Blocking

STAT 6160 • Experimental Design • Fall 2026

Siddharth Vishwanath

University of Virginia

§1. A fitness tracker trial

The question

A company is replacing its fitness band. The **new** model counts steps with a different algorithm from the **old** one.

Does the new band record more steps per day than the old one?

- 20 subjects have volunteered.
- Response: average steps recorded per day over two weeks.

How would you design this experiment?

Design 1: Randomly assign 20 subjects: ten wear the new band, ten wear the old one.

	Old	New
Subjects	10	10
Bands worn	10	10

Design 2: 10 subjects, each wearing both bands.

Subject	1	2	⋯	10
Left wrist	old	old	⋯	old
Right wrist	new	new	⋯	new

Design 3: 10 subjects, each wearing both bands.

Subject	1	2	3	4	⋯	10
Left wrist	old	old	new	old	⋯	old
Right wrist	new	new	old	new	⋯	new

Questions

- What differs between the two groups of subjects?
- What comparisons do the designs make?
- What could go wrong?
- What does the coin flip within each subject protect against?

The data

Ten subjects, collected under design 3. Steps per day.

Subject	1	2	3	4	5	6	7	8	9	10
Old	8151	9967	9374	9178	11609	5539	6784	5652	8203	10761
New	8968	10878	10574	10322	11856	6611	7568	5473	8215	10608

Simulated data. The process that generated it is in §2, so we know the right answer.

Two analyses

Two-sample t test, treating old and new as independent groups of 10:

$$t = 0.63, \qquad \text{df} = 18, \qquad p\text{-value} = 0.54$$

Paired t test on the 10 differences $d_j = y_{\text{new},j} - y_{\text{old},j}$:

$$t = 3.38, \qquad \text{df} = 9, \qquad p\text{-value} = 0.0081$$

Which one is appropriate here? Why do they disagree?

§2. Why pairing helps

Two sources of trouble

In our simulated population:

- People differ greatly in how much they walk.
- The dominant wrist adds **400** recorded steps; 90% of subjects are right handed.
- The new algorithm adds **500** steps. This is what we want to estimate.

Predict: which design will give scattered estimates? Which will systematically overestimate the band effect?

Repeat each design

Simulate 4000 trials, each with 20 bands. Designs 2 and 3 use 10 subjects; design 1 uses 20.

```
import numpy as np
import matplotlib.pyplot as plt
rng = np.random.default_rng(4160)
R, n = 4000, 10
orange, blue = "#E57200", "#232D4B"

# Within-person differences: subject activity cancels.
noise = rng.normal(0, 200, (R, n, 2))
error = noise[:, :, 1] - noise[:, :, 0]
right = rng.random((R, n)) < 0.9
new_dominant = rng.random((R, n)) < 0.5
fixed = (500 + 400 * (2 * right - 1) + error).mean(axis=1)
paired = (500 + 400 * (2 * new_dominant - 1) + error).mean(axis=1)
```

Separate groups retain subject variation

In design 1, everyone wears the band on the non-dominant wrist.

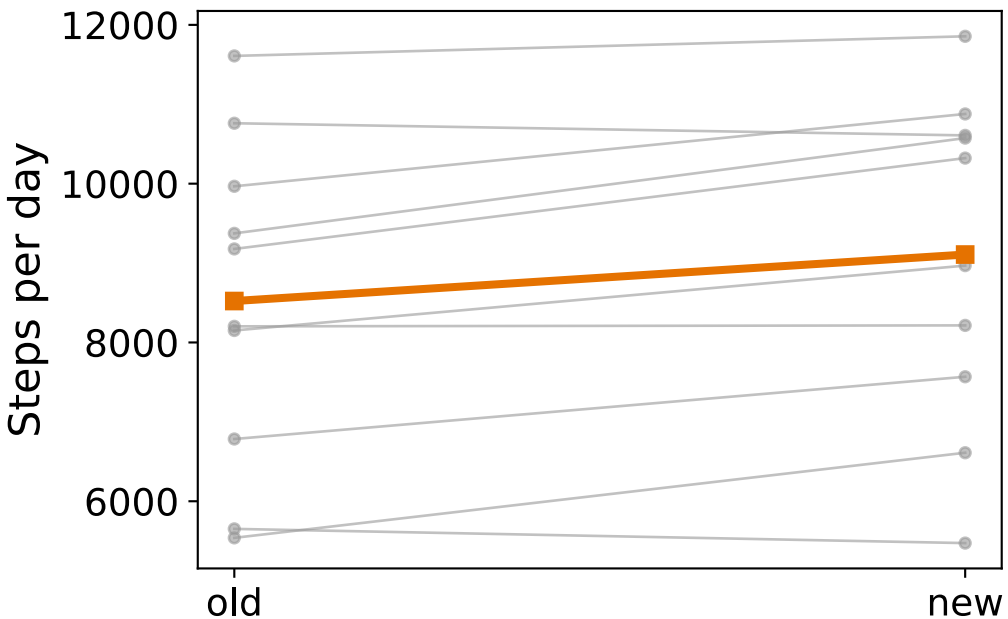
```
activity = rng.normal(8000, 2000, (R, 2 * n))
noise = rng.normal(0, 200, (R, 2 * n))
y = activity + noise
crd_est = 500 + y[:, n:].mean(axis=1) - y[:, :n].mean(axis=1)

estimates = [crd_est, fixed, paired]
labels = ["1: separate groups", "2: fixed wrists", "3: randomized wrists"]
```

An unusually active group can make either band look better.

