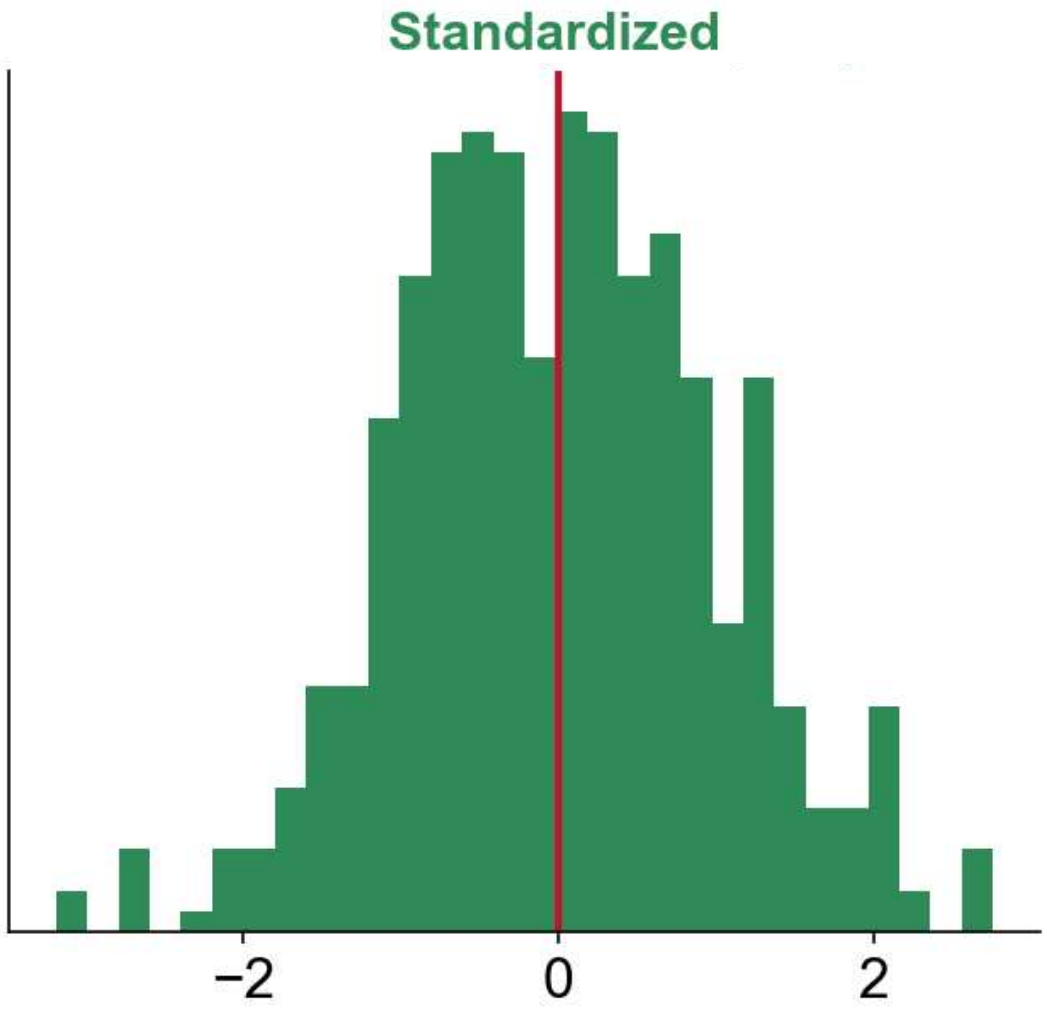
Standard deviation

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^2$$

1) Subtract the mean to **zero-center** data
2) Divide by Z to **standardize** it

