
REVIEW

Overview

Review of ggplot

Review of material covered in the class so far

If there is time/interest

- Bonus features of ggplot: special geoms, animation, interactive graphics

Announcements

Midterm exam is on Thursday

- Bring a pen and a pencil
- One page (2 sides) with code and equations only!
  - You will turn in this page of notes with your exam (put your name on it)
  - You can write down conditions for hypothesis tests. Have SE formulas, etc.

Office hours this week

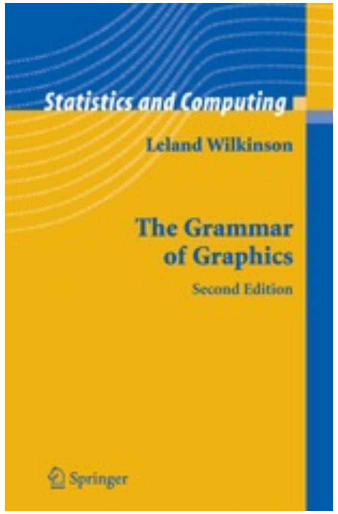
- Stephan and Nathan are doing another review session tonight
- No TA office hours since no homework

Review of the grammar of graphics and ggplot

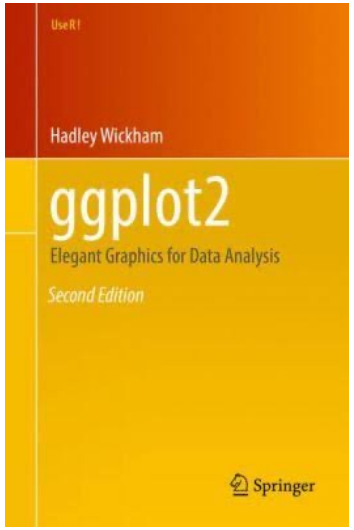
The grammar of graphics

Leland Wilkinson noticed similarities between many graphs and tried to generate a ‘grammar’ that could be used to express a graph