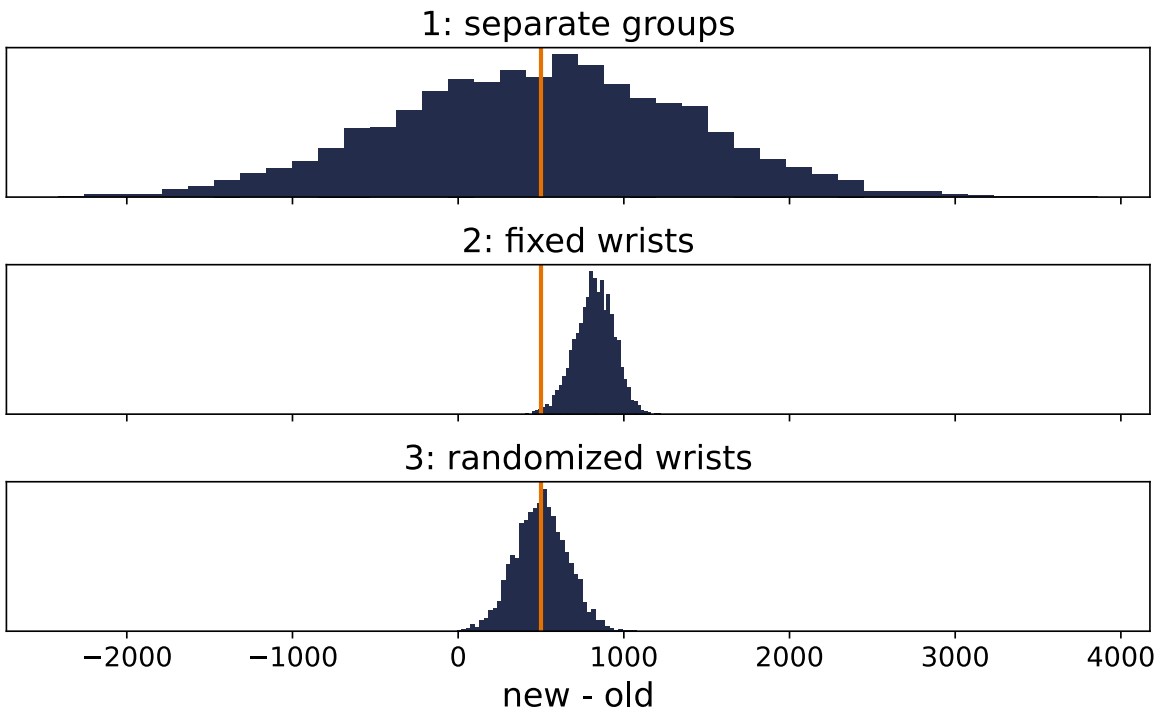
Compare within each subject

The same ten subjects from §1: one gray line per subject; orange shows the means.



Spread and bias are different

```
fig, axes = plt.subplots(3, 1,
    sharex=True, figsize=(8, 5))
for ax, values, label in zip(
        axes, estimates, labels):
    ax.hist(values, bins=40,
            color=blue)
    ax.axvline(500, color=orange,
               linewidth=2)
    ax.set_title(label, fontsize=16)
    ax.set(yticks=[])
    ax.tick_params(labelsize=13)
axes[-1].set_xlabel("new - old",
                    fontsize=16)
fig.tight_layout()
plt.show()
```



The orange line is the true band effect: **500 steps**. All panels use the same horizontal scale.

What each design buys us

Design	What happens over repeated trials?
Separate groups	Centred at **500**, wide spread.
Paired, fixed wrists	Centred at **820**, narrow spread: biased.
Paired, randomized wrists	Centred at **500**, narrow spread.

- **Pairing** removes differences in subject activity from the comparison.
- **Randomization** prevents a systematic wrist advantage; chance imbalance remains.
- The tradeoff: 10 pairs give 9 error degrees of freedom, versus 18 for two independent groups of 10.

§3. Pairing

Pairs

A **pair** is two experimental units that are expected to respond alike apart from the treatment.

- Both treatments appear in every pair.
- Which unit in the pair gets which treatment is randomized, independently across pairs.

Examples:

- the two wrists of one subject
- twins, or two littermates
- the two feet of one boy
- two halves of one field

Your turn: shoe soles

A manufacturer makes boys' shoes with sole material **A**. Material **B** is cheaper, and they want to know whether it wears faster. Ten boys are available for a few months.

- What is the pair?
- What gets randomized, and what is the randomization protecting against?
- How many boys would the unpaired version of this experiment need?

Your turn: a reading program

A district wants to compare a new reading program with the current one. 40 students are available. Unlike wrists and feet, students do not come in natural pairs, but every student has a reading score from last spring.

- How would you build the pairs?
- What gets randomized?
- What goes wrong if you pair on this year's score instead of last spring's?

Notation

y_{ij} = response to treatment i in pair j, $\quad i = 1, 2, \quad j = 1, \ldots, n$

	pair 1	pair 2	$\cdots$	pair n	mean
treatment 1	y_{11}	y_{12}	$\cdots$	y_{1n}	$\bar{y}_{1\bullet}$
treatment 2	y_{21}	y_{22}	$\cdots$	y_{2n}	$\bar{y}_{2\bullet}$
mean	$\bar{y}_{\bullet 1}$	$\bar{y}_{\bullet 2}$	$\cdots$	$\bar{y}_{\bullet n}$	$\bar{y}_{\bullet\bullet}$

- $N = 2n$ observations, one per cell.
- Tracker trial: $n = 10$ and $N = 20$.

The paired model

$$y_{ij} = \mu + \tau_i + \beta_j + \epsilon_{ij}, \qquad \epsilon_{ij} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

$$\sum_{i=1}^{2} \tau_i = 0, \qquad \sum_{j=1}^{n} \beta_j = 0$$

- μ :
- τ_i :
- β_j :
- ϵ_{ij} :

Assumptions

1. **Additivity.** The expected treatment difference is the same in every pair.
2. **Independence.** The errors ϵ_{ij} are independent.
3. **Constant variance.** $\text{Var}\,(\epsilon_{ij}) = \sigma^2$ for every treatment and pair.
4. **Normality.** ϵ_{ij} is normally distributed.