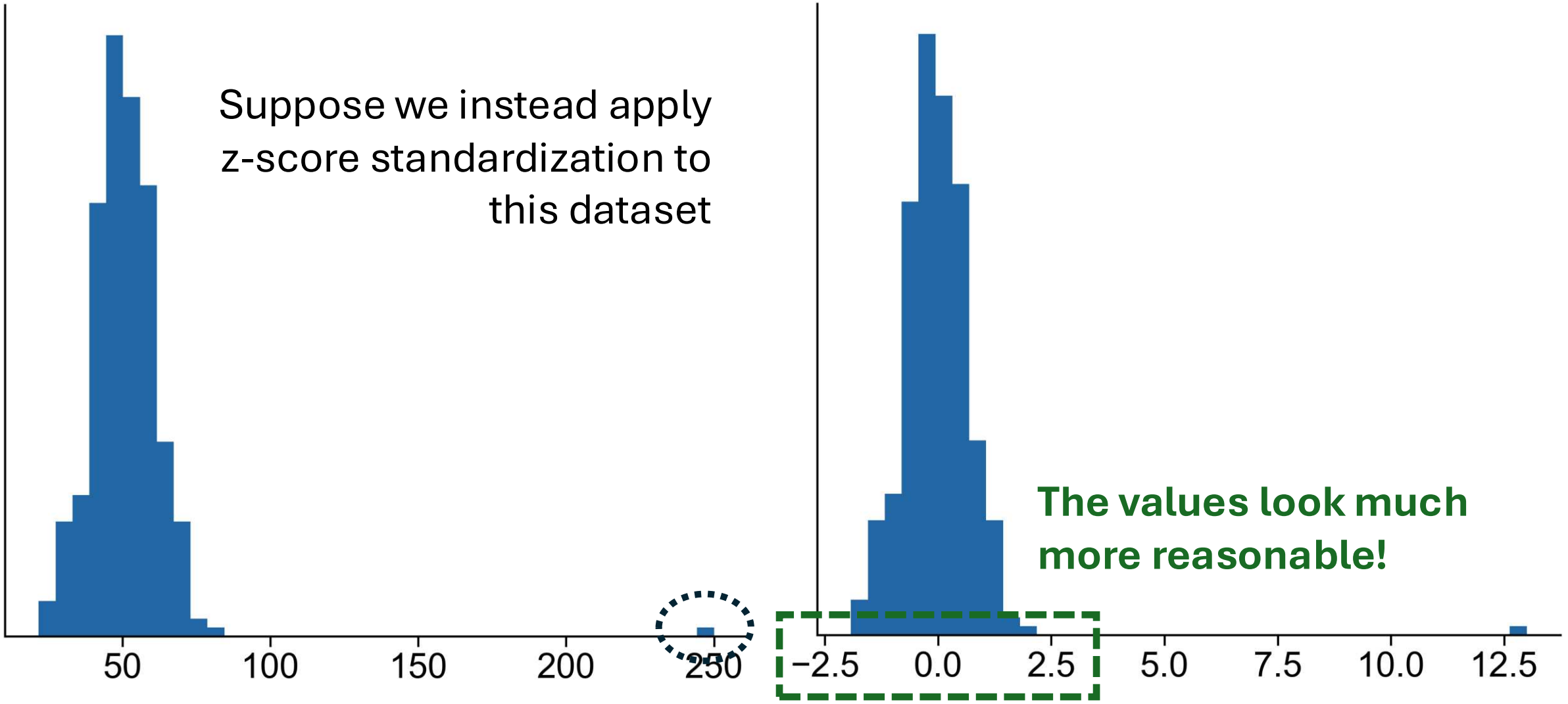
Standardization (z-score)





Standardization (z-score)

Suppose we instead apply z-score standardization to this dataset



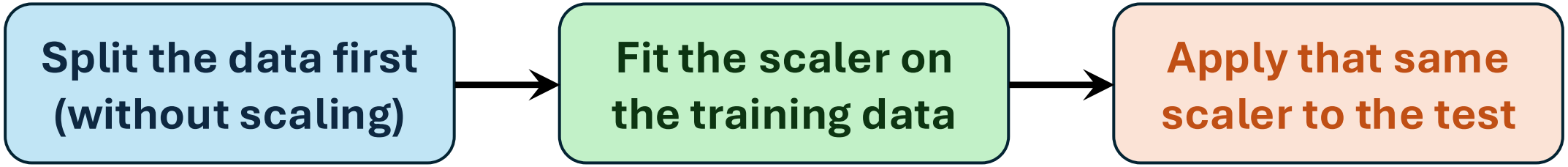
The values look much more reasonable!

But be careful…Scaling is the #1 leakage trap!

