University of Pittsburgh | School of Public Health

Training statistical models
- over- and under-fitting
- resampling techniques

BIOST 2155

Class 2
September 5th, 2025

Week 1: fantastic error metrics and where to find them

	Features (input)	Targets/ labels (output)	Task type	ML type	Metric
Prediction of brain volume from gestational age	✅	✅	Regression	**Supervised learning**	***RMSE***, *R-squared*
Classification of heart failure using ultrasound or magnetic resonance imaging	✅	✅	Classification		***Accuracy***, *Precision, Recall, AUC*
Patient health profiles using wearable device data	✅	🚫	Clustering	**Unsupervised learning**	***Silhouette score***
Compressing information from 30 features to study similarities/differences in benign and malignant breast tumors	✅	🚫	Dimensionality reduction		***Explained Variance***

https://soumik.shinyapps.io/class2app1/

More models, more problems…

How good is our model?

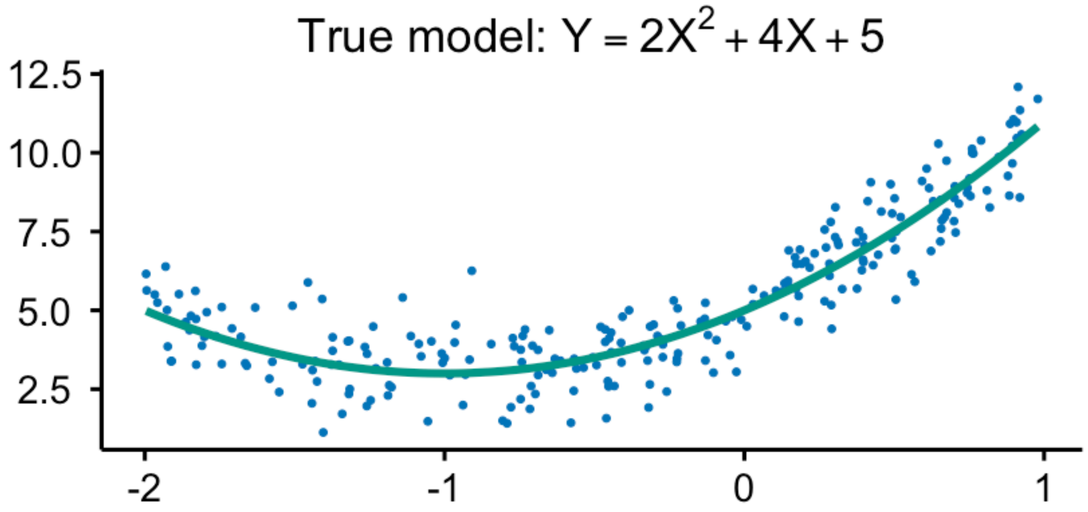
1. Our goal isn't just to **explain** the data we **observe**.

2. We want to build a model that **generalizes** well to new, **unseen** data.
   A. A model that fits our training data perfectly might not be the best model.
   B. It might have just "memorized" the data and not "understood" it.
   C. This leads to a central challenge of ML: **under-fitting** vs. **over-fitting**.

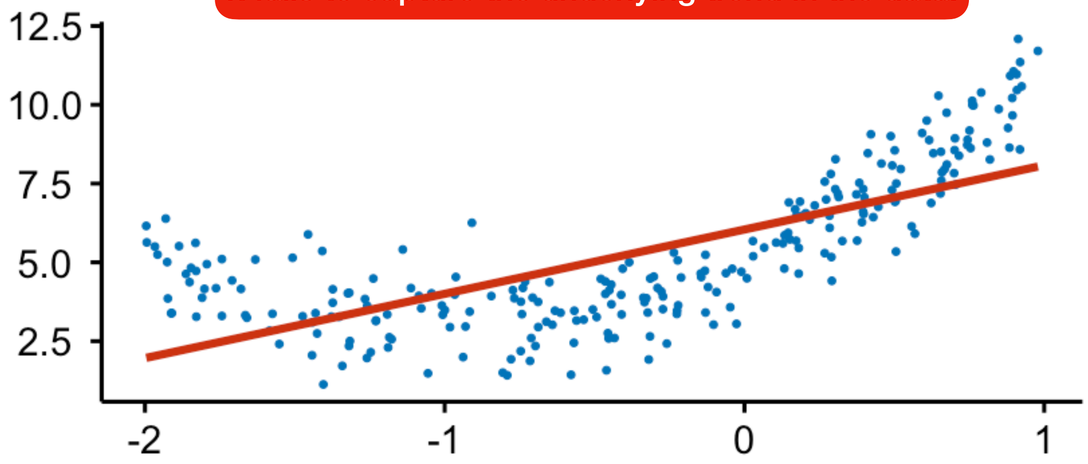
3. Consider $X \sim U(-2,1)$ and $Y = 2X^2 + 4X + 5 + N(\mu = 0, \sigma = 1)$.
   1. We observe $\{(X_i, Y_i)\}_{i=1}^{250}$ but don't know the relationship between X, Y
   2. Want to fit a model $Y = f(X)$.

