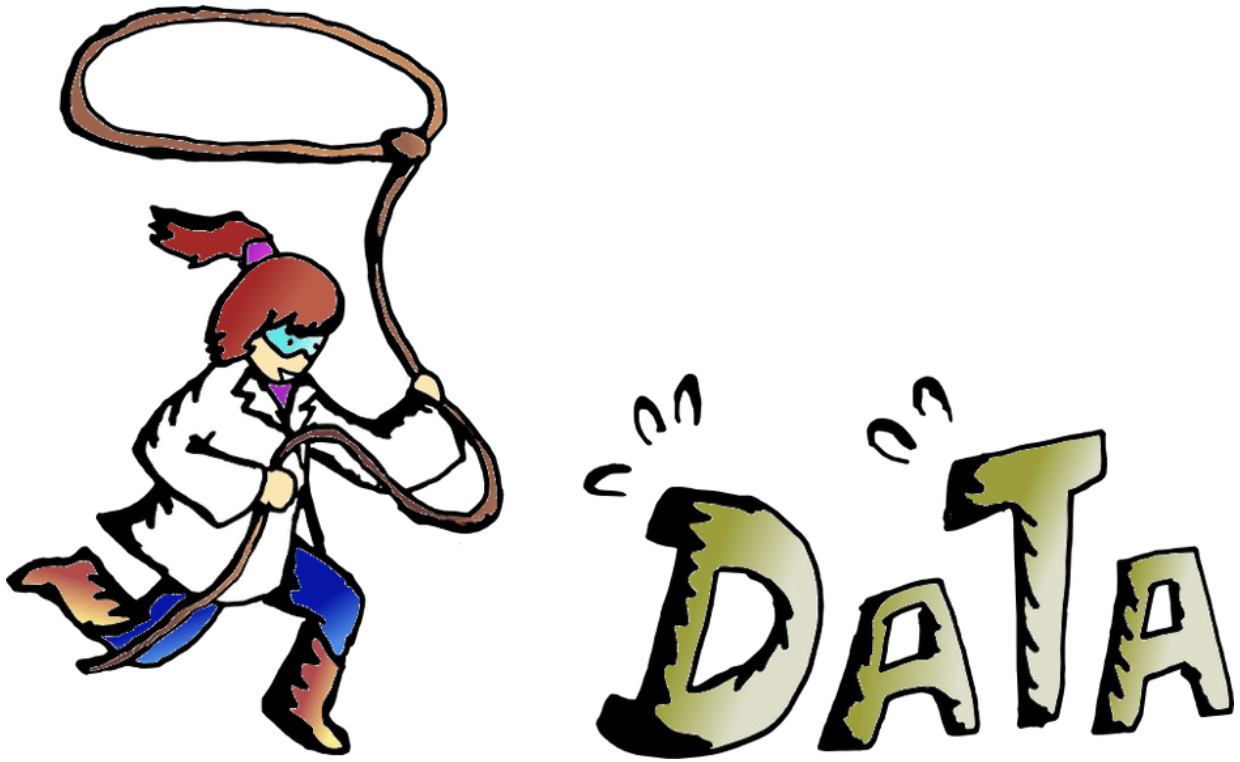
Data wrangling/manipulation



Overview

Review and continuation of theories of hypothesis tests

Data wrangling/manipulation with dplyr

If there is time:

- start on visualizing data using the grammar of graphics

Midterm exam

Homework 5 has been posted

- I strongly recommend you do the first two parts prior to next class

Midterm exam is on Thursday Octobter 10th in person during regular class time

- The exam is on paper
- If you have accommodations, please schedule to take your exam with SAS and let me know

A practice exam has been posted soon under the class 10 material

Midterm exam “cheat sheet”

You are allowed an exam “cheat sheet”

One page, double sided, that contains **only code and equations**

- No code comments allowed

Cheat sheet must be on a regular 8.5 x 11 piece of paper

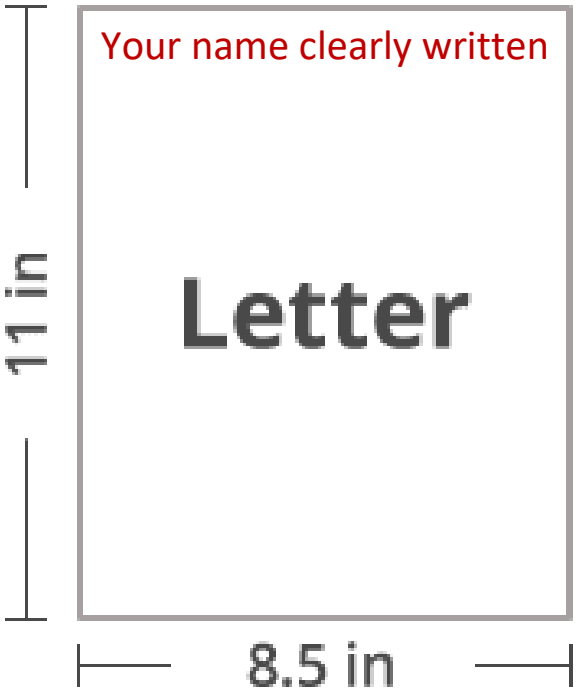
- Your name on the upper left of both sides of the paper

Strongly recommend making a typed list of all functions discussed in class and on the homework

- This will be useful beyond the exam

You must turn in your cheat sheet with the exam

- Failure to do so will result in a 20 point deduction



Plan for the semester

1	Aug 29	Course overview, introduction to R, descriptive statistics
2	Sep 3-6	Review of central statistical concepts and exploratory analysis using R
3	Sep 10-12	Confidence Intervals and the bootstrap
4	Sep 17-19	Review of hypothesis tests and permutation tests in R
5	Sep 24-26	Parametric tests and theories of hypothesis testing
6	Oct 1-3	Data manipulation and visualization
7	Oct 8-10	Review and midterm exam
8	Oct 15-17	Functions, misc, and October break