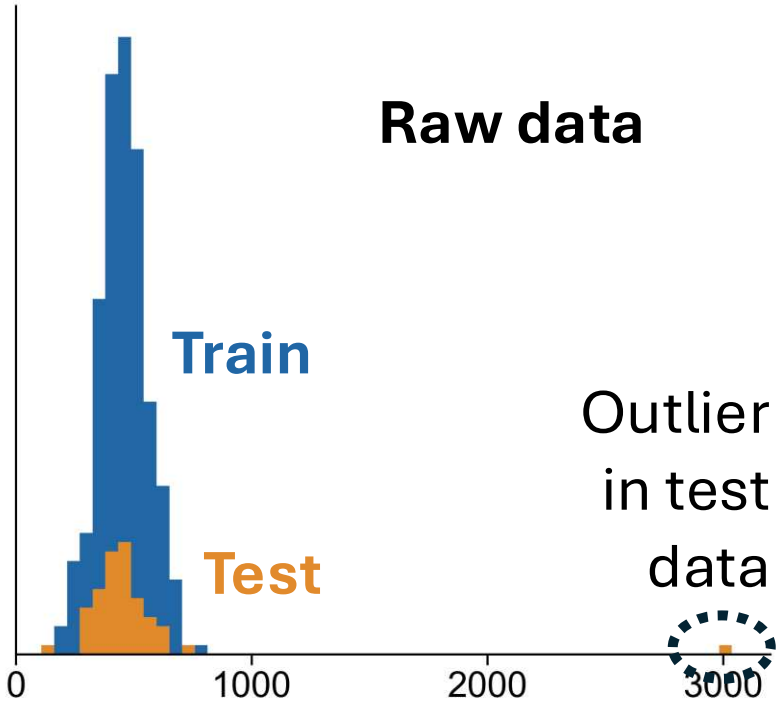


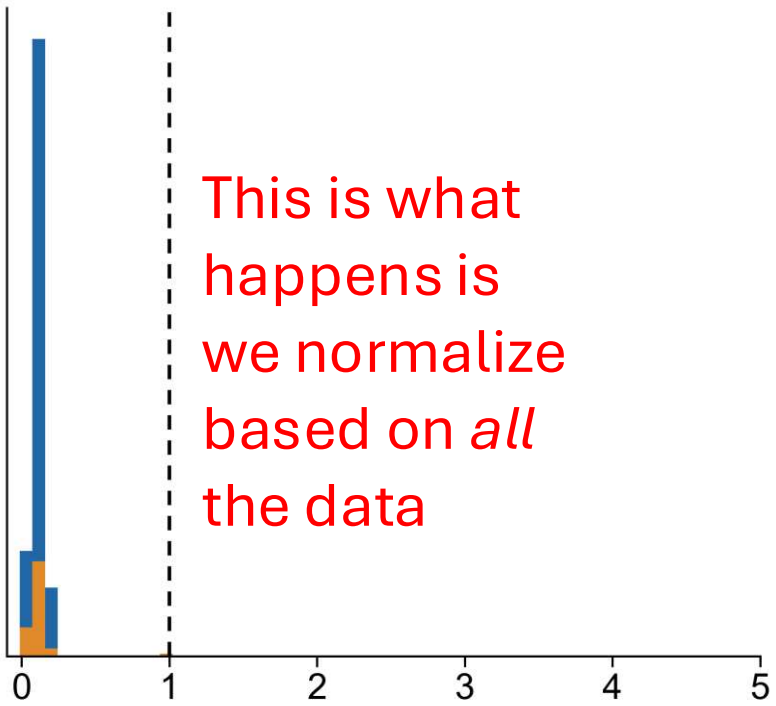
Common mistake: people scale data using the entire population (training + test)

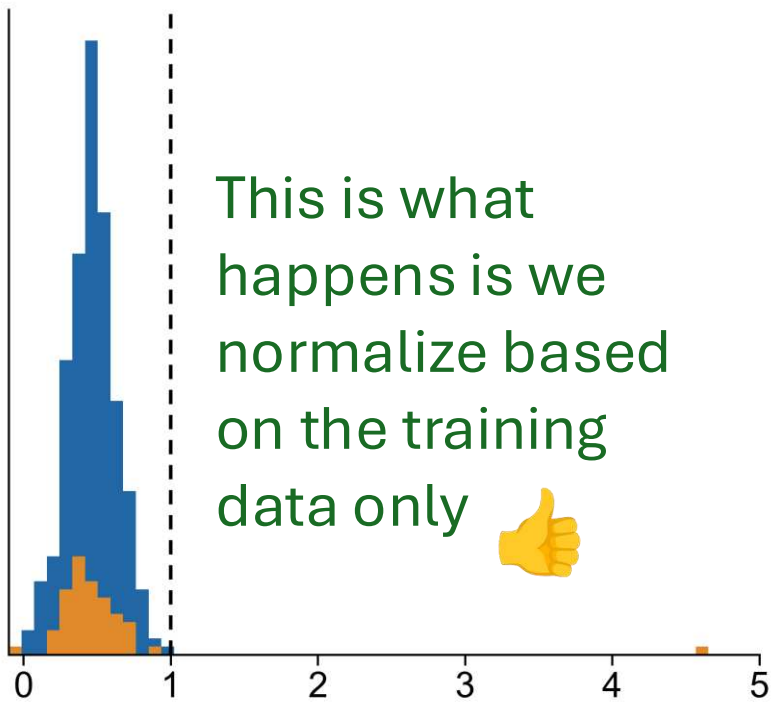
The model has indirectly been given information about the test set *before* evaluation.



But be careful…Scaling is the #1 leakage trap!