This is an idealized model. Randomizing wrists removes systematic bias, but wrist effects still add noise to the differences. The simulated differences need not be exactly normal.

§4. Blocking

A taste test

A company has four recipes for a granola bar and wants to know which are preferred.

- 8 tasters are available.
- Response: a rating from 0 to 10.

How would you design this?

One recipe per taster?

Give each taster one recipe: 8 tasters, two per recipe.

Taster	1	2	3	4	5	6	7	8
Recipe	R3	R1	R4	R2	R1	R3	R2	R4

- Some people rate everything a 2, some rate everything an 8.
- With one rating each, a generous taster and a good recipe look the same.

Same tasting order for everyone

Four tasting slots per taster, always in the order R1, R2, R3, R4.

Taster	slot 1	slot 2	slot 3	slot 4
1	R1	R2	R3	R4
2	R1	R2	R3	R4
⋮				
8	R1	R2	R3	R4

- Palates dull. The fourth bar is tasted by someone who has already eaten three.
- Slot and recipe now change together, so nothing in the data can separate them.
- This is design 2 of the tracker trial, with four treatments instead of two.

Randomize the tasting order

Four tasting slots per taster, in an order drawn independently for each taster.

Taster	slot 1	slot 2	slot 3	slot 4
1	R2	R4	R1	R3
2	R3	R1	R4	R2
⋮				
8	R1	R3	R2	R4

- Each taster is a **block** of four units.
- Each recipe is now equally likely to land in the tired fourth slot.

Blocks

A **block** is a set of g experimental units that are expected to respond alike apart from the treatment.

- Every treatment appears once in every block.
- The assignment within a block is randomized, independently across blocks.

Examples:

- a taster, with one slot per recipe
- a field, split into g plots
- a batch, a plate, or a day's run of a machine
- a litter of g animals

Complete blocks

Every treatment appears in every block, so the design is **complete**: the randomized complete block design, or RCBD.

- $N = gb$ observations, one per (treatment, block) cell.
- It needs a block big enough to hold all g treatments.

That is not always possible.

- A plate holds 4 samples but there are 7 treatments.
- A taster can manage 3 bars before the palate is gone, but there are 6 recipes.

- Then you need an **incomplete** block design, where each block holds only some of the treatments. Time permitting, later in the course.

Pairing is blocking with $g = 2$

	Pairing	Blocking
Treatments	2	g
Units per group	2	g
Groups	n pairs	b blocks
Observations	$N = 2n$	$N = gb$
Randomize	a coin flip per pair	a permutation per block
Compare within	a pair	a block

- Everything from here on covers both. Set $g = 2$ to get pairing back.

A second example: the variety trial

Four wheat varieties are to be compared on a field that slopes from a dry ridge down to a wet bottom.

- Yield depends far more on where a plot sits on the slope than on which variety is sown.
- Divide the field into strips running **across** the slope. Within a strip the plots are alike: same soil, same drainage, same exposure.
- Each strip is a block of four plots, one per variety.

- Choosing the blocks is a judgement about the units. Assigning within them is randomization.

Randomizing within blocks

	plot 1	plot 2	plot 3	plot 4
Strip I (ridge)	C	A	D	B
Strip II	B	D	A	C
Strip III	A	C	B	D
Strip IV (bottom)	D	B	C	A

- One independent permutation per strip, not one permutation reused down the field.
- Reusing one permutation ties each variety to the same position in every strip. Any repeated position effect could then be confused with variety.

Randomizing within blocks

```
recipes = np.array(["R1", "R2",
                    "R3", "R4"])
rng = np.random.default_rng(6160)

order = np.array(
    [rng.permutation(recipes)
     for _ in range(8)])

print(pd.DataFrame(order,
      index=range(1, 9),
      columns=["slot 1", "slot 2",
               "slot 3", "slot 4"]))
```

```
  slot 1 slot 2 slot 3 slot 4
1     R2     R3     R1     R4
2     R1     R2     R3     R4
3     R3     R2     R4     R1
4     R3     R4     R2     R1
5     R4     R1     R3     R2
6     R4     R1     R2     R3
7     R1     R3     R2     R4
8     R1     R3     R4     R2
```

- The same two lines give the tasting orders and the field layout.

Why include blocks?

Tasters differ in generosity. We account for that to compare **recipes** more precisely.

- Our question concerns recipe differences, so the treatment test is the focus.
- A block F test is possible under the fixed-effects model, but it answers a different question: do the block means differ?
- Choose blocks before seeing the outcomes. Keep them in the analysis because they were part of the design.

Your turn: an assay

A lab compares five protocols for an assay. Each run of the machine holds five samples, and a run takes a full day. Twenty-five samples are available.

- What is the block?
- How many blocks, and what gets randomized within one?
- What would go wrong if you ran all of protocol 1 on Monday and all of protocol 2 on Tuesday?

Your turn: back to the variety trial

Four wheat varieties, a field sloping from a dry ridge to a wet bottom.

- Should the strips run along the slope, or across it? What happens if you get it backwards?
- Suppose a fifth variety arrives and a strip still holds only four plots. What now?
- You have 16 plots either way. What has blocking bought you, and what has it cost?

§5. Parameters and estimates

Notation

$$y_{ij} = \text{response to treatment } \ i \ \text{ in block } \ j, \qquad i = 1, ..., g, \quad j = 1, ..., b$$

$$y_{ij} = \mu + \tau_i + \beta_j + \epsilon_{ij}, \qquad \epsilon_{ij} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2), \qquad \sum_i \tau_i = 0, \qquad \sum_j \beta_j = 0$$

Why sum to zero?

μ is the overall mean; treatment and block effects are departures from it.