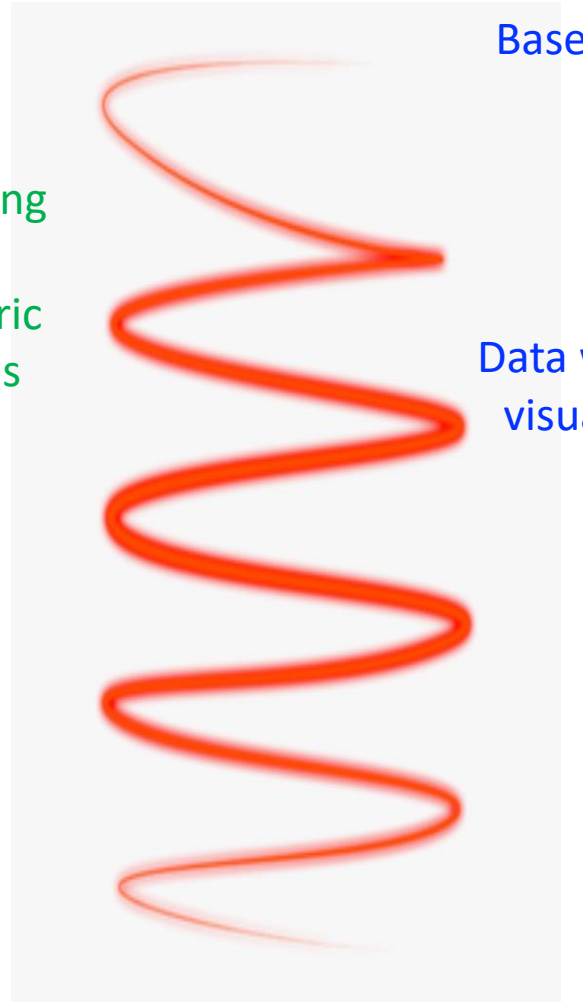
Analysis

Resampling and parametric methods

R

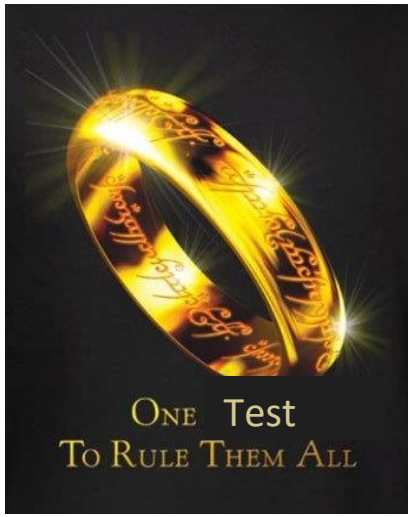
Base R

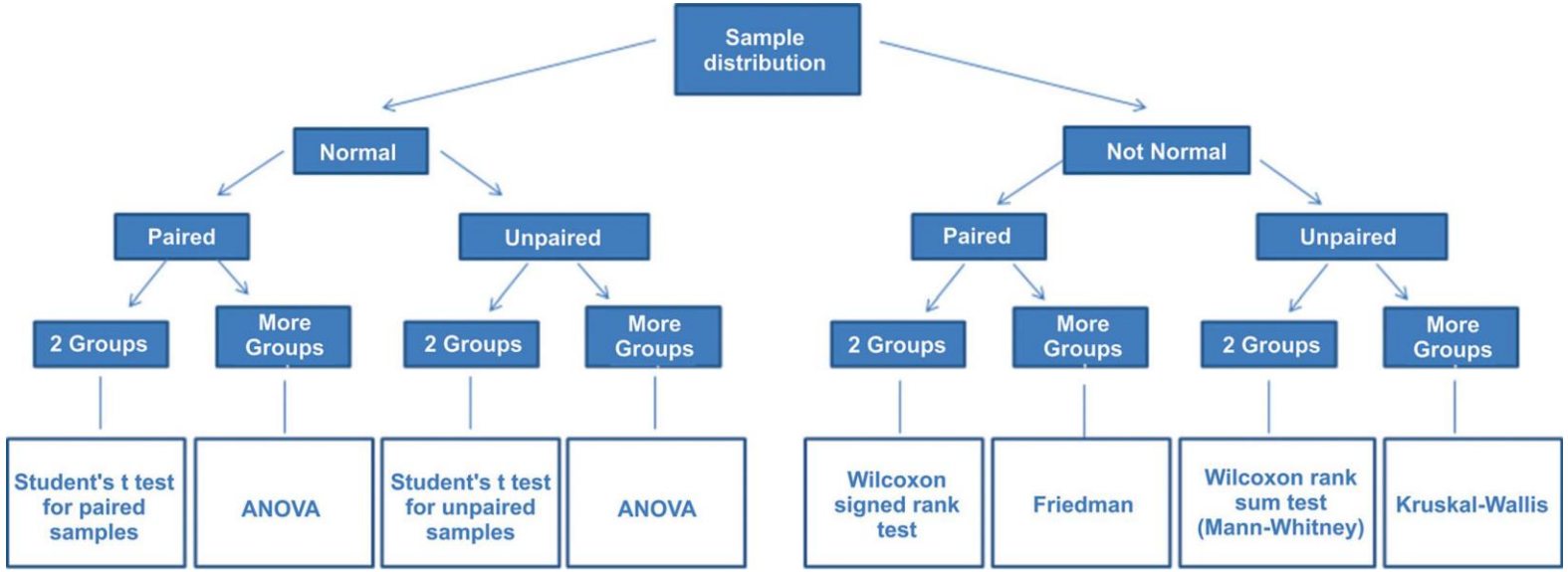
Data wrangling visualization



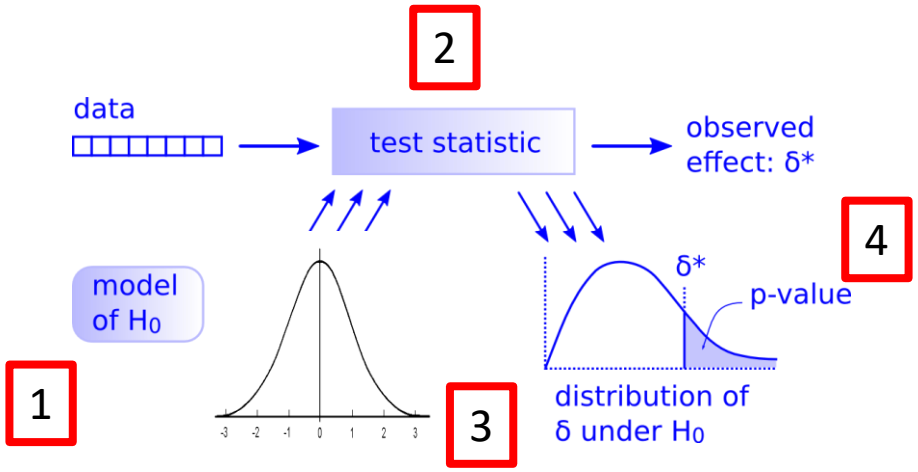
Questions about anything?