figure1

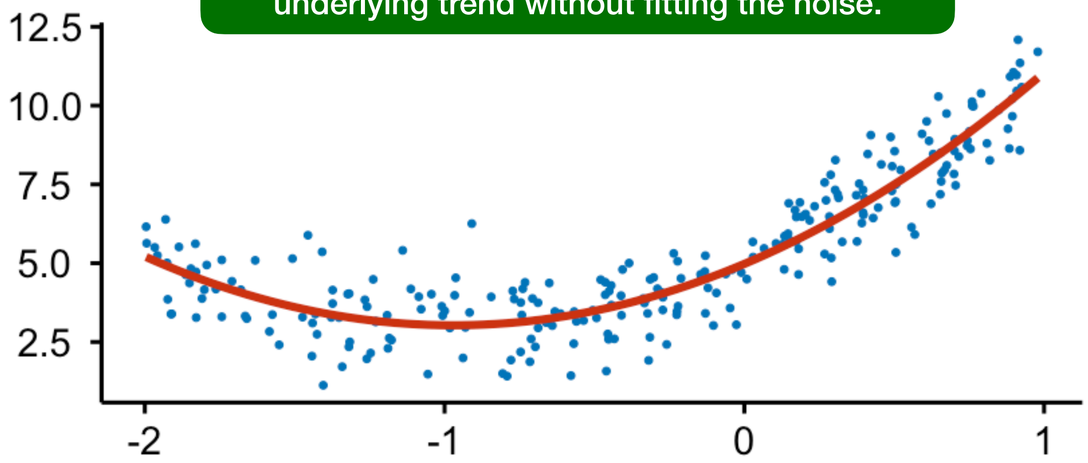
More models, more problems…



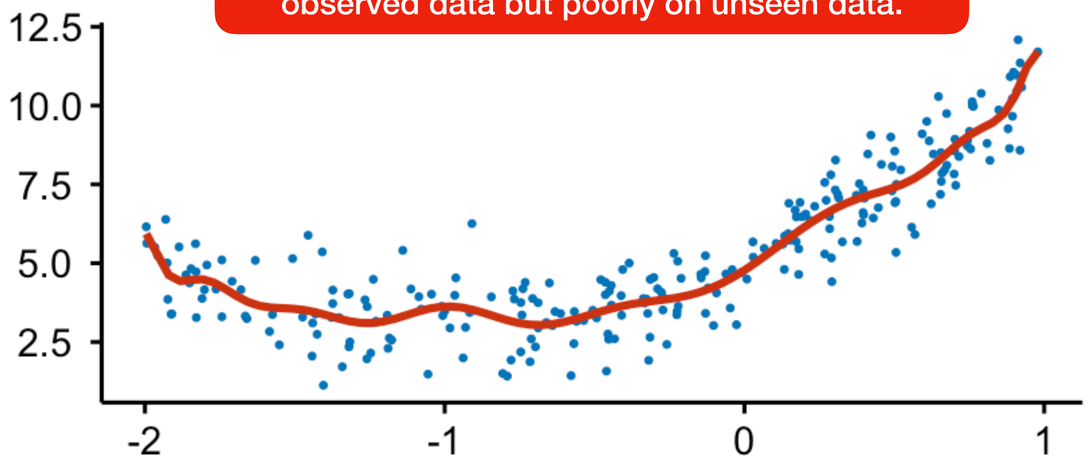
The model is “too simple”.
It fails to capture the underlying trend in the data.



The model is "just right." It captures the underlying trend without fitting the noise.



The model is “too complex”. Will perform well on observed data but poorly on unseen data.



Assessing model accuracy

How good is our model?

1. Our goal isn't just to **explain** the data we **observe**.

2. Want to build a model that **generalizes** well to new, **unseen** data.
    1. We split the data up into two smaller sets
    2. Training data **Tr:** $\{x_i, y_i\}_{i=1}^{N}$
    3. Test data **Ts:** $\{x_i, y_i\}_{i=1}^{M}$
    4. Typically, $N/M \approx 4$