- Adding a constant to every treatment effect and subtracting it from μ leaves every fitted mean unchanged.
- Setting each set of effects to sum to zero gives us one definite choice of labels.
- Treatment differences $\tau_i - \tau_k$ do not depend on that choice.

Taste test: what does a positive β_j say about taster j?

Two factors, additive effects

Two factors index the ratings: **recipe** and **taster**.

$$y_{ij} = \mu + \tau_i + \beta_j + \epsilon_{ij}$$

- A generous taster shifts all recipe ratings upward.
- The expected difference between two recipes is the same for every taster.
- This is **additivity**. Mean profiles across recipes would be parallel.

What if one taster loves nuts and another dislikes them?

Recipe-by-taster interaction

That is a **recipe-by-taster interaction**:

$$y_{ij} = \mu + \tau_i + \beta_j + \gamma_{ij} + \epsilon_{ij}$$

- An interaction changes recipe differences across tasters.
- We have only **one rating per recipe and taster**. Unrestricted interaction and random error cannot be estimated separately.
- The blocked F test here assumes additivity. Its residuals contain any departures from additivity as well as noise.

Independent replication within cells would allow a model with interaction and a separate error estimate.

Marginal means

$$\bar{y}_{i\bullet} = \frac{1}{b}\sum_j y_{ij}, \qquad \bar{y}_{\bullet j} = \frac{1}{g}\sum_i y_{ij}, \qquad \bar{y}_{\bullet\bullet} = \frac{1}{gb}\sum_{i,j} y_{ij}$$

Under the model:

$$\mathbb{E}(\bar{y}_{i\bullet}) = \qquad \mathbb{E}(\bar{y}_{\bullet j}) = \qquad \mathbb{E}(\bar{y}_{\bullet\bullet}) =$$

- $N = gb$ observations, one per cell. Taste test: $g = 4$, $b = 8$, $N = 32$.

Parameters and their estimates

Parameter	Estimate
μ	$\hat{\mu} = \bar{y}_{\bullet\bullet}$
τ_i	$\hat{\tau}_i = \bar{y}_{i\bullet} - \bar{y}_{\bullet\bullet}$
β_j	$\hat{\beta}_j = \bar{y}_{\bullet j} - \bar{y}_{\bullet\bullet}$
$\tau_i - \tau_k$	$\bar{y}_{i\bullet} - \bar{y}_{k\bullet}$
σ^2	$\mathrm{MS}_{\mathrm{Res}}$

Fitted values and residuals:

$$\hat{y}_{ij} = \bar{y}_{i\bullet} + \bar{y}_{\bullet j} - \bar{y}_{\bullet\bullet}, \qquad e_{ij} = y_{ij} - \bar{y}_{i\bullet} - \bar{y}_{\bullet j} + \bar{y}_{\bullet\bullet}$$

Block effects cancel

$$\text{Var}\,(\bar{y}_{i\bullet} - \bar{y}_{k\bullet}) = \frac{2\sigma^2}{b}$$

- The block effects cancel from every treatment difference, because each β_j appears once in $\bar{y}_{i\bullet}$ and once in $\bar{y}_{k\bullet}$.
- That is the whole point of the design, and it is why the design is called complete.
- Taste test: what happens if taster 5 skips recipe R2?

§6. The design matrix

One row, one rating

Use a small taste test: **2 recipes, 3 tasters**. List all ratings for R1, then all ratings for R2.

$$\begin{pmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{21} \\ y_{22} \\ y_{23} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}}_{X_{\text{ind}}} \begin{pmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} + \epsilon$$

Each row selects **overall level + recipe effect + taster effect**.

- Read the first row: $y_{11} = \mu + \tau_1 + \beta_1 + \epsilon_{11}$.
- What does the last row say?

Why are there redundant columns?

As in one-way ANOVA, the recipe indicators add up to the intercept column. Now the taster indicators do too.

$$\text{R1 column} + \text{R2 column} = \mathbf{1}_6$$

$$\text{taster 1 column} + \text{taster 2 column} + \text{taster 3 column} = \mathbf{1}_6$$

- Six columns, but only **four independent columns**.
- In general: $1 + g + b$ columns, rank $g + b - 1$.
- Two familiar choices: omit one indicator for each factor, or impose sum-to-zero constraints.

Sum-to-zero coding

For the same two recipes and three tasters, set $\tau_2 = -\tau_1$ and $\beta_3 = -\beta_1 - \beta_2$.

$$X_S = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 0 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & -1 & -1 \end{pmatrix}, \qquad \boldsymbol{\theta}_S = \begin{pmatrix} \mu \\ \tau_1 \\ \beta_1 \\ \beta_2 \end{pmatrix}.$$

- Same six expected ratings; only four free coefficients.
- Last row: $\mu - \tau_1 - \beta_1 - \beta_2 = \mu + \tau_2 + \beta_3$.

Same model, different coordinates

Recall the one-way argument. Let X have **full column rank**, and let $\tilde{X} = XC$ with C invertible.

$$\begin{aligned}\widetilde{H} &= XC\,(C^\top X^\top XC)^{-1}C^\top X^\top \\ &= XC\,C^{-1}(X^\top X)^{-1}C^{-\top}C^\top X^\top \\ &= X(X^\top X)^{-1}X^\top = H.\end{aligned}$$

Reference coding and sum-to-zero coding describe the same additive model: **same H, fitted values, residual SS, and treatment F test**.

- We can choose the coding that makes the calculation easiest.

Why not one mean per cell?

In one-way ANOVA, each treatment mean was estimated from several observations. Here, each recipe–taster cell has **only one rating**.

$$X_C = I_{gb} \quad \Rightarrow \quad H_C = I_{gb}, \qquad \hat{y} = y, \qquad \mathrm{SS}_{\mathrm{Res}} = 0.$$

- A separate mean for every cell uses all gb degrees of freedom.
- The additive model shares information across cells and uses only $g + b - 1$.
- For our six ratings: **6 cell means versus 4 additive-model coefficients**.