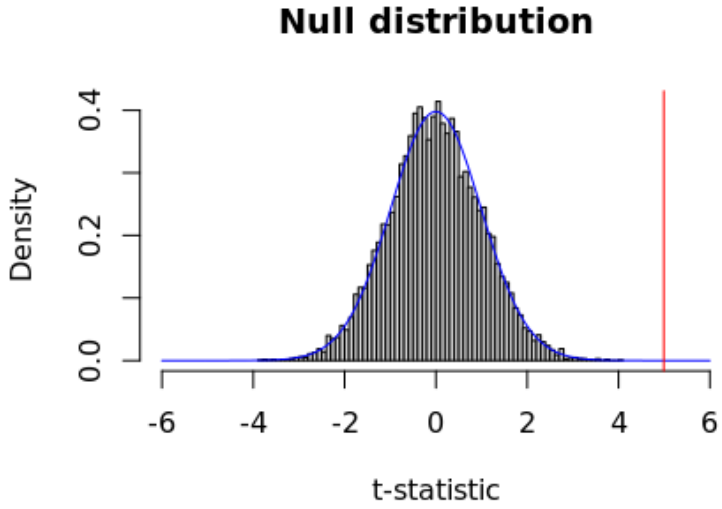
Very quick review



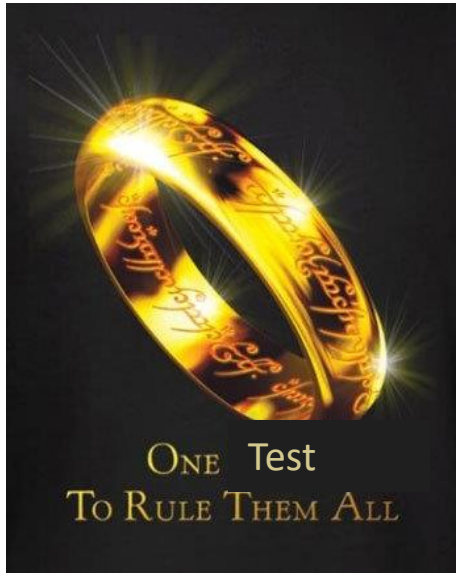


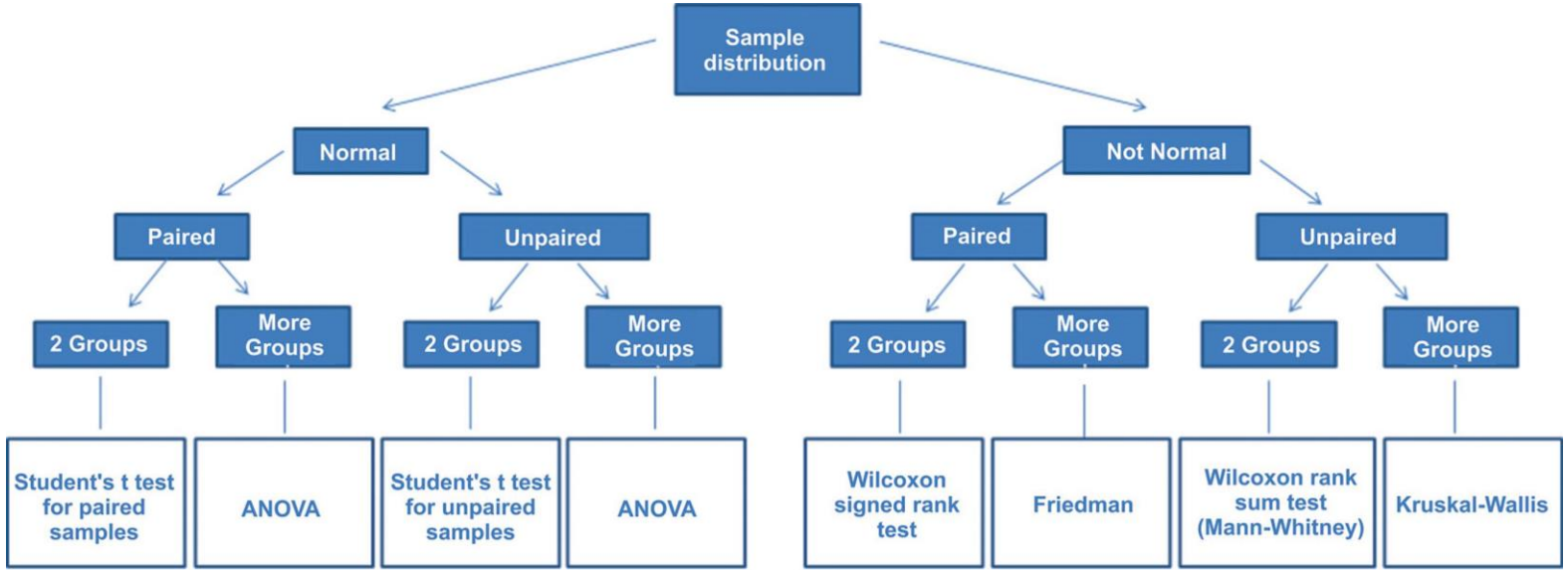
Just need to follow 5 steps!





Very quick review





To select the appropriate parametric test, focus on the parameters being tested in the null hypothesis

- E.g., $H_0: \pi = 0.5$ $H_0: \mu = 0.5$ $H_0: \mu_T = \mu_C$ $H_0: \mu_1 = \mu_2 = \ldots = \mu_k$

Parametric tests are derived from particular mathematical assumptions

- E.g., data from the two samples comes from normal populations with the same variance
- Some hypothesis tests are "robust" to violations of these assumptions
  - The robustness can be evaluated this through computer simulations

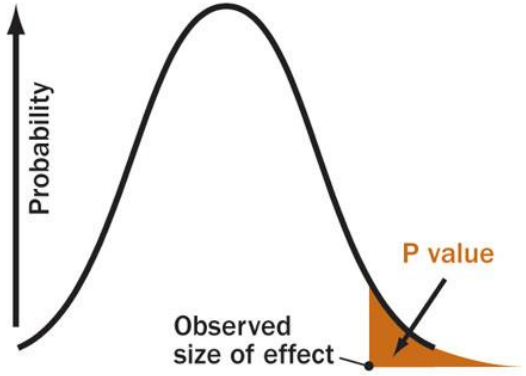
Very quick review: theories of hypothesis testing

Fisher (1890-1962)

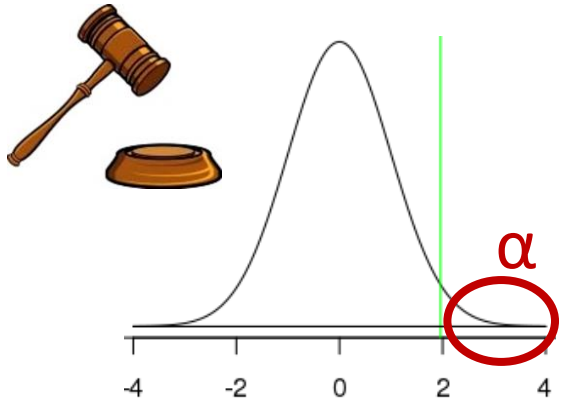
Neyman (1894-1981)

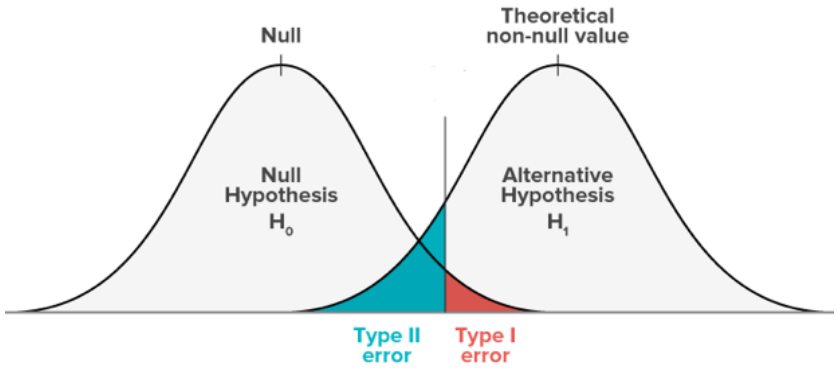
Pearson (1895-1980)