https://soumik.shinyapps.io/class2app2/

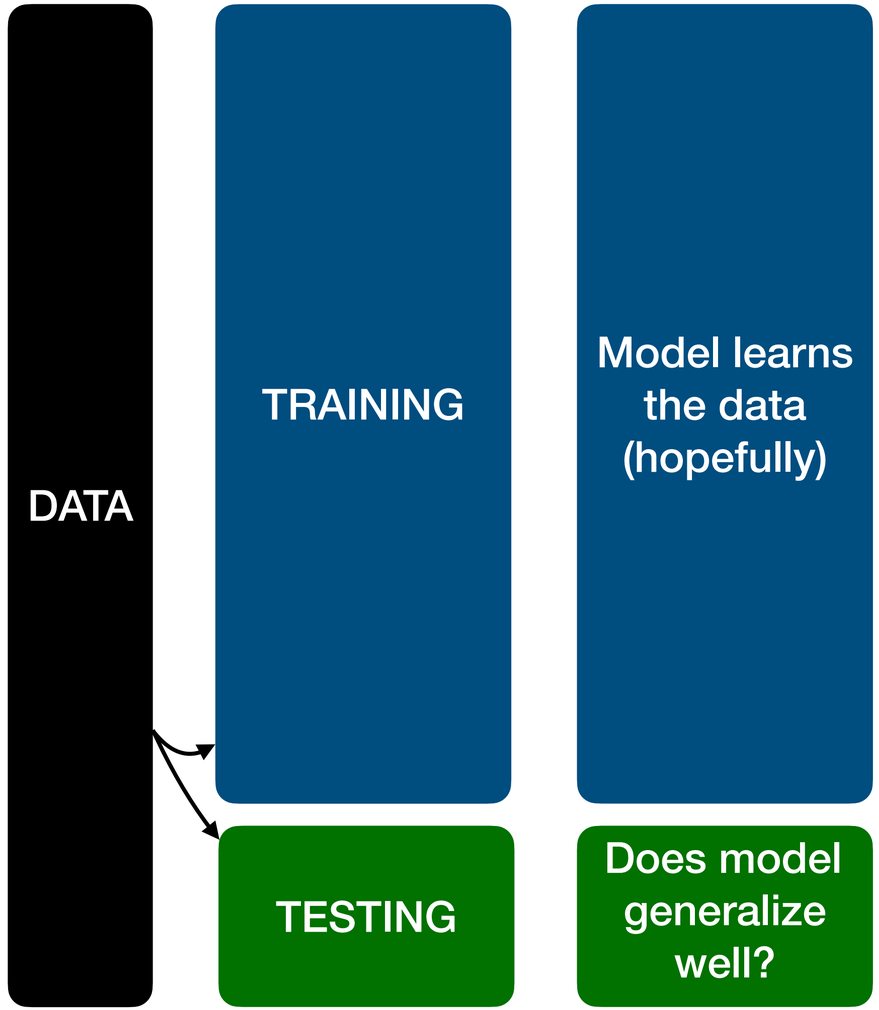
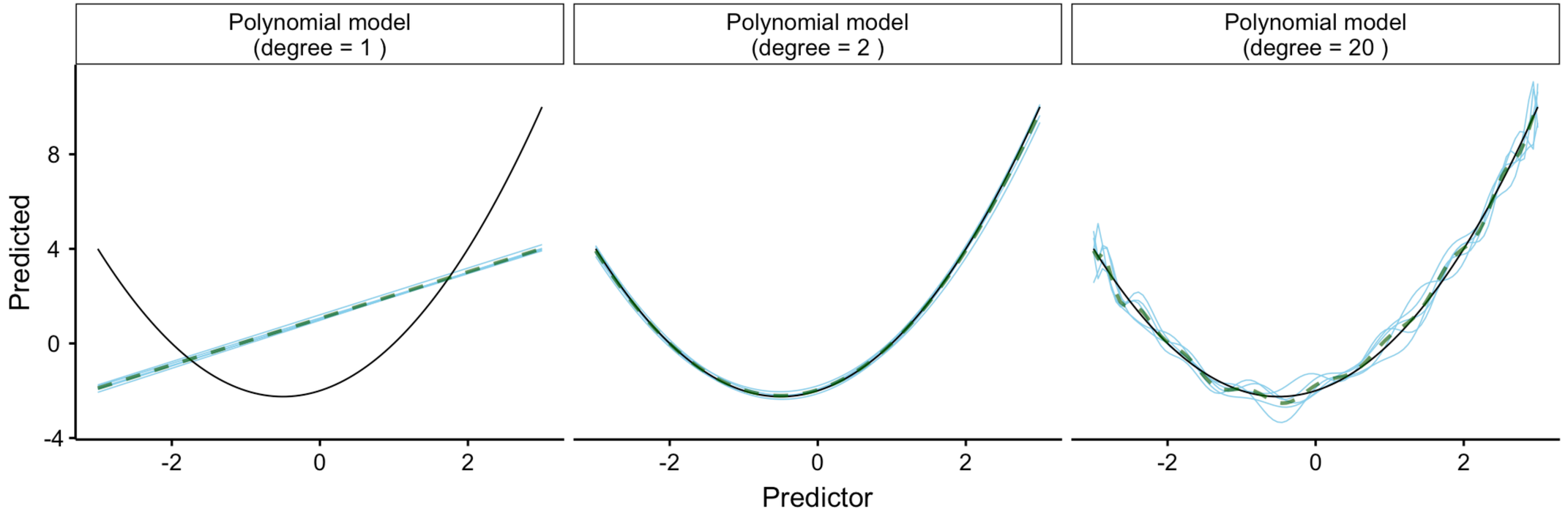


figure2

Bias, variance, and noise



1. True model: $Y_{TRUE} = X^2 + X - 2$
2. Observe sample: $Y_{OBS} = Y +$ noise, noise $\sim N(0,1)$
3. Obtain predicted: $Y_{PRED} = \hat{Y}$ based on fitted model