- i.e., a list elements that can be combined together to create a graph



Hadley Wickham implemented these ideas in R in the ggplot2 package



Graphs are composed of…

A Frame: Coordinate system on which data is placed

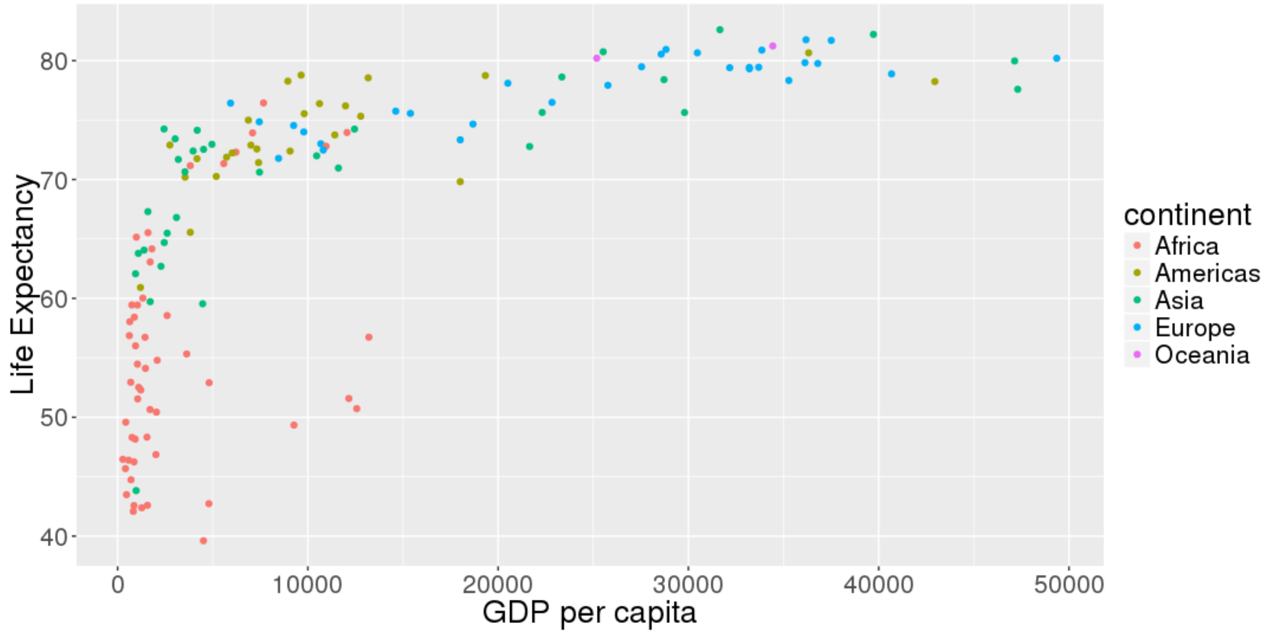
- ggplot()   +

Glyphs: basic graphic unit representing cases or statistics

- Data is **mapped** onto these aesthetics such as: shape, color, size, etc.   and/or   aesthetics can be set to a fixed value
  - geom_point(aes(x = gdpPercap, y = lifeExp, color = continent))   geom_point(aes(x = gdpPercap, y = lifeExp), color = "red")

Scales and guides: shows how to interpret axes and other properties of the glyphs

- scale_x_continuous(trans = "log10")   scale_color_brewer(type = "qua", palette = 2)



Plots can also contain…

Facets: allows for multiple side-by-side graphs based on a categorical variable

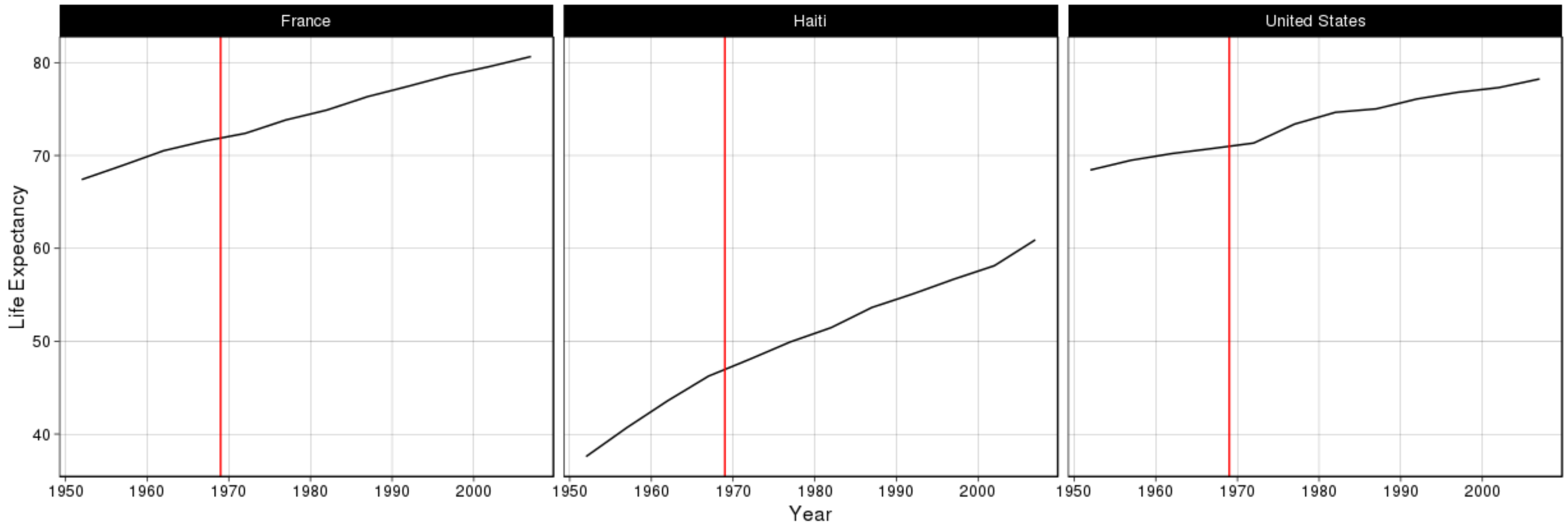
- facet_wrap(~country)

Layers: allows for more than one types of data to be mapped onto the same figure

- geom_vline(xintercept = 1969, col = "red")