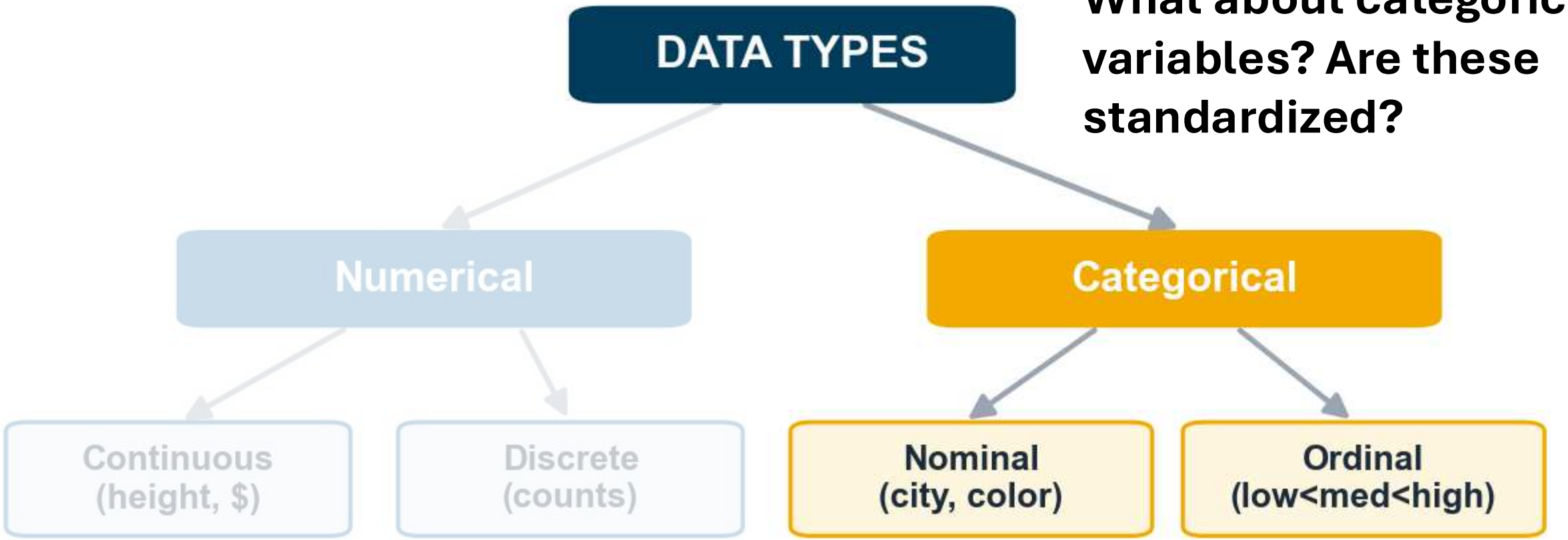






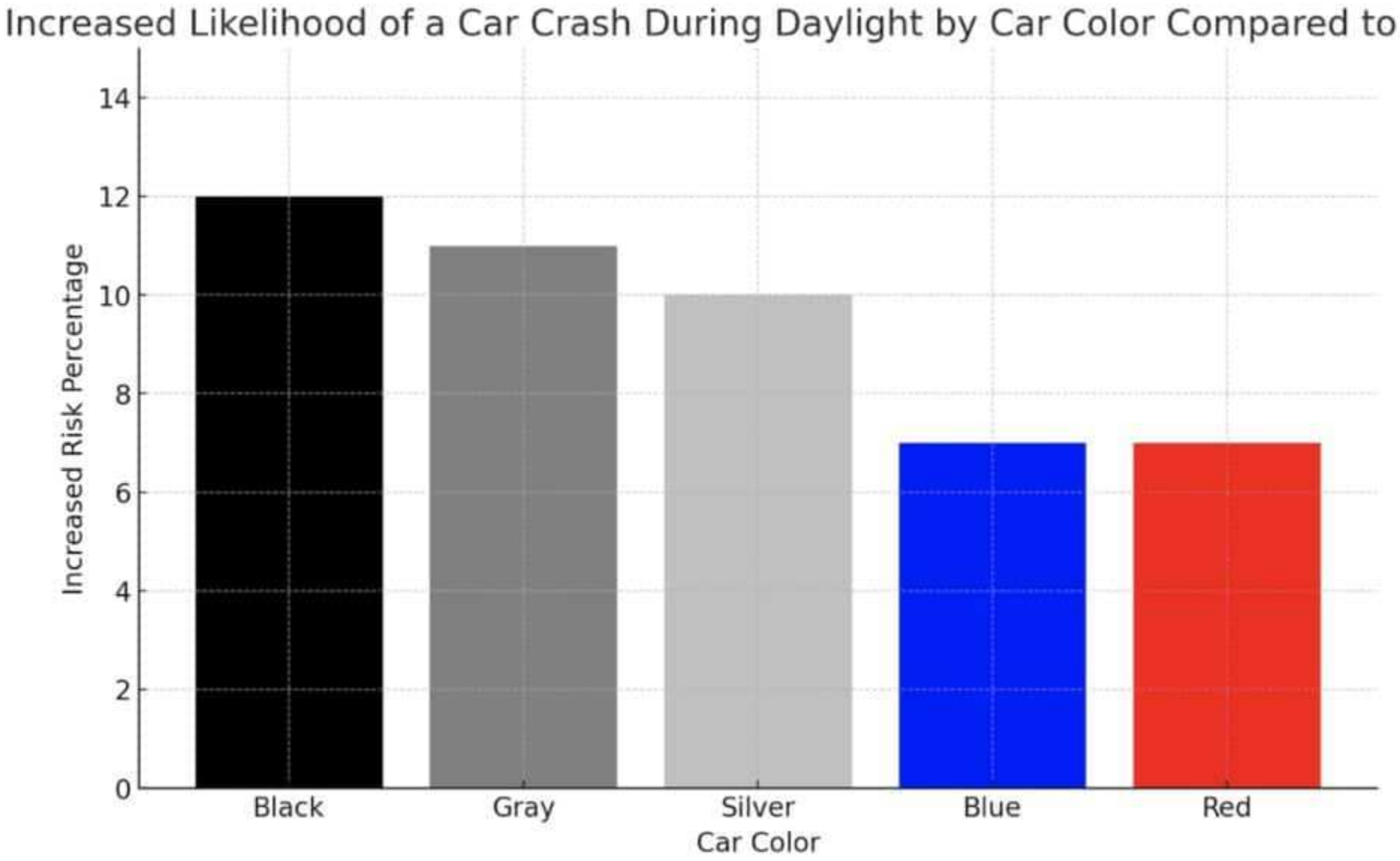
If we normalize based on *all* the data, our training data now has some implicit knowledge of the test data…Not good!
Don't do this! Very common error!

Know your data types



What about categorical variables? Are these standardized?

An example of a categorical variable: Car color





Likely reason:
Darker cars are harder to see

What if we wanted to build a model to *predict* crash rate from car color?

One-hot encoding of categorical variables