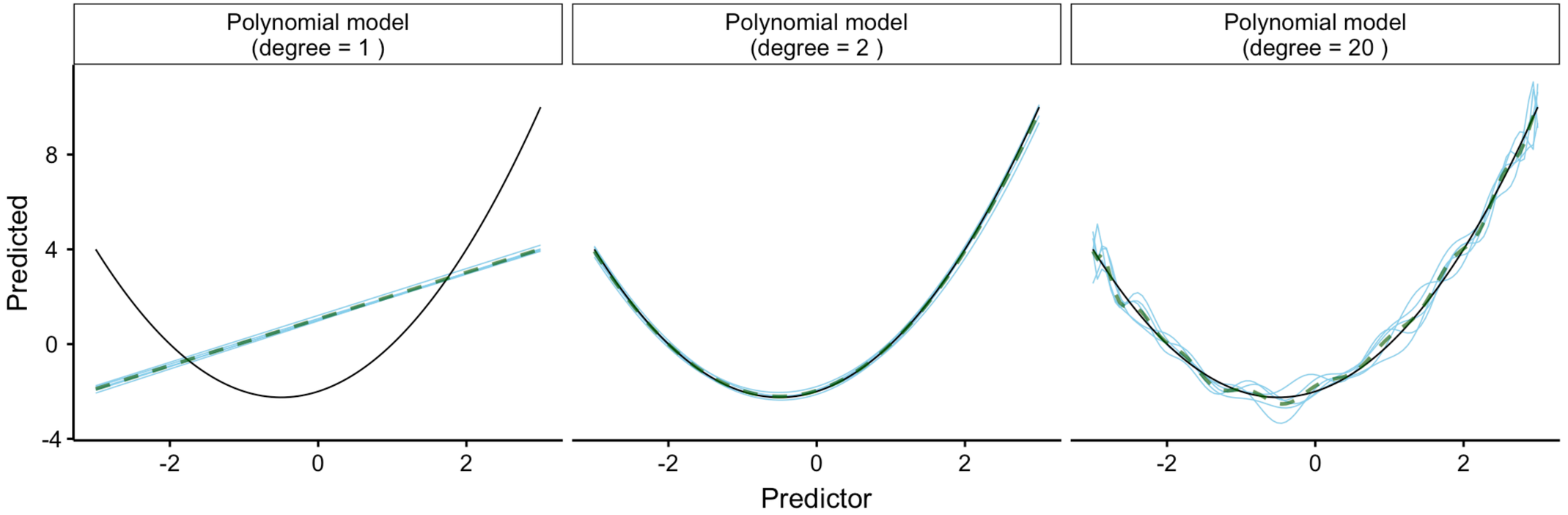
Repeat 5 times
Plot Y_{PRED}

Average over replicates
$Ave(Y_{PRED})$

Bias, variance, and noise

Repeat 5 times
Plot Y_{PRED}

Average over replicates
$Ave(Y_{PRED})$



1. **Variance:** how much the predicted target values from models trained using different samples that vary from each other
2. **Bias:** how much, on average, the predicted target values differ from the true model
3. **Noise:** how much the sample target values differ from the true model

Bias, variance, and noise

Can use parametric/nonparametric models

True model: $Y = f(X) + \epsilon = E(Y \mid X = x) + \epsilon$

1. Fit model training set *Tr*: $\hat{f}(x)$
2. We pick an observation (x_0, y_0) from test set *Ts*.

$$MSE(x_0, y_0) = E\left(y_0 - \hat{f}(x_0)\right)^2 = V\left(\hat{f}(x_0)\right) + \text{Bias}^2\left(\hat{f}(x_0)\right) + V(\epsilon)$$

1. **Variance:** how much the predicted target values from models trained using different samples that vary from each other
2. **Bias:** how much, on average, the predicted target values differ from the true model
3. **Noise:** how much the sample target values differ from the true model

Two Perspectives on Error:

The Measurement vs. The Cause

1. Train/test error is what we measure to see if a model performance is good.

2. Bias-variance decomposition explains why the error is what it is.

Beyond a Single Split: The Power of Resampling

Questions around generalizability…

- Entire analysis is based on the one specific sample of data we happened to collect.
    - How reliable is our model?
    - How certain are we about our model's findings?
    - How significant is the pattern we found?

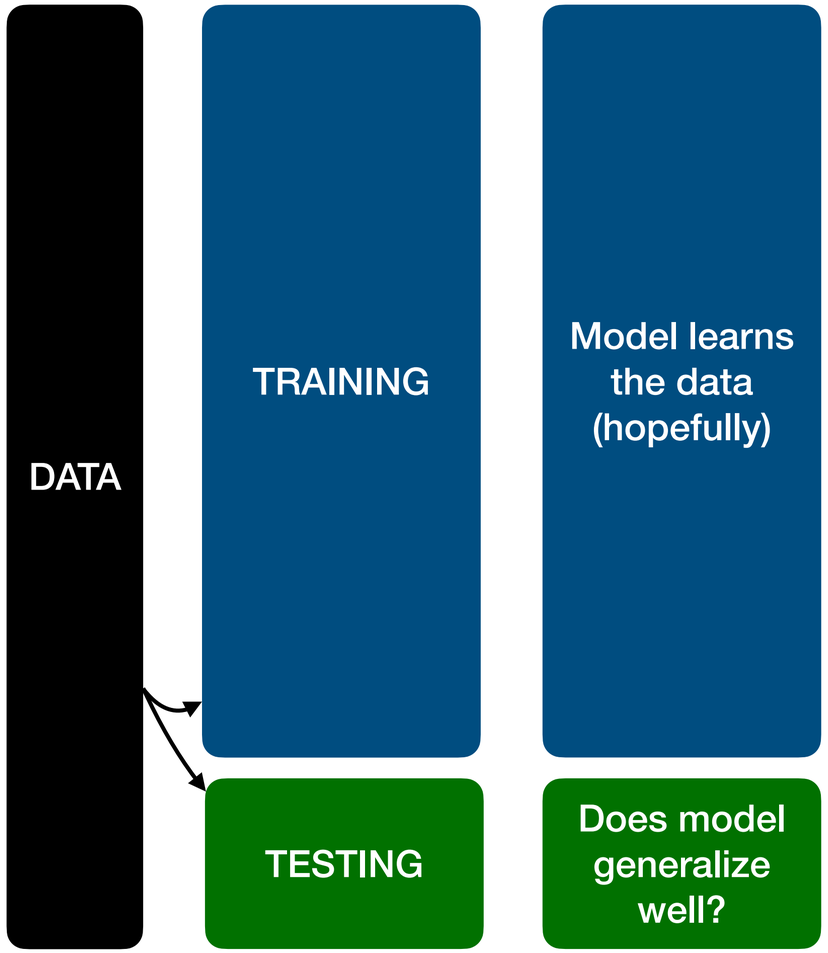
While traditional statistical formulas can sometimes provide answers, a more powerful and flexible approach is to use resampling methods.

- The core idea: we repeatedly create new, simulated datasets from our original data sample.
    - Estimate model reliability: k-fold cross-validation
    - Quantify the uncertainty of our estimates: bootstrap.
    - Test for statistical significance: permutation tests.