Example

Six illustrative ratings: two recipes, each rated by three tasters.

```
import pandas as pd
import statsmodels.formula.api as smf

small = pd.DataFrame({
    "recipe": ["R1"] * 3 + ["R2"] * 3,
    "taster": [1, 2, 3] * 2,
    "y": [6, 8, 5, 7, 8.5, 6.5],
})

fit = smf.ols(
    "y ~ C(recipe, Sum) + C(taster, Sum)",
    data=small,
).fit()
```

- `Sum` requests sum-to-zero coding for each factor.

The same matrix in code

```
X = fit.model.exog
names = ["mean", "recipe",
         "taster1", "taster2"]
print(pd.DataFrame(X.astype(int),
      columns=names))
```

	mean	recipe	taster1	taster2
0	1	1	1	0
1	1	1	0	1
2	1	1	-1	-1
3	1	-1	1	0
4	1	-1	0	1
5	1	-1	-1	-1

$$X = (\mathbf{1} \;\; T \;\; B)$$

One column for the overall mean, one for recipe, and two for taster. From here on, X uses **sum-to-zero coding**.

Look at $X^\top X$

```
print(pd.DataFrame(X.T @ X,
      columns=names,
      index=names))
```

```
         mean  recipe  taster1  taster2
mean      6.0     0.0      0.0      0.0
recipe    0.0     6.0      0.0      0.0
taster1   0.0     0.0      4.0      2.0
taster2   0.0     0.0      2.0      4.0
```

- Circle the three diagonal blocks: **overall mean, recipe, taster**.
- The zeros between them mean that these groups of columns are orthogonal.
- The two taster columns need not be orthogonal to each other.

Why the zeros appear

Within every taster, the two recipe codes are $+1$ and -1. Their contributions cancel in $\boldsymbol{T}^\top \boldsymbol{B}$.

$$\mathbf{1}^\top \boldsymbol{T} = \mathbf{0}^\top, \qquad \mathbf{1}^\top \boldsymbol{B} = \mathbf{0}^\top, \qquad \boldsymbol{T}^\top \boldsymbol{B} = \mathbf{0}.$$

The same cancellation holds for g treatments and b complete blocks:

$$\boldsymbol{X}^\top \boldsymbol{X} = \begin{pmatrix} gb & \mathbf{0}^\top & \mathbf{0}^\top \\ \mathbf{0} & \boldsymbol{T}^\top \boldsymbol{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{B}^\top \boldsymbol{B} \end{pmatrix}.$$

- Every treatment appears equally often with every block.

The geometry of the Hat matrix

Let $X = [X_1 \mid X_2]$ have full column rank, with **orthogonal groups of columns**:

$$X_1^\top X_2 = \mathbf{0}.$$

The hat matrix for each group is $H_j = X_j(X_j^\top X_j)^{-1}X_j^\top, j = 1, 2.$

$$\boxed{H = X(X^\top X)^{-1}X^\top = H_1 + H_2}$$

- Projecting onto the combined space adds the two orthogonal projections.
- **On the next quiz:** show that $H = H_1 + H_2$ under these conditions.

The fitted response splits

Because those groups of columns are orthogonal, the hat matrix splits:

$$\boldsymbol{H} = \boldsymbol{H}_1 + \boldsymbol{H}_T + \boldsymbol{H}_B$$

$$\boldsymbol{H}_1 = \frac{\mathbf{1}\mathbf{1}^\top}{gb}, \qquad \boldsymbol{H}_T = \boldsymbol{T}(\boldsymbol{T}^\top\boldsymbol{T})^{-1}\boldsymbol{T}^\top, \qquad \boldsymbol{H}_B = \boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{B})^{-1}\boldsymbol{B}^\top.$$

$$\hat{\boldsymbol{y}} = \underbrace{\boldsymbol{H}_1\boldsymbol{y}}_{\text{overall mean}} + \underbrace{\boldsymbol{H}_T\boldsymbol{y}}_{\text{recipe departures}} + \underbrace{\boldsymbol{H}_B\boldsymbol{y}}_{\text{taster departures}}.$$

- This is the fitted-value formula from §5, written as three projections.

Counting the degrees of freedom

For a projection, **trace = rank = number of independent directions**.

Component	Contribution to observation (i, j)	df
Overall mean	$\bar{y}_{\bullet\bullet}$	1
Treatment	$\bar{y}_{i\bullet} - \bar{y}_{\bullet\bullet}$	$g - 1$
Block	$\bar{y}_{\bullet j} - \bar{y}_{\bullet\bullet}$	$b - 1$
Residual	$y_{ij} - \hat{y}_{ij}$	$(g-1)(b-1)$

Our six ratings: $1 + 1 + 2 + 2 = 6$.

- After removing the overall mean, the remaining components give the ANOVA table.

Checking the split in Python

```
def hat(M):
    return M @ np.linalg.solve(M.T @ M, M.T)

H = hat(X)
H1, HT, HB = hat(X[:, :1]), hat(X[:, 1:2]), hat(X[:, 2:])
parts = (H1, HT, HB, np.eye(6) - H)

print("H = H1 + HT + HB:", np.allclose(H, H1 + HT + HB))
print("HT @ HB = 0:", np.allclose(HT @ HB, 0))
print("Degrees of freedom:", [round(np.trace(M)) for M in parts])
```

```
H = H1 + HT + HB: True
HT @ HB = 0: True
Degrees of freedom: [1, 1, 2, 2]
```

- Same fitted response, separated into mean, recipe, and taster contributions.