Most of the models we'll cover perform mathematical operations **on numbers**
...But **categorical variables (like color) are *not* numbers**...

The most common solution is **one-hot encoding**

Categorical column

color
red
blue
green

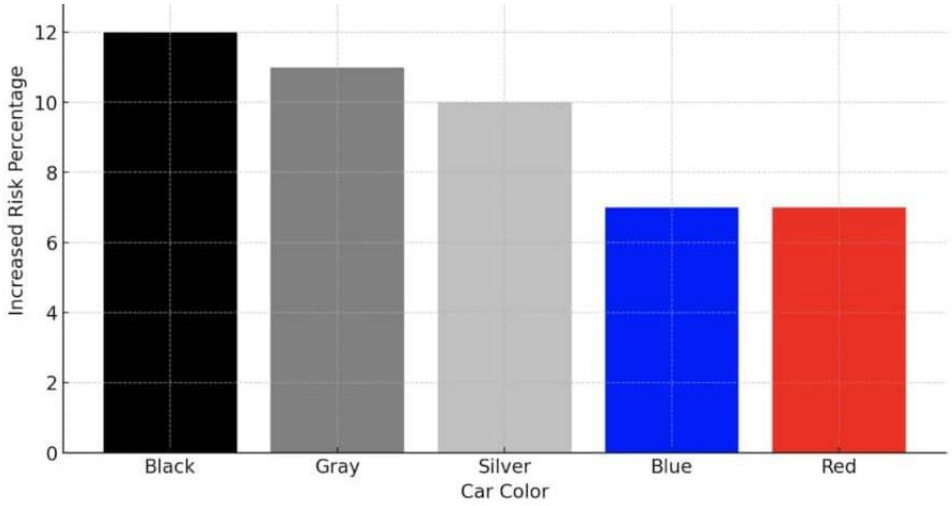
One-hot encoded

is_red	is_blue	is_green
1	0	0
0	1	0
0	0	1

Problems with one-hot encoding

These vectors are all **completely orthogonal** to one another

This means our representation of the data has **no knowledge of "similarity"** between different colors