p-value a strength of evidence



Use p-value to make a decision





Neyman-Pearson frequentist logic

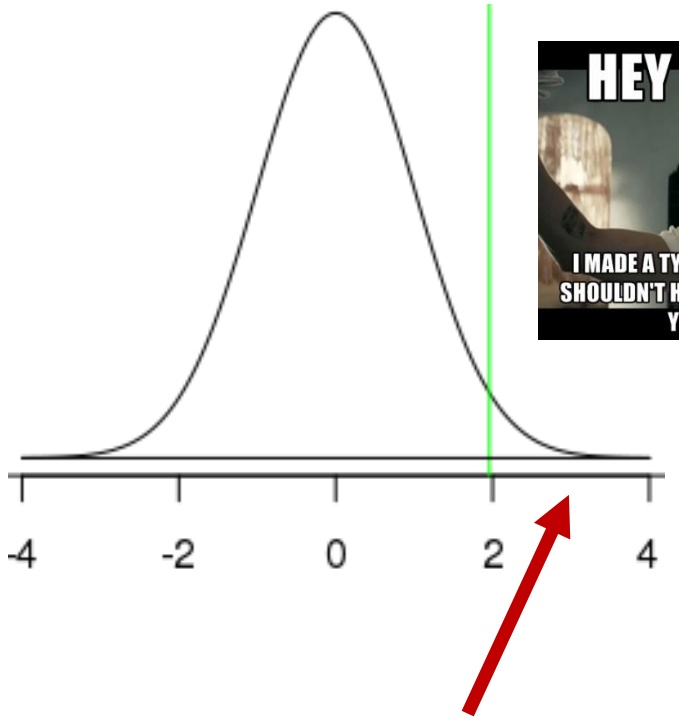
Type I error: incorrectly rejecting the null hypothesis when it is true

If we were in a world where the null hypothesis was always true…

Then only ~5% of the time would we falsely report an effect (for $\alpha = 0.05$)

- i.e., we would only make type I errors 5% of the time

Null distribution



The null distribution is true but statistic landed here

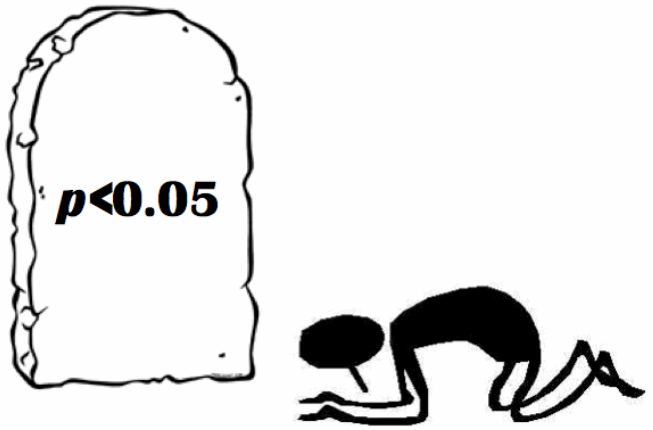
Problems with the NP hypothesis tests

Problem 1: we are interested in the results of a specific experiment, not whether we are right most of the time

- E.g., 95% of these statements are true:
  - Joy can't smell Parkinson's disease, there is no difference in beer consumption across continents, Gingko has no benefits for your memory, …

Problem 2: Arbitrary thresholds for alpha levels

- P-value = 0.051, we don't reject H_0



Problems with the NP hypothesis tests

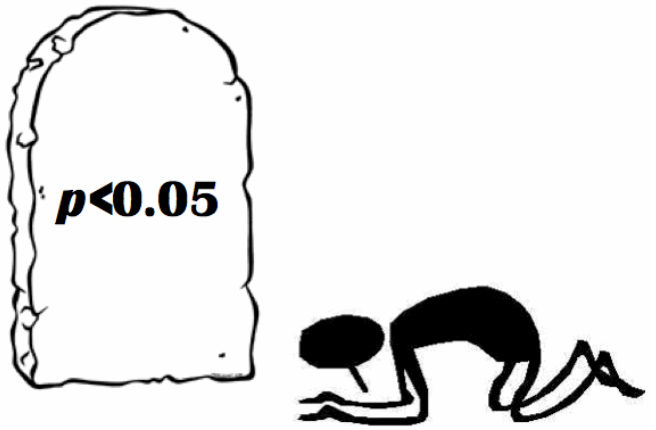
Problem 1: we are interested in the results of a specific experiment, not whether we are right most of the time

- E.g., 95% of these statements are false:
  - Joy can't smell Parkinson's disease, there is no difference in beer consumption across continents, Gingko has no benefits for your memory, …