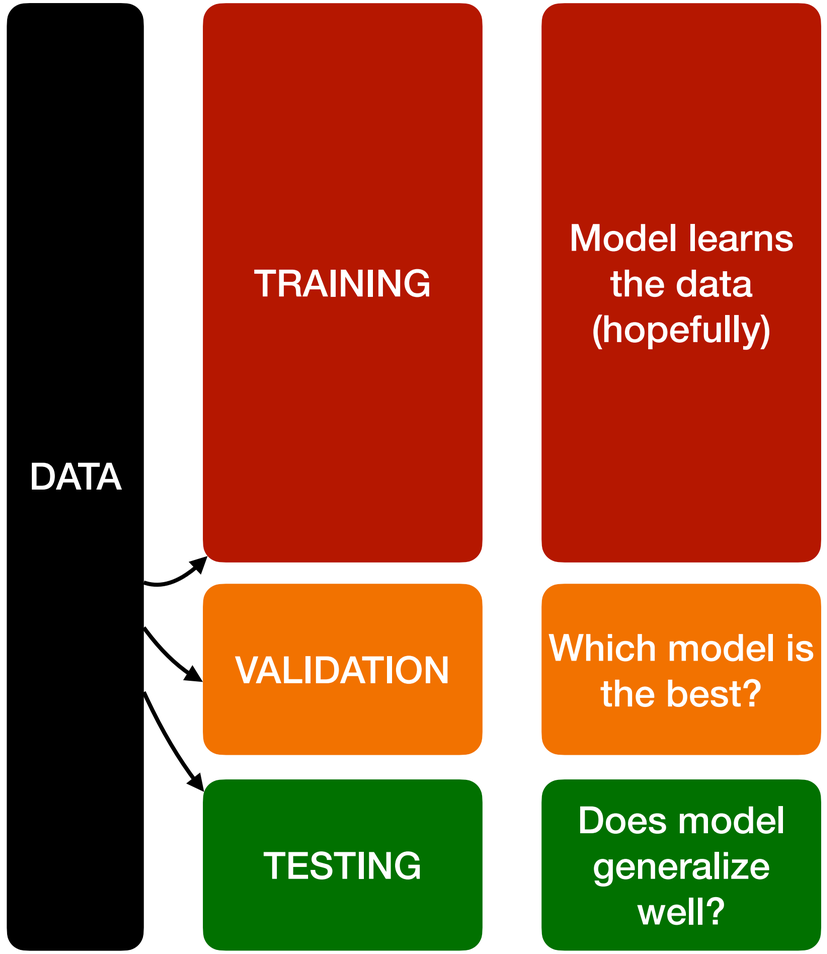
Three techniques are cornerstones of modern machine learning and statistics because they allow us to use our own data to understand the reliability, certainty, and significance of our findings.

Cross-validation

Finding a balance between over- and under-fitted models



- Training error often underestimates test error as model complexity increases.
- In CV, we consider a class of methods that estimate the test error by holding out a validation subset of the training data during fitting, and then applying the model to the validation set.



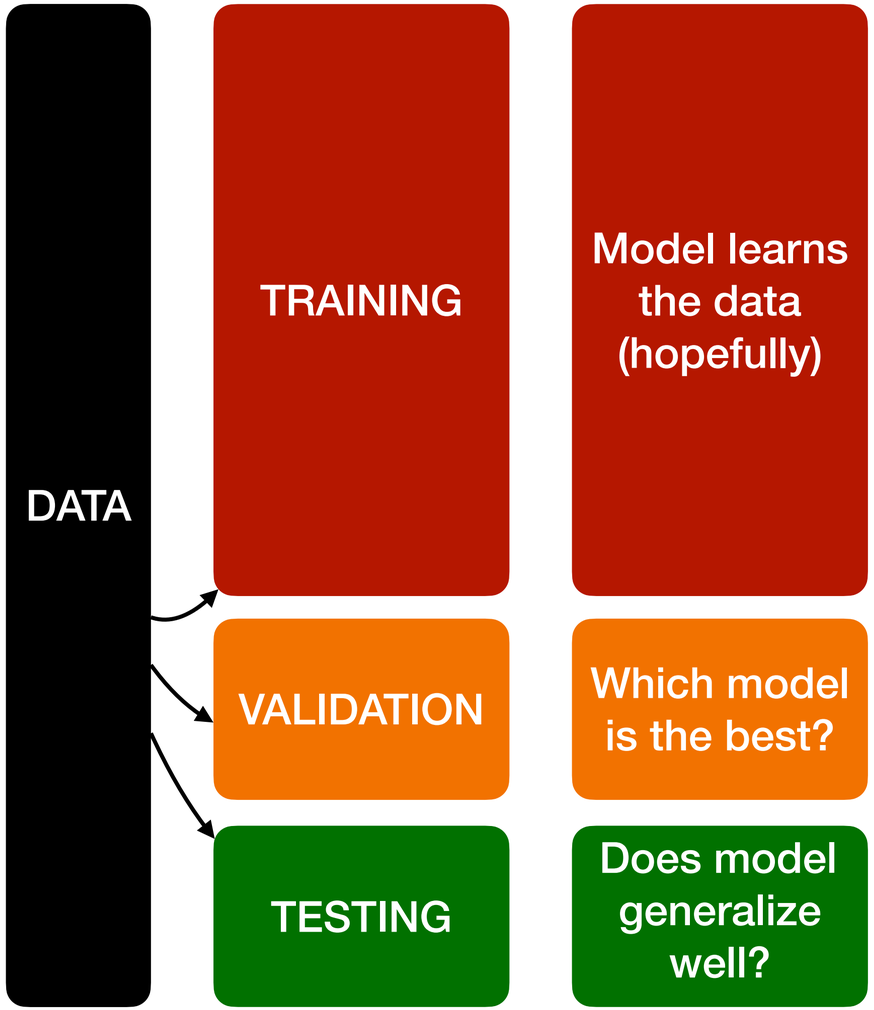
The solution: splitting our data (once)

"Sweet spot" between under-fitting and over-fitting ?