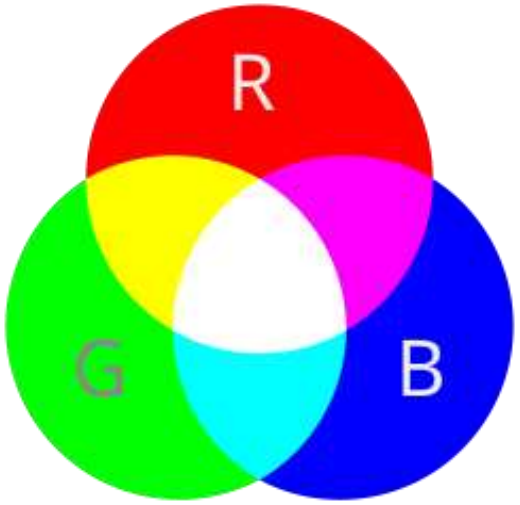
In reality, we know that black and gray are more similar to one another than black and red…

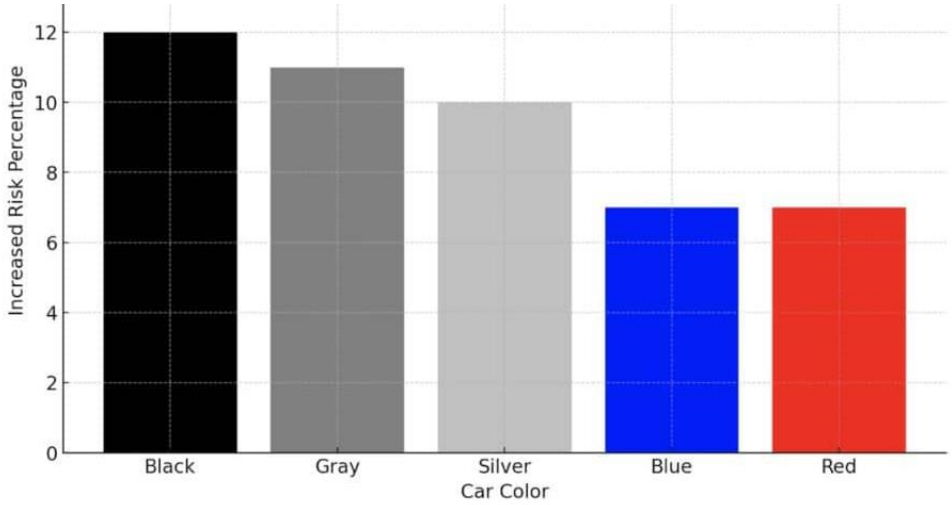
The question is:
How to encode this?

is_red	is_blue	is_green
1	0	0
0	1	0
0	0	1

Use *meaningful* encodings when possible

We can instead use RGB values on the color wheel





Color	R	G	B
Black	0	0	0
Gray	128	128	128
White	255	255	255
Red	255	0	0

Remember: We should standardize these values

Use ordered encodings when possible

Ordinal (has order)

size	code
small	0
medium	1
large	2

fine: large > medium > small

Nominal (NO order)

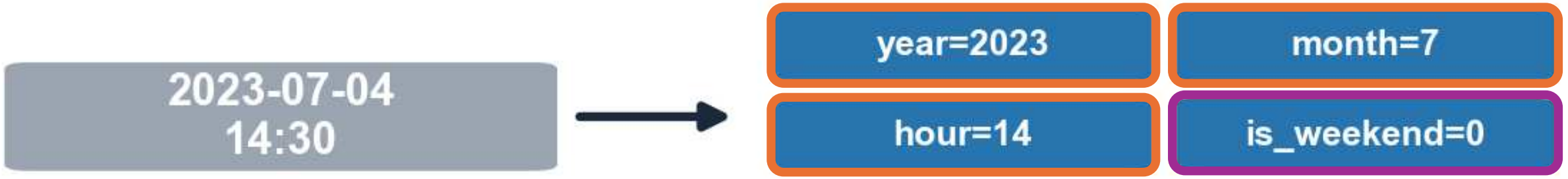
city	code
LA	0
NYC	1
SF	2

danger: implies SF > NYC > LA !

- Use one-hot encoding or a meaningful representation (depends on target)
- **EX:** if the target is weather-related, perhaps use longitude/latitude