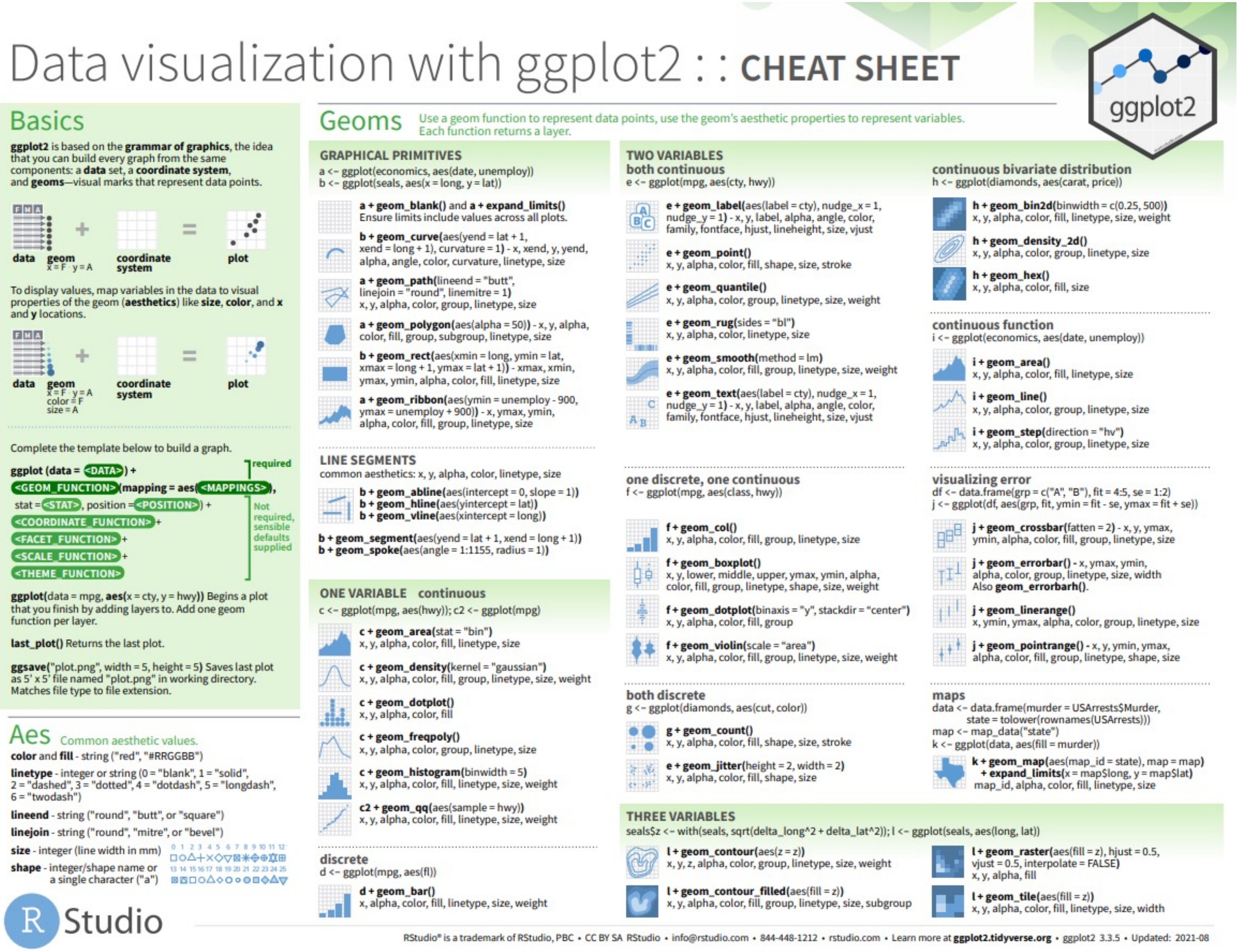
Theme: contains finer points of display (e.g., font size, background color, etc.)

- theme_wsj()



Questions?

[ggplot2 cheat sheet]



Data visualization with ggplot2 : : CHEAT SHEET

ggplot2

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data** set, a **coordinate system**, and **geoms**—visual marks that represent data points.

data + geom x = F · y = A + coordinate system = plot

To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.

data + geom x = F · y = A color = F size = A + coordinate system = plot

Complete the template below to build a graph.

ggplot (data = <DATA>) +
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),
stat = <STAT>, position = <POSITION>) +
<COORDINATE_FUNCTION> +
<FACET_FUNCTION> +
<SCALE_FUNCTION> +
<THEME_FUNCTION>

required

Not required, sensible defaults supplied

ggplot(data = mpg, **aes**(x = cty, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

last_plot() Returns the last plot.

ggsave("plot.png", width = 5, height = 5) Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Aes

Common aesthetic values.