Problem 2: Arbitrary thresholds for alpha levels

- P-value = 0.051, we don't reject H_0

Problem 3: running many tests can give rise to a high number of type I errors

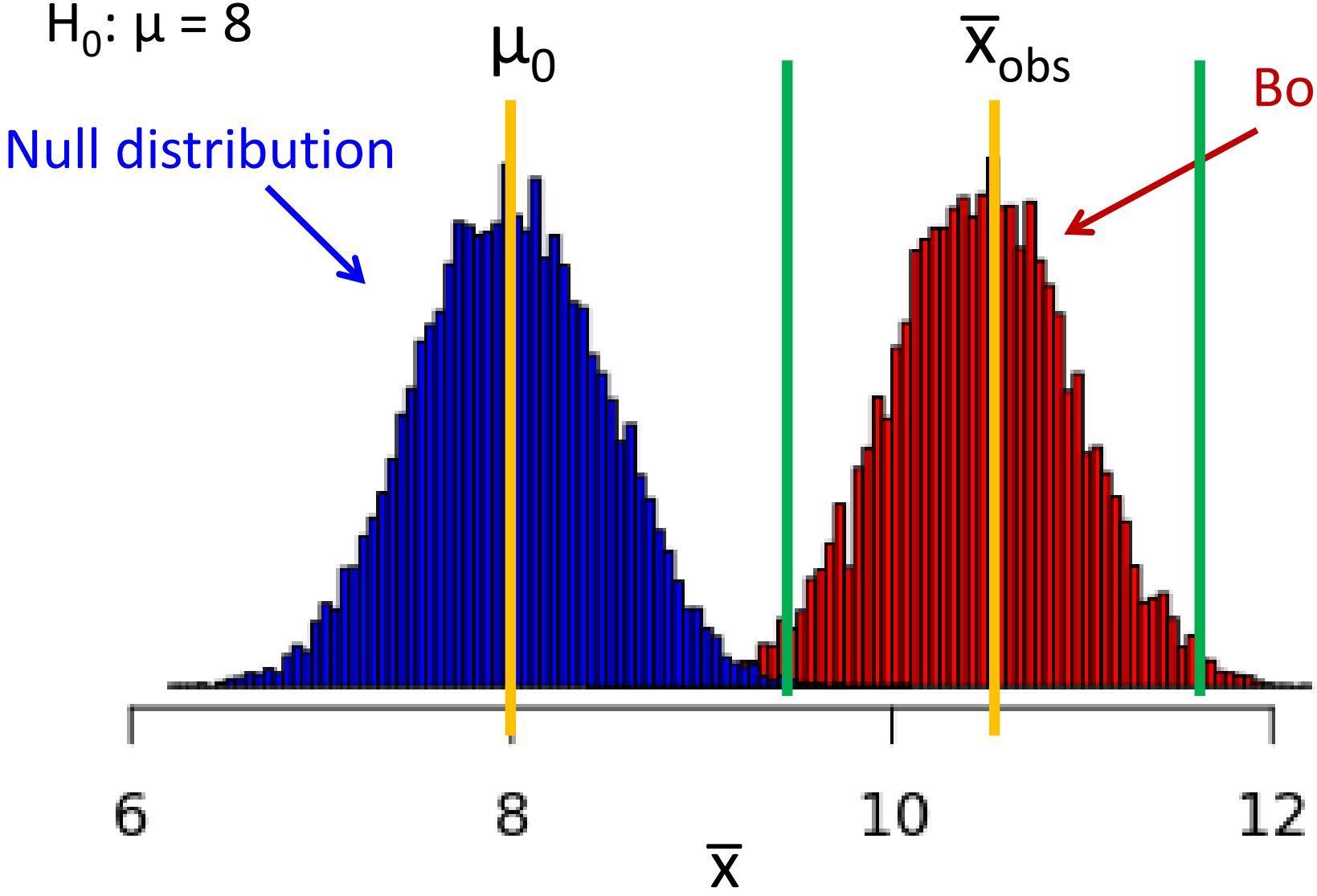


Don't ever do this!

http://xkcd.com/882/

Relationship between null and bootstrap distributions

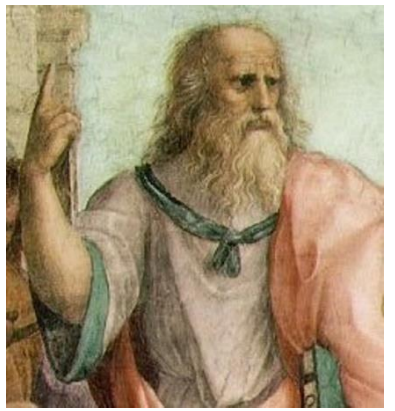
H_0: $\mu = 8$



95% CI limits

Where is:

- 1. The true μ?
- 2. The true alternative sampling distribution?



Questions?

The tidyverse and dplyr

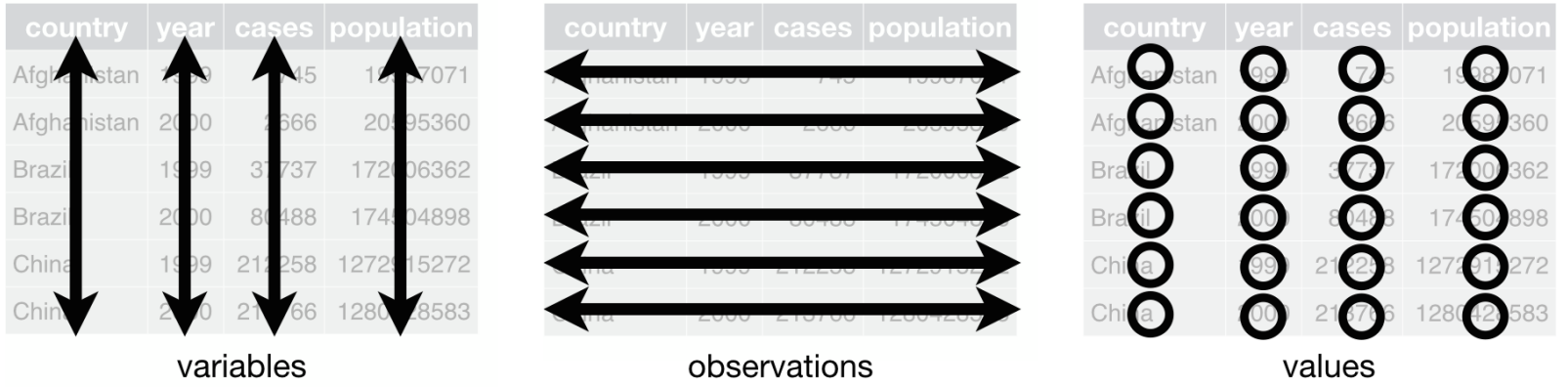
The ‘tidyverse’

The tidyverse is set of R packages that operate ‘tidy data’

- i.e., that operate on data frames (or tibbles)

Tidy data is data where:

- Each variable must have its own column
- Each observation must have its own row
- Each value must have its own cell



Messy data…

Messy data can be difficult to deal with

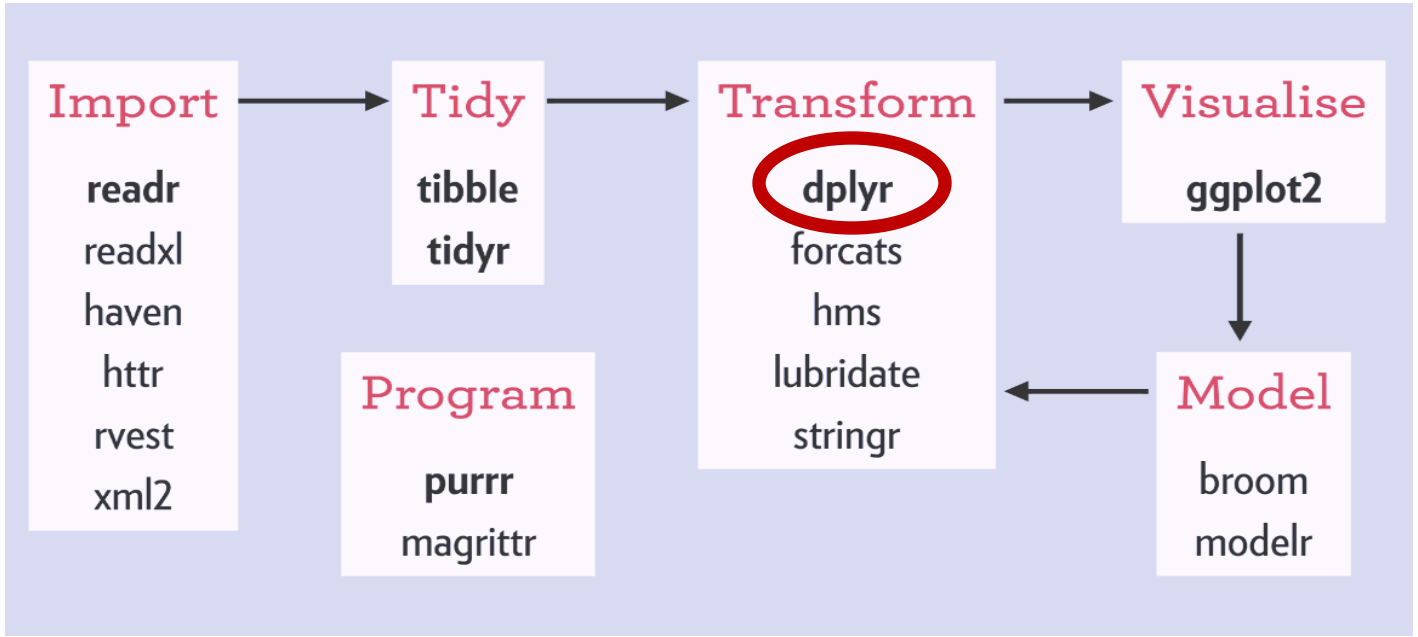
Curve information - Curve c		
Name	Formula	Slope at
Standard	Calc 1: C	standar
Plate information		
Plate	Repeat	Barcode
1	1	
Background information		
Plate	Label	Result
1	PicoGree	0
Calculate	standarc	standar
	1	2
A	-0.0011	-0.0011
B	0.0012	0.0014
C	0.0016	0.0013
D	0.0019	0.0024
E	-0.001	-0.0011
F	-0.001	-0.0011
G	-0.0011	-0.0011
H	-0.0011	-0.0012

			te
			rc 10.12.2013 10:23:33

The ‘tidyverse’