Ordering also applies naturally to dates

Make each part of the date its own feature



You can also add descriptive features like “is_weekend” if you think it matters

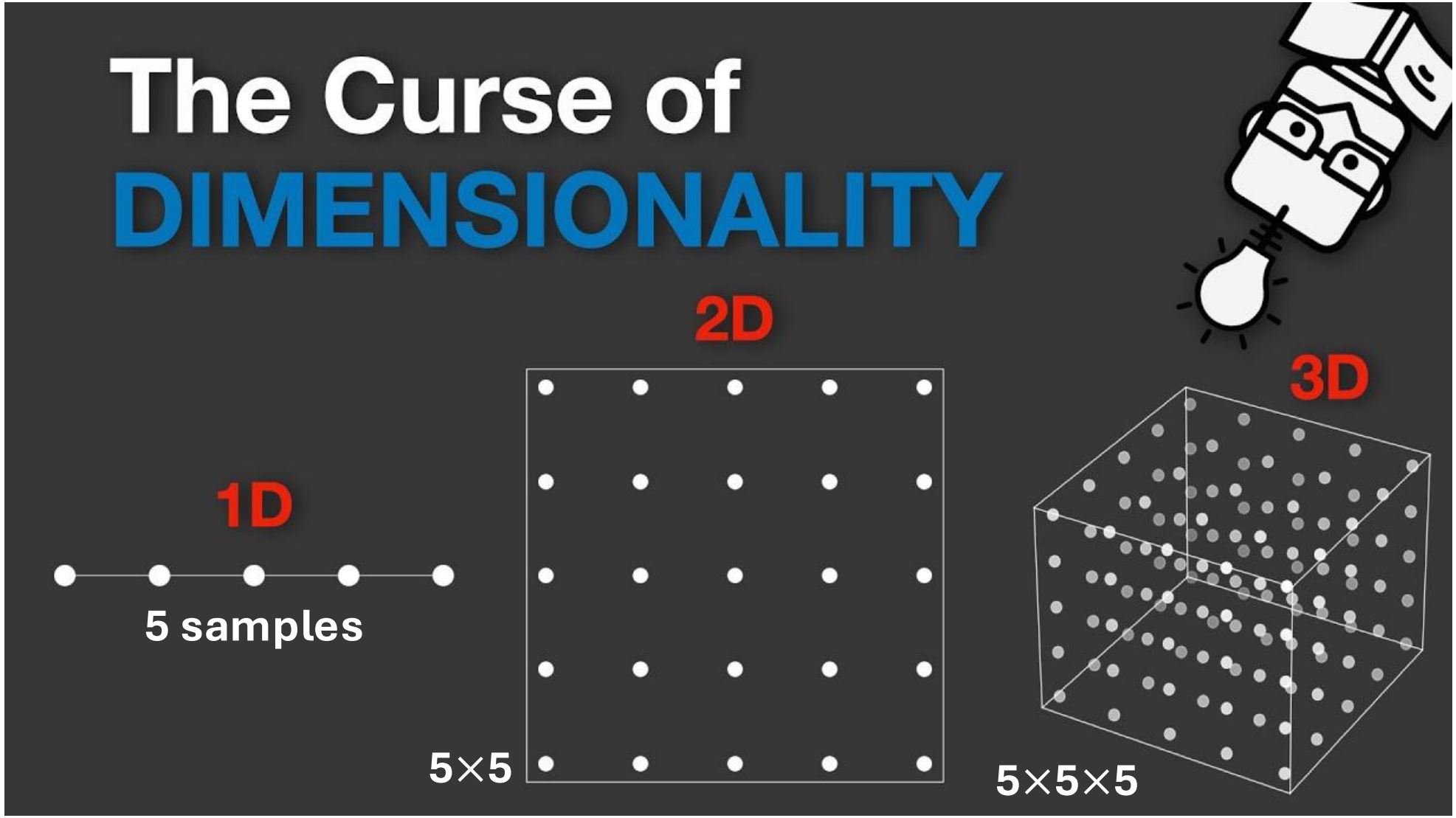
What about natural language?

"I walked the dog on a leash."

I	walked	the	dog	on	a	leash
1	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	1

1) This leads to a huge number of rows/features…***The curse of dimensionality***

The curse of dimensionality



As the number of features increases, **exponentially more data is needed to adequately sample** and learn patterns in the space.

What about natural language?