color and **fill** - string ("red", "#RRGGBB")

linetype - integer or string (0 = "blank", 1 = "solid", 2 = "dashed", 3 = "dotted", 4 = "dotdash", 5 = "longdash", 6 = "twodash")

lineend - string ("round", "butt", or "square")

linejoin - string ("round", "mitre", or "bevel")

size - integer (line width in mm)

shape - integer/shape name or a single character ("a")

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

a + geom_blank() and **a + expand_limits()** Ensure limits include values across all plots.

b + geom_curve(aes(yend = lat + 1, xend = long + 1), curvature = 1) - x, xend, y, yend, alpha, angle, color, curvature, linetype, size

a + geom_path(lineend = "butt", linejoin = "round", linemitre = 1) x, y, alpha, color, group, linetype, size

a + geom_polygon(aes(alpha = 50)) - x, y, alpha, color, fill, group, subgroup, linetype, size

b + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1)) - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

a + geom_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900)) - x, ymax, ymin, alpha, color, fill, group, linetype, size

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size

b + geom_abline(aes(intercept = 0, slope = 1))
b + geom_hline(aes(yintercept = lat))
b + geom_vline(aes(xintercept = long))

b + geom_segment(aes(yend = lat + 1, xend = long + 1))
b + geom_spoke(aes(angle = 1:1155, radius = 1))

ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + geom_area(stat = "bin") x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian") x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot() x, y, alpha, color, fill

c + geom_freqpoly() x, y, alpha, color, group, linetype, size

c + geom_histogram(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy)) x, y, alpha, color, fill, linetype, size, weight

discrete

d <- ggplot(mpg, aes(fl))

d + geom_bar() x, alpha, color, fill, linetype, size, weight

TWO VARIABLES

both continuous

e <- ggplot(mpg, aes(cty, hwy))

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_point() x, y, alpha, color, fill, shape, size, stroke

e + geom_quantile() x, y, alpha, color, group, linetype, size, weight

e + geom_rug(sides = "bl") x, y, alpha, color, linetype, size

e + geom_smooth(method = lm) x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

one discrete, one continuous

f <- ggplot(mpg, aes(class, hwy))

f + geom_col() x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot() x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + geom_dotplot(binaxis = "y", stackdir = "center") x, y, alpha, color, fill, group

f + geom_violin(scale = "area") x, y, alpha, color, fill, group, linetype, size, weight

both discrete

g <- ggplot(diamonds, aes(cut, color))

g + geom_count() x, y, alpha, color, fill, shape, size, stroke

e + geom_jitter(height = 2, width = 2) x, y, alpha, color, fill, shape, size

continuous bivariate distribution

h <- ggplot(diamonds, aes(carat, price))

h + geom_bin2d(binwidth = c(0.25, 500)) x, y, alpha, color, fill, linetype, size, weight

h + geom_density_2d() x, y, alpha, color, group, linetype, size

h + geom_hex() x, y, alpha, color, fill, size

continuous function

i <- ggplot(economics, aes(date, unemploy))

i + geom_area() x, y, alpha, color, fill, linetype, size

i + geom_line() x, y, alpha, color, group, linetype, size

i + geom_step(direction = "hv") x, y, alpha, color, group, linetype, size

visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))

j + geom_crossbar(fatten = 2) - x, y, ymax, ymin, alpha, color, fill, group, linetype, size

j + geom_errorbar() - x, ymax, ymin, alpha, color, group, linetype, size, width Also **geom_errorbarh()**.

j + geom_linerange() x, ymin, ymax, alpha, color, group, linetype, size

j + geom_pointrange() - x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

maps

data <- data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))

k + geom_map(aes(map_id = state), map = map) **+ expand_limits**(x = map$long, y = map$lat) map_id, alpha, color, fill, linetype, size

THREE VARIABLES

seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)); l <- ggplot(seals, aes(long, lat))

l + geom_contour(aes(z = z)) x, y, z, alpha, color, group, linetype, size, weight

l + geom_contour_filled(aes(fill = z)) x, y, alpha, color, fill, group, linetype, size, subgroup

l + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE) x, y, alpha, fill

l + geom_tile(aes(fill = z)) x, y, alpha, color, fill, linetype, size, width

RStudio

RStudio® is a trademark of RStudio, PBC • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at **ggplot2.tidyverse.org** • ggplot2 3.3.5 • Updated: 2021-08