1. **Training Set:** The largest portion of your data.
   - Used to fit the model's parameters (e.g., the coefficients in a regression).

2. **Validation Set:** A separate portion of the data.
   - Used to tune the model's **hyperparameters** (e.g., the polynomial degree) and choose the best model **architecture**.
   - We evaluate different models on this set and choose the one that performs best.

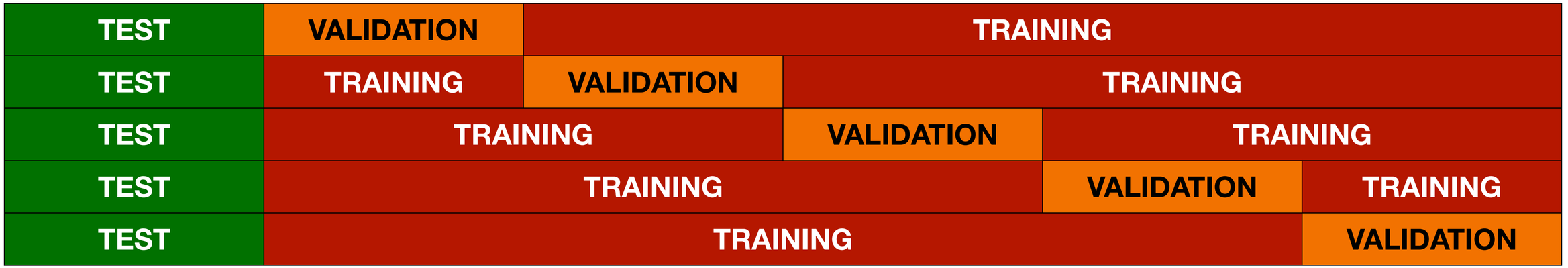
3. **Test Set:** Held back until the very end. Used only once to get an idea of the chosen model's performance on **unseen** data.



Splitting data into a single training/validation set can be problematic!
Model performance on the validation/training set might depend heavily on which data points are in it.

Better solution: splitting our data (multiple times)

k-fold cross-validation is a more robust approach!



1. Set aside the **TEST** data.
2. Split the remaining data into k equal-sized "folds" ($k = 5$ in the example above).
3. For each fold:
    1. Treat the fold as a temporary **VALIDATION** set.
    2. Fit the model on the other $k - 1$ **TRAINING** folds.
    3. Evaluate model on the **VALIDATION** fold through performance score $PS(k)$.
4. Average the performance scores from all k folds.

Must ensure that the test set is representative of the whole dataset!

Avoid selection bias! Try to ensure the distribution of target values in the test set is similar to the training set.

$$CV_K := \sum_{k=1}^{k} \frac{n_k}{n} PS(k)$$

k-fold cross-validation - some more comments

1. **Leave-one-out cross-validation (LOO-CV):**
   1. An extreme case is to take $K = n$.
   2. Each time, an observation is left as validation set and the remaining $n - 1$ observations are used to train the model.

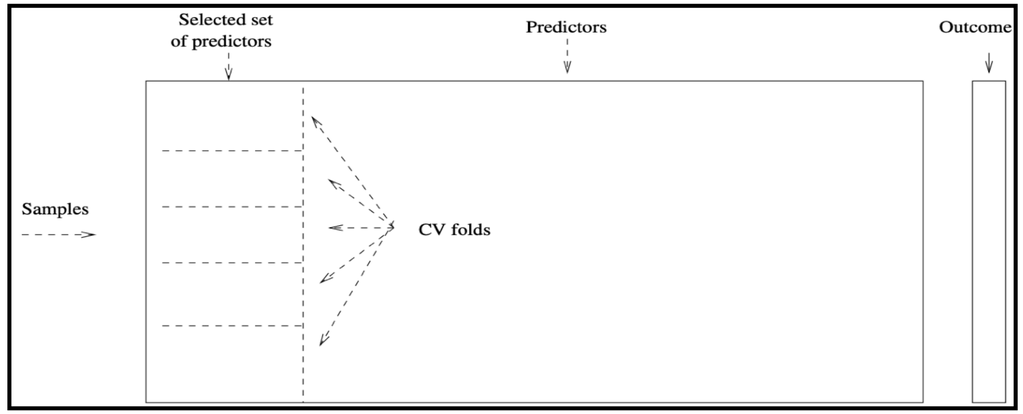
2. When K is small, the resulting performance score $CV(K)$ may be an overestimate of the underlying model error.
3. People have been *incredibly* careless with k-CV…..but, we won't.