The packages share a common design philosophy

- Most written by Hadley Wickham



dplyr: A grammar for data wrangling

Grammar: a set of components that can be combined to achieve a goal

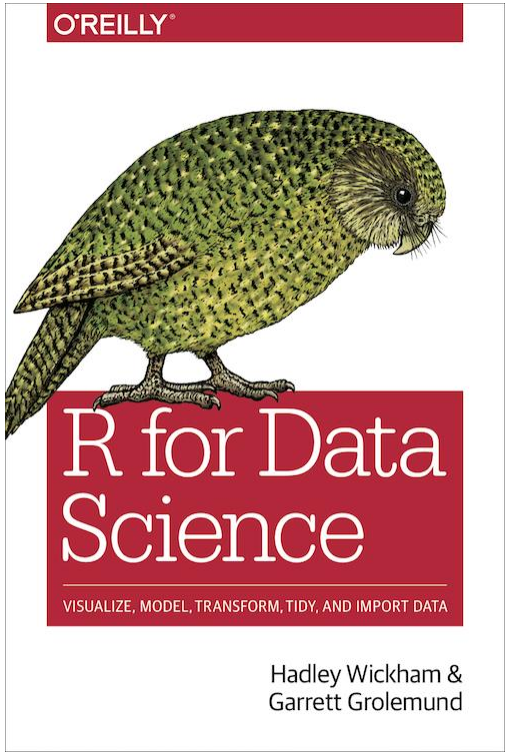
dplyr is a package that has a set of verbs that are useful for transformations data:

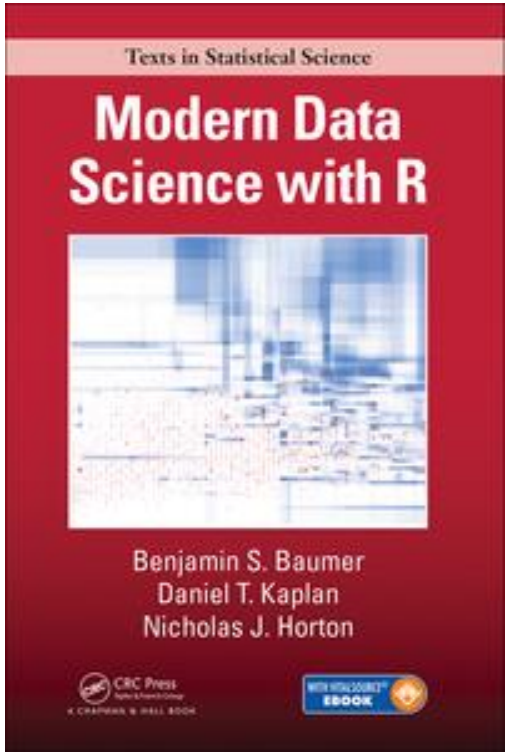
1. filter()
2. select()
3. mutate()
4. arrange()
5. group_by()
6. summarize()

All these function **take a data frame** and other arguments and **return a data frame**

```
> library(dplyr)   # load the dplyr package
```

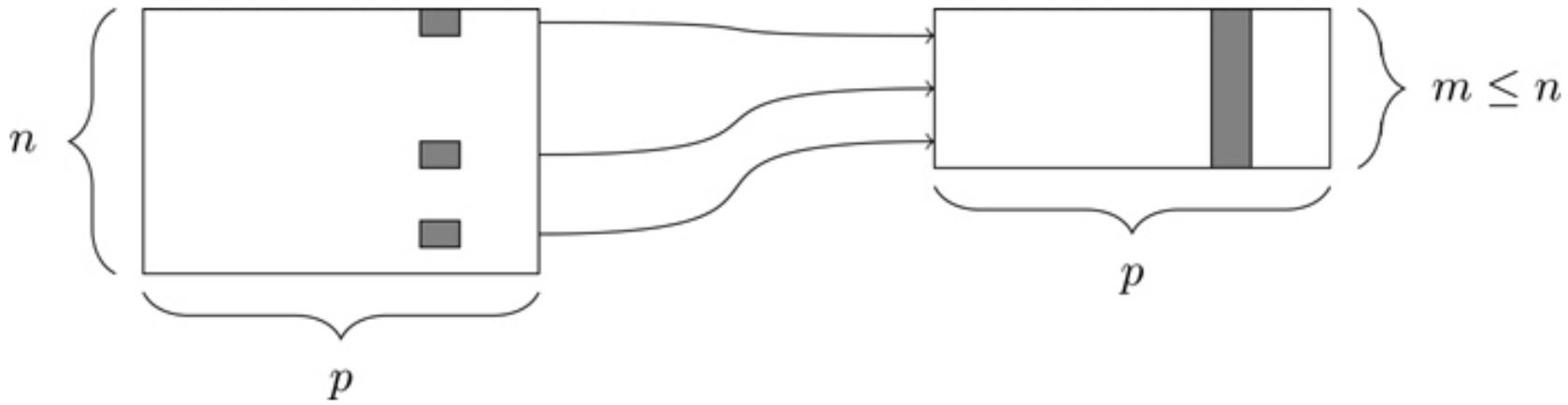
Quick overview of the dplyr functions





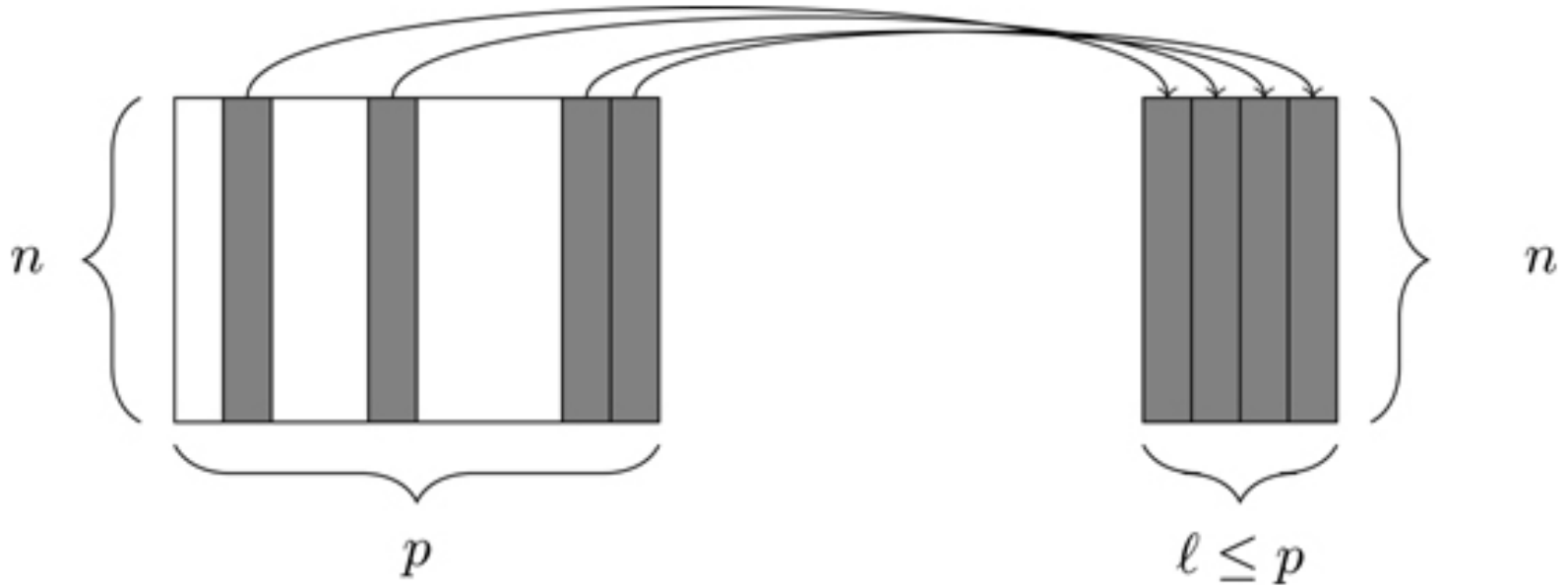
1. filter()