§7. Sums of squares

Splitting $I - H_1$

$$\text{SS}_{\text{Tot}} = \sum_{i,j} (y_{ij} - \bar{y}_{\bullet\bullet})^2 = \boldsymbol{y}^\top (\boldsymbol{I} - \boldsymbol{H}_1)\boldsymbol{y}$$

The fitted response splits into an overall mean, a treatment effect, and a block effect:

$$\boldsymbol{H} = \boldsymbol{H}_1 + \boldsymbol{H}_T + \boldsymbol{H}_B$$

$$\boxed{\text{SS}_{\text{Tot}} = \underbrace{\boldsymbol{y}^\top \boldsymbol{H}_T \boldsymbol{y}}_{\text{SS}_{\text{Trt}}} + \underbrace{\boldsymbol{y}^\top \boldsymbol{H}_B \boldsymbol{y}}_{\text{SS}_{\text{Blk}}} + \underbrace{\boldsymbol{y}^\top (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y}}_{\text{SS}_{\text{Res}}}}$$

- Orthogonal components account for separate parts of the variation.

The same thing in means

$$y_{ij} - \bar{y}_{\bullet\bullet} = (\bar{y}_{i\bullet} - \bar{y}_{\bullet\bullet}) + (\bar{y}_{\bullet j} - \bar{y}_{\bullet\bullet}) + (y_{ij} - \bar{y}_{i\bullet} - \bar{y}_{\bullet j} + \bar{y}_{\bullet\bullet})$$

$$\underbrace{\sum_{i,j}(y_{ij} - \bar{y}_{\bullet\bullet})^2}_{\mathrm{SS_{Tot}}} = \underbrace{b\sum_{i}(\bar{y}_{i\bullet} - \bar{y}_{\bullet\bullet})^2}_{\mathrm{SS_{Trt}}} + \underbrace{g\sum_{j}(\bar{y}_{\bullet j} - \bar{y}_{\bullet\bullet})^2}_{\mathrm{SS_{Blk}}} + \underbrace{\sum_{i,j} e_{ij}^2}_{\mathrm{SS_{Res}}}$$

- Square and sum: the cross-products cancel in this balanced, complete layout.
- Taste test: variation between recipes + variation between tasters + what remains.

Where the residual df comes from

$$\mathrm{SS}_{\mathrm{Res}} = \sum_{i,j} (y_{ij} - \bar{y}_{i\bullet} - \bar{y}_{\bullet j} + \bar{y}_{\bullet\bullet})^2$$

- The gb residuals satisfy $\sum_i e_{ij} = 0$ for every block and $\sum_j e_{ij} = 0$ for every treatment.
- That is $g + b - 1$ independent constraints, leaving $(g-1)(b-1)$.
- With $g = 2$: $(g-1)(b-1) = b - 1$, the paired case.

§8. The ANOVA table

The ANOVA table

Write $\nu = (g-1)(b-1)$ for the residual degrees of freedom.

Source	df	SS	MS	F
Treatment	$g-1$	$\mathrm{SS}_{\mathrm{Trt}}$	$\mathrm{SS}_{\mathrm{Trt}}/(g-1)$	$\mathrm{MS}_{\mathrm{Trt}}/\mathrm{MS}_{\mathrm{Res}}$
Blocks	$b-1$	$\mathrm{SS}_{\mathrm{Blk}}$	$\mathrm{SS}_{\mathrm{Blk}}/(b-1)$	
Residual	ν	$\mathrm{SS}_{\mathrm{Res}}$	$\mathrm{SS}_{\mathrm{Res}}/\nu$	
Total	$gb-1$	$\mathrm{SS}_{\mathrm{Tot}}$		

- The df column is the list of block sizes from $\boldsymbol{X}^\top \boldsymbol{X}$, plus what is left over.
- We omit the block F here because our question is about treatments.

Expected mean squares

$$\mathbb{E}(\text{MS}_{\text{Trt}}) = \sigma^2 + \frac{b}{g-1}\sum_{i=1}^{g}\tau_i^2, \qquad \mathbb{E}(\text{MS}_{\text{Res}}) = \sigma^2$$

$$\mathbb{E}(\text{MS}_{\text{Blk}}) = \sigma^2 + \frac{g}{b-1}\sum_{j=1}^{b}\beta_j^2$$

- F compares a mean square that grows with $\sum_i \tau_i^2$ to one that does not.

One-way ANOVA versus blocking

	One-way	Two-way with blocks
Model	$\mu + \tau_i + \epsilon_{ij}$	$\mu + \tau_i + \beta_j + \epsilon_{ij}$
Design matrix	$(\mathbf{1} \mid \boldsymbol{T})$	$(\mathbf{1} \mid \boldsymbol{T} \mid \boldsymbol{B})$
$\mathrm{SS}_{\mathrm{Trt}}$	$b \sum_i (\bar{y}_{i\bullet} - \bar{y}_{\bullet\bullet})^2$	the same
Residual SS	$\mathrm{SS}_{\mathrm{Blk}} + \mathrm{SS}_{\mathrm{Res}}$	$\mathrm{SS}_{\mathrm{Res}}$
Residual df	$g(b-1)$	$(g-1)(b-1)$
$\mathrm{MS}_{\mathrm{Res}}$ estimates	σ^2 plus block spread	σ^2

- Adding $\boldsymbol{B}$ never changes $\mathrm{SS}_{\mathrm{Trt}}$, because $\boldsymbol{H}_T$ does not involve $\boldsymbol{B}$.
- It moves $\mathrm{SS}_{\mathrm{Blk}}$ out of the residual, and spends $b-1$ degrees of freedom to do it.