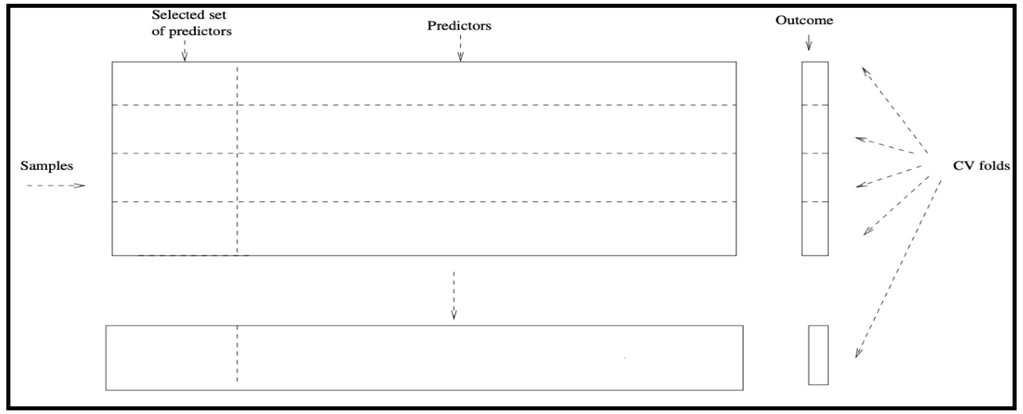
Consider a simple classifier applied to some two-class data:

1. Starting with 5000 predictors and 50 samples, find the 100 predictors having the largest correlation with the class labels.
2. We then apply a classifier such as logistic regression, using only these 100 predictors.

How do we estimate $CV(k)$?

Option 1: Apply cross-validation in step 2.
Option 2: Apply cross-validation to steps 1 and 2.





The bootstrap

Introduction

The bootstrap is a flexible and powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method.

Question: How certain are we about a calculated value, like a model coefficient?

Resampling Method Answer: We simulate new data by repeatedly drawing samples of the same size with replacement from our original data.

For n observations, we randomly sample n subjects with replacement to obtain a new "bootstrap dataset". Each bootstrap dataset is a "pseudo-observed" dataset. The procedure is often repeated for B times to obtain B bootstrap datasets.

The bootstrap

Schematic

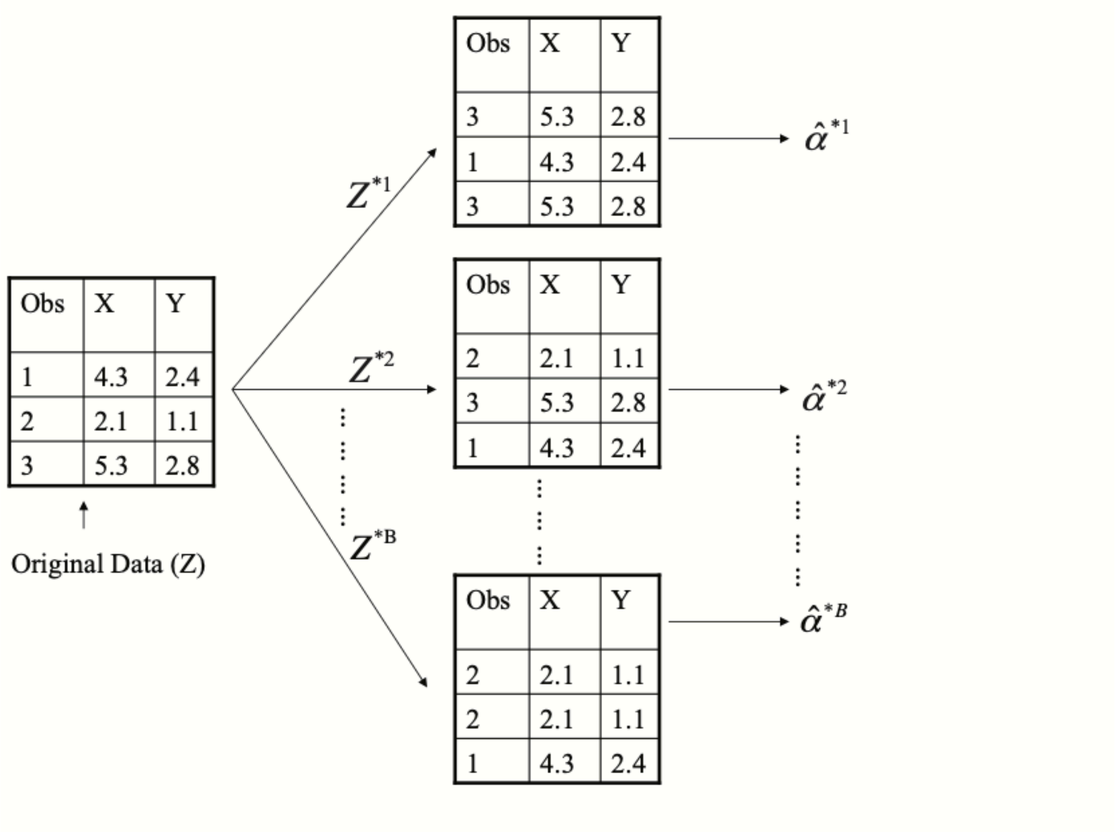
Denoting the first bootstrap data set by $Z^{\star 1}$, we use $Z^{\star 1}$ to produce a new bootstrap estimate for α, which we call $\hat{\alpha}^{\star 1}$

This procedure is repeated B times for some large value of B (say 100 or 1000), to produce

1. B bootstrap data sets, $Z^{\star 1}, Z^{\star 2}, \ldots, Z^{\star B}$
2. B α estimates, $\hat{\alpha}^{\star 1}, \hat{\alpha}^{\star 2}, \ldots, \hat{\alpha}^{\star B}$

We estimate the standard error of these bootstrap estimates using the formula

$$SE_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\hat{\alpha}^{\star b} - \bar{\hat{\alpha}}^{\star})^2}$$



A graphical illustration of the bootstrap approach on a small sample containing $n = 3$ observations.