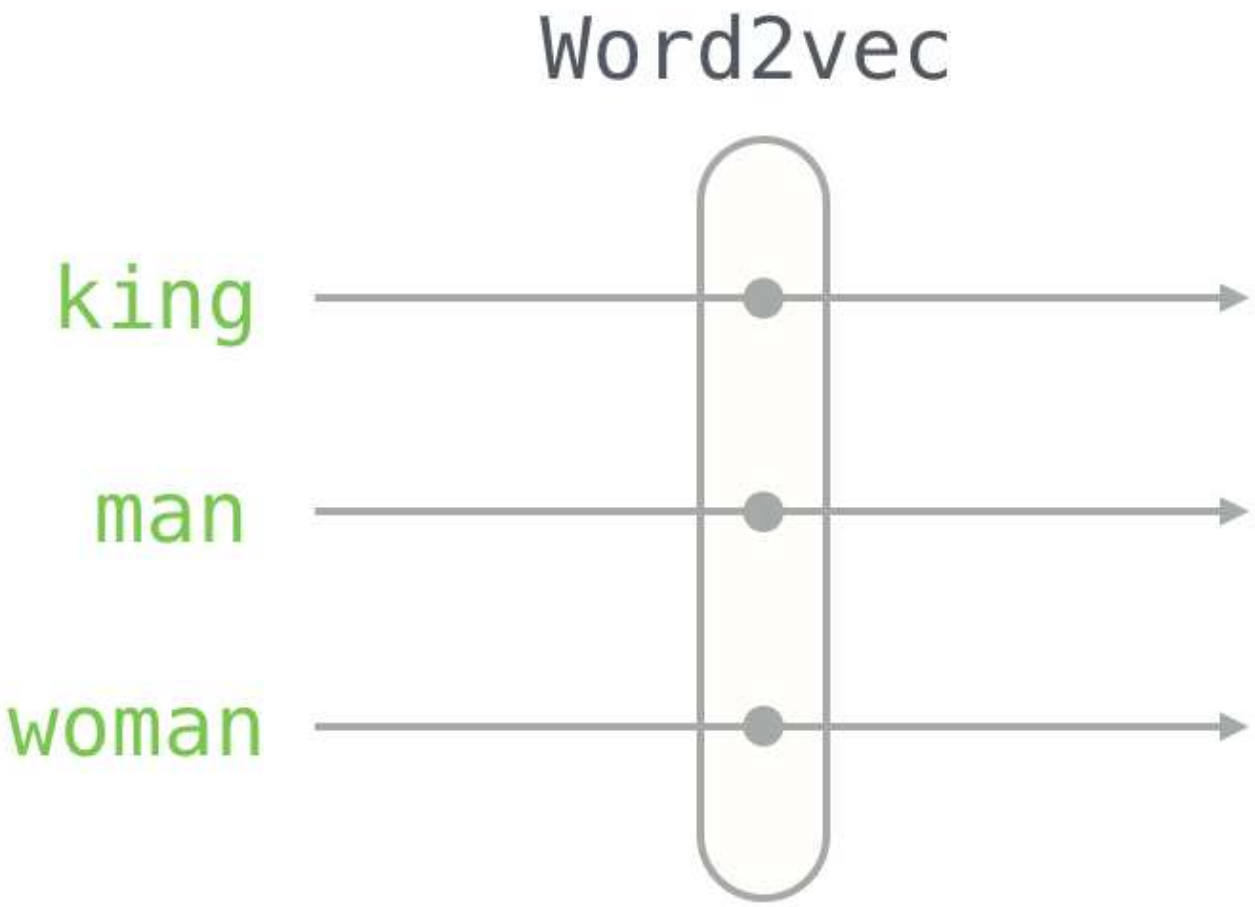
"I walked the dog on a leash."

I	walked	the	dog	on	a	leash
1	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	1

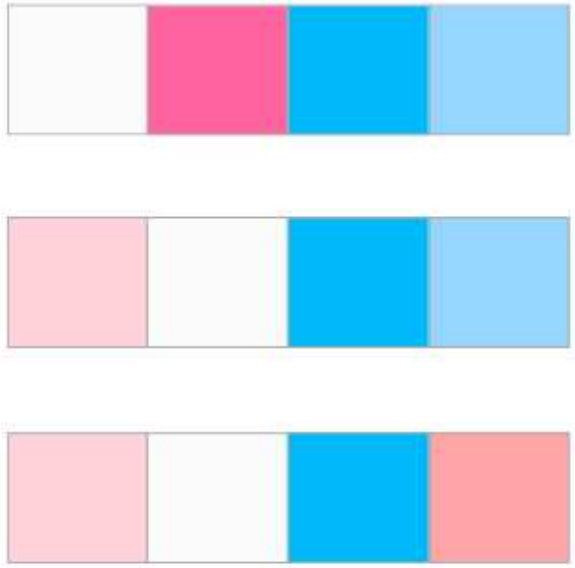
1) This leads to a huge number of rows/features…***The curse of dimensionality***

2) There is **no relation between words**; but *we* know "dog" and "leash" are related

Encoding text using Word2Vec



Breakthrough from Google in 2013:
Train an ML model to learn optimal embeddings for natural language



These vectors capture **similarity*** between highly related words!

***Often appearing in the same sentences.**

Mikolov, Tomas; Chen, Kai; Corrado, Greg; Dean, Jeffrey (2013). "Efficient Estimation of Word Representations in Vector Space". arXiv:1301.3781

Questions?

What to know for HW and midterm

- Effects of **sample size and bias**
- **Types of data** (numerical and categorical)
- Different **ways to sample the data**
- The kinds of ***missing*** **data** (MCAR, MAR, MNAR)
- **Imputation** approaches
- **Scaling and standardization** (while avoiding data leakage)