When does blocking pay off?

$$F_{\text{blocked}} = \frac{\text{SS}_{\text{Trt}}/(g-1)}{\text{SS}_{\text{Res}}/\nu}$$

$$F_{\text{one-way}} = \frac{\text{SS}_{\text{Trt}}/(g-1)}{(\,\text{SS}_{\text{Blk}} + \text{SS}_{\text{Res}}\,)\,/g(b-1)}$$

- The numerators are identical. Only the denominator changes.
- Blocking wins when SS_{Blk} is large relative to the $b-1$ degrees of freedom it costs.
- Weak block differences may not compensate for the lost error degrees of freedom. Choose useful blocks at the design stage; do not select the analysis by testing blocks.

§9. The F test

Quadratic forms

$$\boldsymbol{y} \sim \mathcal{N}_{gb}(\boldsymbol{X\theta}, \sigma^2 \boldsymbol{I})$$

For a projection matrix $\boldsymbol{H}_a$ of rank r_a:

$$\frac{\boldsymbol{y}^\top \boldsymbol{H}_a \boldsymbol{y}}{\sigma^2} \sim \chi^2_{r_a}(\lambda_a), \qquad \lambda_a = \frac{\|\boldsymbol{H}_a \boldsymbol{X\theta}\|^2}{\sigma^2}$$

- If $\boldsymbol{H}_a \boldsymbol{H}_c = \boldsymbol{0}$, the two quadratic forms are independent.
- $\lambda_a = 0$ when $\boldsymbol{H}_a \boldsymbol{X\theta} = \boldsymbol{0}$, and then the χ^2 is central.

Applying it

$$\frac{\mathrm{SS}_{\mathrm{Trt}}}{\sigma^2} \sim \chi^2_{g-1}, \qquad \frac{\mathrm{SS}_{\mathrm{Res}}}{\sigma^2} \sim \chi^2_{\nu},$$

- Independent, because $\boldsymbol{H}_T(\boldsymbol{I} - \boldsymbol{H}) = \boldsymbol{0}$.
- $(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{X}\boldsymbol{\theta} = \boldsymbol{0}$, so the residual χ^2 is always central.
- Under H_0 every $\tau_i = 0$ and $\lambda = 0$.

The F statistic

$$F = \frac{\mathrm{SS}_{\mathrm{Trt}}/(g-1)}{\mathrm{SS}_{\mathrm{Res}}/\nu}, \qquad \nu = (g-1)(b-1)$$

- Under H_0: $F \sim F_{g-1,\,\nu}$
- Under H_a: $F \sim F_{g-1,\,\nu}(\lambda)$
- Reject H_0 at level α when F exceeds the $1-\alpha$ quantile.

The paired case

With $g = 2$ the treatment sum of squares has one degree of freedom, and everything reduces to the b differences d_j:

$$\mathrm{SS}_{\mathrm{Trt}} = \frac{b}{2}\bar{d}^2, \qquad \mathrm{SS}_{\mathrm{Res}} = \frac{1}{2}\sum_j (d_j - \bar{d})^2, \qquad F = \left(\frac{\bar{d}}{s_d/\sqrt{b}}\right)^2 = t^2$$

- $t \sim t_{b-1}$ under H_0, and $t_{b-1}^2 \overset{d}{=} F_{1,b-1}$.
- The paired t test of §1 is the $g = 2$ case of the blocked F test.

§10. The taste test in Python

Simulating the ratings

```
import statsmodels.formula.api as smf
from statsmodels.stats.anova import anova_lm

recipes, g, b = ["R1", "R2", "R3", "R4"], 4, 8
tau = np.array([2.1, 4.0,  1.9, 2.05])       # recipe effects, sum to zero
rng = np.random.default_rng(2)

taster = rng.normal(0, 1.5, b)               # how generous each taster is
rating = 3 + np.repeat(tau, b) + np.tile(taster, g) + rng.normal(0, 0.6, g * b)

bars = pd.DataFrame({
    "recipe": np.repeat(recipes, b),
    "taster": np.tile(np.arange(1, b + 1), g),
    "rating": rating.round(1),
})
```

The ratings

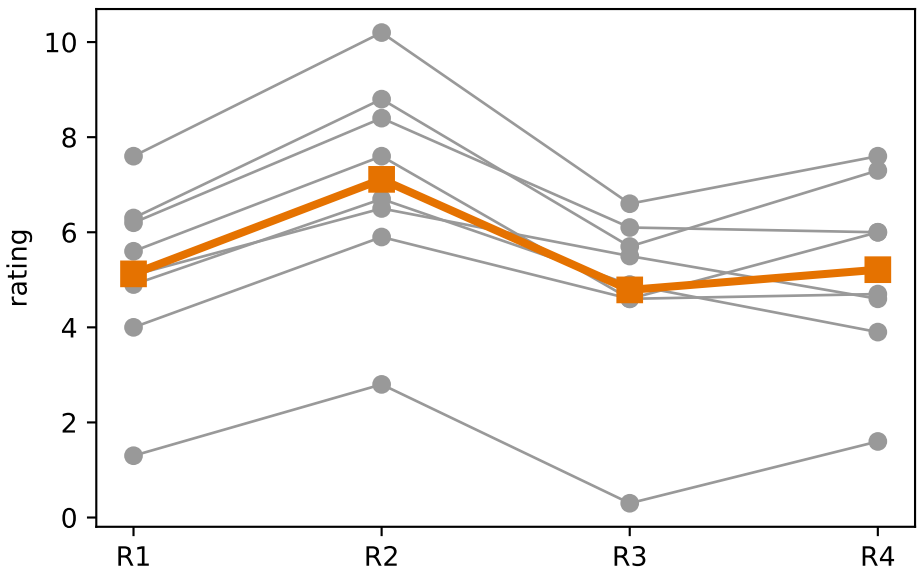
```
wide = bars.pivot(index="taster",
                  columns="recipe",
                  values="rating")
print(wide)
```

```
recipe    R1    R2   R3   R4
taster
1        5.6   7.6  4.6  6.0
2        4.0   5.9  4.6  4.7
3        5.1   6.5  5.5  4.6
4        1.3   2.8  0.3  1.6
5        7.6  10.2  6.6  7.6
6        6.3   8.8  5.7  7.3
7        4.9   6.7  4.9  3.9
8        6.2   8.4  6.1  6.0
```

- Taster 4 rates everything low, taster 5 rates everything high.
- Read down a column and the spread is enormous. Read across a row and it is not.