1. Each bootstrap data set contains n observations, sampled with replacement from the original data set.
2. Each bootstrap data set is used to obtain an estimate of α

figure3

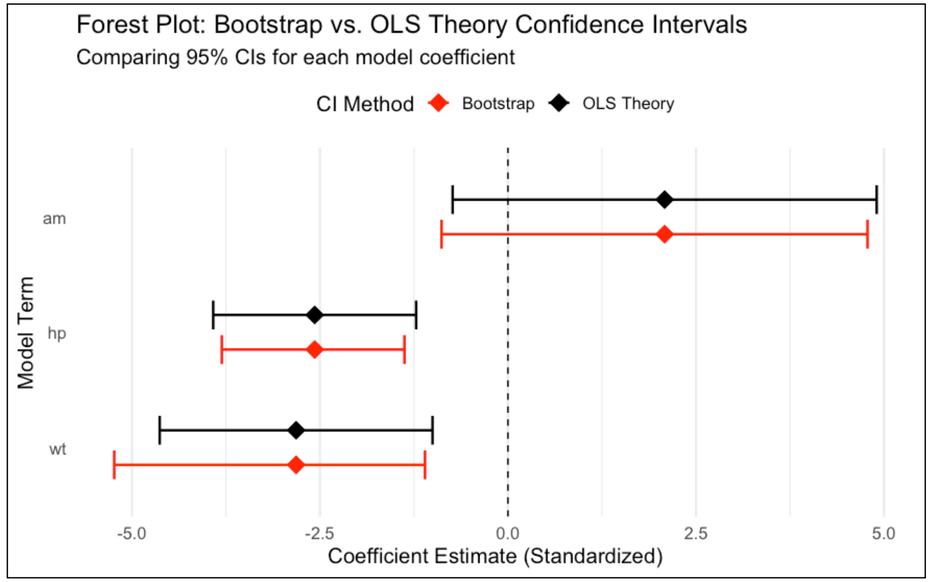
The bootstrap

An example

We used the `mtcars` dataset to estimate how a car's weight, horsepower, and transmission type affect its fuel efficiency and to quantify the uncertainty of these estimates.

We compared confidence intervals from

1. Standard OLS theory, which relies on regression model assumptions.
2. Bootstrapping, a resampling method, which relies on bootstrap consistency assumptions (?).



The bootstrap

When to NOT bootstrap

A critical assumption of the bootstrap procedure is that the observations are **independent and identically distributed**.

The method of sampling individual observations with replacement only makes sense if the order and grouping of the data don't matter.

This assumption is violated in several common data types:

1. Time Series Data
2. Longitudinal or Hierarchical Data

For these cases, **modified bootstrap procedures exist** (*block bootstrap for time series data*). Instead of sampling individual observations, we sample blocks of consecutive observations to preserve the time-dependent structure.

Bootstrap for error estimation?

Out-of-Bag (OOB) Estimation: When you draw a bootstrap sample, on average, only about **63.2%** of the original observations are included. The remaining 36.8% that are left out are called the "out-of-bag" (OOB) observations. We can use these OOB samples as a "test set" for the model trained on the main bootstrap sample.

This OOB error estimate tends to be pessimistic (too high). Why? Because the model was trained on only ~63% of the unique data points, which is smaller than the full dataset, and models trained on less data are typically worse.

The .632 Bootstrap Solution: A popular fix is the .632 bootstrap, which tries to correct for this pessimistic bias. It calculates the final error estimate as a weighted average:

$$E_{0.632} = 0.368 \times \text{Training error} + 0.632 \times \text{OOB error}$$

k-fold cross-validation is still preferred for estimating prediction error.

Permutation tests

Introduction: The Problem with Standard Tests

Imagine we're comparing the heart activity (measured by some score) between two groups: one that meditates and one that doesn't. We collect data and find:

1. Meditators (Group A): {10, 12, 15, 17, 100}
2. Non-Meditators (Group B): {11, 13, 14, 16, 18}

A standard t-test assumes our data (or residuals) are normally distributed. ***But what if one data point is an extreme outlier, like the 100 in Group A****? This can severely distort the mean and inflate the variance, making the t-test's p-value unreliable, potentially leading to a false conclusion of "no significant difference" or vice-versa.*

Permutation tests

Technical details

The Core Idea: Exchangeability

The fundamental principle behind permutation tests is exchangeability under the null hypothesis.

Null Hypothesis (H_0): There is no real difference between the groups or no association between variables.

Exchangeability: If H_0 is true, then the observed data points for the groups are exchangeable. This means that any random arrangement (permutation) of the data points among the groups is just as likely as the arrangement we actually observed.

How It Works

1. **Calculate Observed Statistic:** Compute a test statistic (e.g., difference in means, t-statistic, correlation) from your original data.
2. **Permute Labels:** Randomly shuffle (permute) the group labels among all observations. This breaks any true association while preserving the overall distribution of the data.
3. **Calculate Permuted Statistic:** Compute the same test statistic for this new, permuted dataset.
4. **Repeat:** Repeat steps 2 and 3 thousands of times (e.g., 10,000 times) to build a null distribution of the test statistic.
5. **Calculate P-value:** The p-value is the proportion of permuted test statistics that are as extreme as, or more extreme than, your original observed statistic.

This process allows us to derive a p-value without assuming a specific theoretical distribution for the data.

figure4

Permutation tests

An example

Let's use the `chickwts` dataset, which contains the weight of chicks fed different diets.

Some feed types might lead to non-normal weight distributions, making standard ANOVA p-values less trustworthy.

We'll compare two specific feed types: *casein* and *meatmeal*.