The filter() function allows you to select a subset of rows in data frame



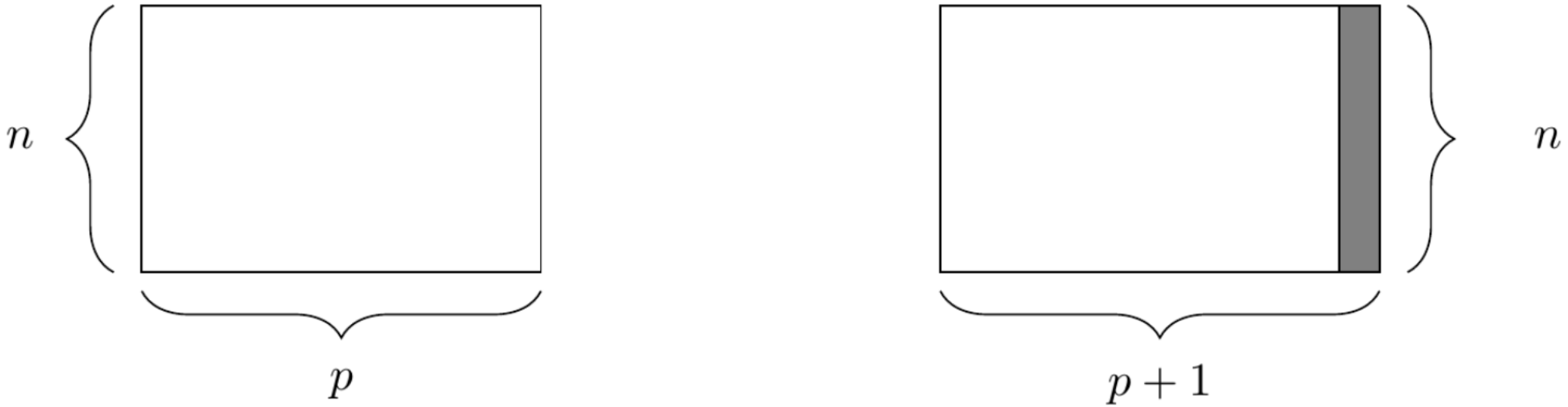
2. select()

The select() function allows you to select a subset of columns



3. mutate()

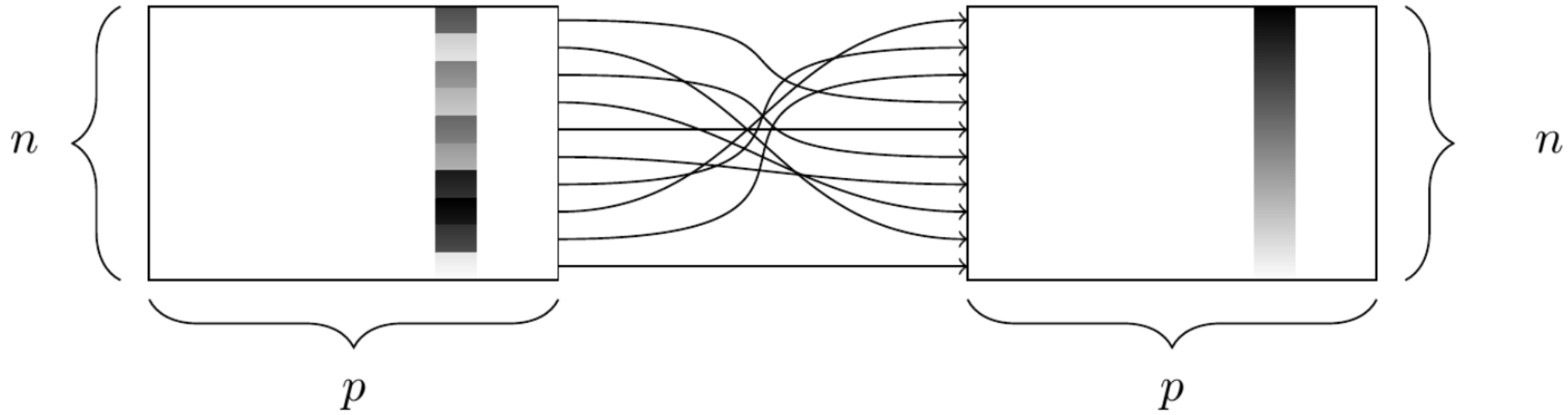
The mutate() function allows you to create new columns that are functions of existing columns



4. arrange()