The ratings

```
fig, ax = plt.subplots()
for t, row in wide.iterrows():
    ax.plot(recipes, row, "o-",
            color="0.6", lw=1)
ax.plot(recipes, wide.mean(), "s-",
        color=orange, ms=9, lw=3)
ax.set(ylabel="rating")
plt.show()
```



Treating tasters as blocks

```
two_way = smf.ols(
    "rating ~ C(recipe, Sum) + C(taster, Sum)", data=bars
).fit()

table = anova_lm(two_way)
table.index = ["recipe", "taster", "Residual"]
print(table.round(4))
```

```
             df    sum_sq  mean_sq        F  PR(>F)
recipe      3.0   26.5359   8.8453  31.7168     0.0
taster      7.0  107.1447  15.3064  54.8844     0.0
Residual   21.0    5.8566   0.2789      NaN     NaN
```

- $3 + 7 + 21 = 31 = gb - 1$, and $21 = (g - 1)(b - 1)$.

Ignoring the tasters

```
one_way = smf.ols("rating ~ C(recipe)", data=bars).fit()

print(anova_lm(one_way).round(4))
```

```
               df     sum_sq  mean_sq       F  PR(>F)
C(recipe)    3.0    26.5359   8.8453  2.1917  0.1112
Residual    28.0   113.0012   4.0358     NaN     NaN
```

- Same $\mathrm{SS}_{\mathrm{Trt}} = 14.4675$. The residual line has swallowed $\mathrm{SS}_{\mathrm{Blk}} = 107.635$.
- F falls from 17.30 to 1.19, while the p -value rises from less than 0.001 to 0.33.

The same picture in one line

```
ms_res = [anova_lm(m).loc["Residual",
          "mean_sq"] for m in
          (one_way, two_way)]

print(f"one-way  MS_Res = {ms_res[0]:.4f}")
print(f"blocked  MS_Res = {ms_res[1]:.4f}")
print(f"ratio           = "
      f"{ms_res[0] / ms_res[1]:.1f}")
```

```
one-way  MS_Res = 4.0358
blocked  MS_Res = 0.2789
ratio           = 14.5
```

- Blocking did not find a bigger effect. It found a smaller ruler.

The sums of squares by hand

```
y = bars["rating"].values
Xb = two_way.model.exog
Tb, Bb = Xb[:, 1:g], Xb[:, g:]

def hat(M):
    return M @ np.linalg.solve(M.T @ M, M.T)

H, HT, HB = hat(Xb), hat(Tb), hat(Bb)
N = g * b

print(f"SS_Trt = {y @ HT @ y:.4f}")
print(f"SS_Blk = {y @ HB @ y:.4f}")
print(f"SS_Res = {y @ (np.eye(N) - H) @ y:.4f}")
```

```
SS_Trt = 26.5359
SS_Blk = 107.1447
SS_Res = 5.8566
```

- The same three numbers as `anova_lm`, read straight off the projections.

The test

```
from scipy import stats

F = table.loc["recipe", "F"]
df1, df2 = g - 1, (g - 1) * (b - 1)
crit = stats.f.ppf(0.95, df1, df2)
pval = stats.f.sf(F, df1, df2)

print(f"F       = {F:.3f}")
print(f"F crit  = {crit:.3f}")
print(f"p-value = {pval:.6f}")
print(f"df      = {df1}, {df2}")
```

```
F       = 31.717
F crit  = 3.072
p-value = 0.000000
df      = 3, 21
```

An aside: a real paired data set

```
# Boys' shoe soles, two materials, one on each foot: Box, Hunter & Hunter (2005).
wearA = [13.2, 8.2, 10.9, 14.3, 10.7, 6.6, 9.5, 10.8, 8.8, 13.3]
wearB = [14.0, 8.8, 11.2, 14.2, 11.8, 6.4, 9.8, 11.3, 9.3, 13.6]

shoes = pd.DataFrame({"boy": np.tile(np.arange(1, 11), 2),
                      "material": np.repeat(["A", "B"], 10),
                      "wear": wearA + wearB})
shoe_fit = smf.ols("wear ~ C(material) + C(boy)", data=shoes).fit()
print(anova_lm(shoe_fit).round(4))
```

```
              df    sum_sq  mean_sq        F  PR(>F)
C(material)  1.0    0.8405   0.8405   11.215  0.0085
C(boy)       9.0  110.4905  12.2767  163.811  0.0000
Residual     9.0    0.6745   0.0749      NaN     NaN
```

- $g = 2, b = 10$: the paired design, with $(g-1)(b-1) = 9$ residual df.

Summary

	Ignore the blocks	Block
Design matrix	$(\mathbf{1} \mid \boldsymbol{T})$	$(\mathbf{1} \mid \boldsymbol{T} \mid \boldsymbol{B})$
$\boldsymbol{X}^\top \boldsymbol{X}$	two blocks	three blocks
$\mathrm{SS}_{\mathrm{Trt}}$	same	same
Residual SS	$\mathrm{SS}_{\mathrm{Blk}} + \mathrm{SS}_{\mathrm{Res}}$	$\mathrm{SS}_{\mathrm{Res}}$
Residual df	$g(b-1)$	$(g-1)(b-1)$
Taste test	$F = 1.19$	$F = 17.30$

- Block on what you expect to matter and cannot control. Randomize within.
- Next: what to do when a block is too small to hold every treatment.