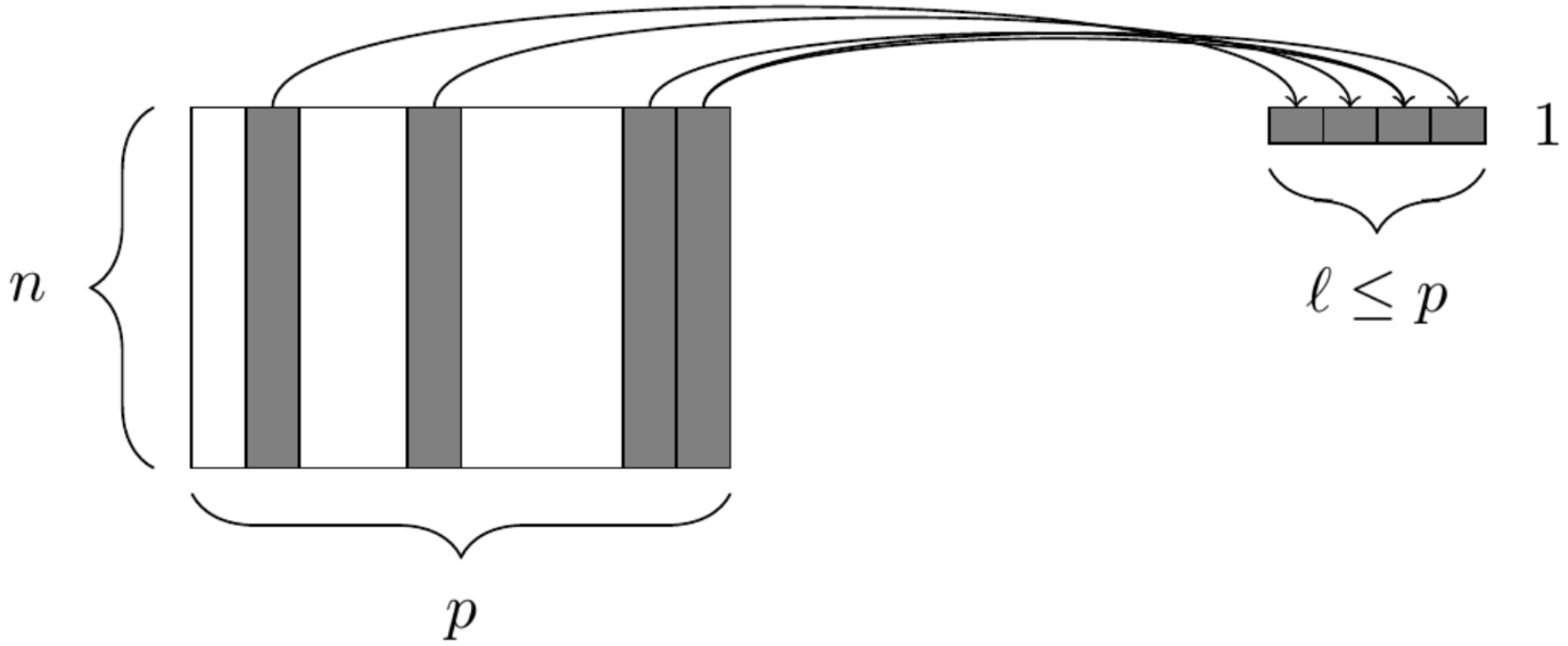
The arrange() function arranges the rows based values in a column

- arrange(desc()) arranges from largest to smallest



5. summarize()

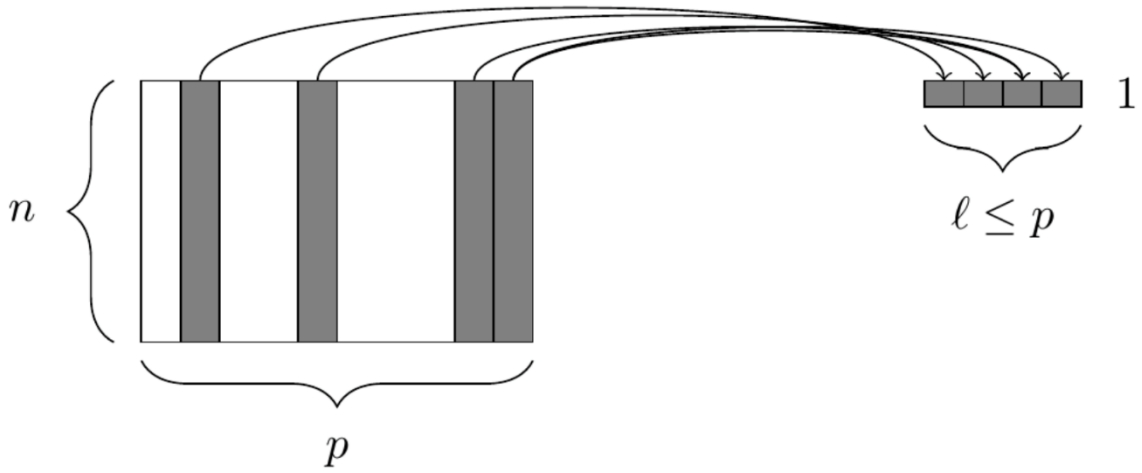
The summarize() function reduces values in many rows into single values

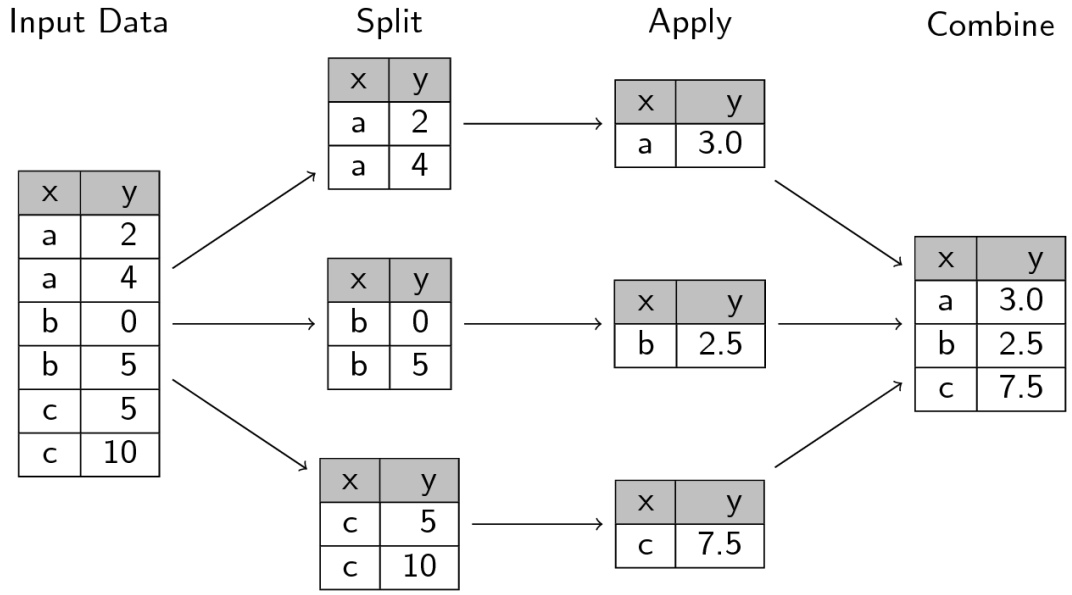


6. The group_by() function

The group_by() function groups variables for future operations

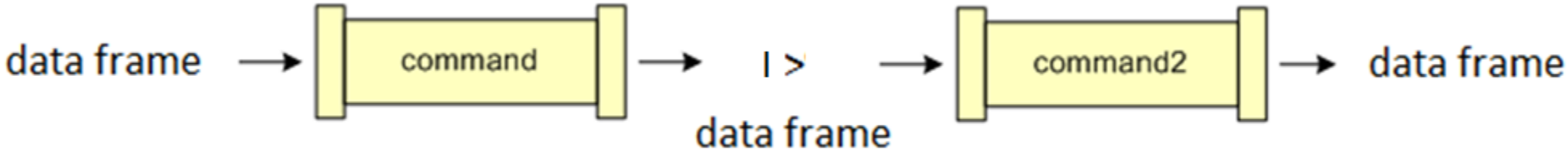
- It works in conjunction with summarize() and mutate() to do split, apply, combine





The pipe operator

The pipe operator |> allows us to chain commands together



Let’s try it out!

Homework 5: weather predictions

Assessing the accuracy and visualizing weather predictions

I recommend you get started soon

Next class: a grammar of graphics and ggplot