TIME FOR REVIEW

What we have covered so far…

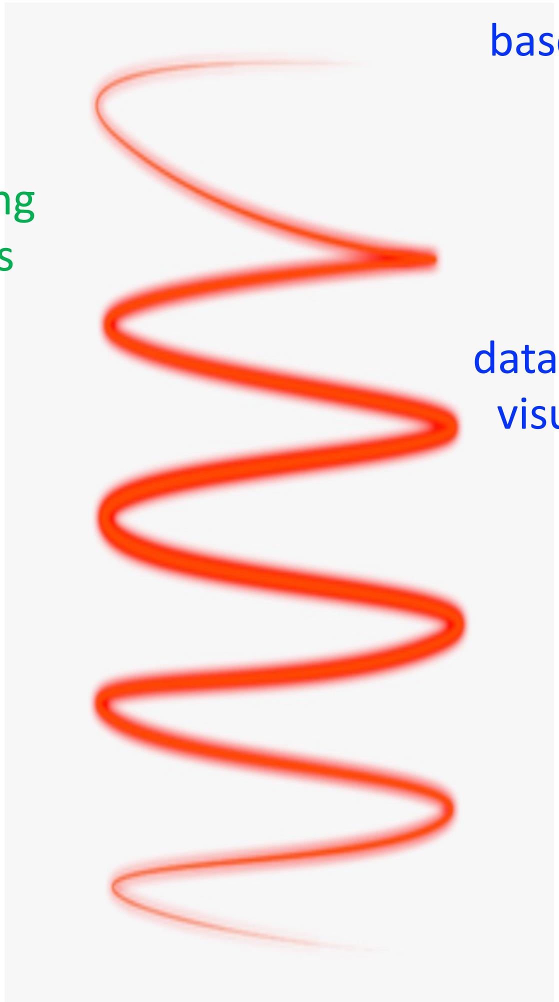
1	Sep 1	Course overview, introduction to R, descriptive statistics
2	Sep 6-8	Review of central statistical concepts and exploratory analysis using R
3	Sep 13-15	Confidence Intervals and the bootstrap
4	Sep 20-22	Review of hypothesis tests and permutation tests in R
5	Sep 27-29	Parametric, non-parametric and theories of hypothesis testing
6	Oct 4-6	Data manipulation and visualization
7	Oct 11-13	Review and midterm exam
8	Oct 18-22	Odds and ends, October break

Analysis

R

base R

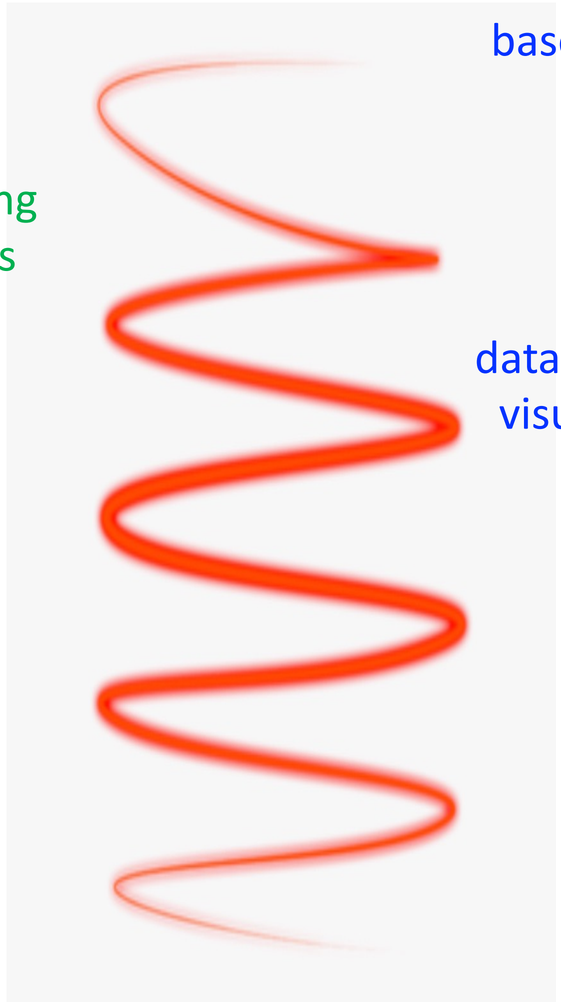
resampling
methods



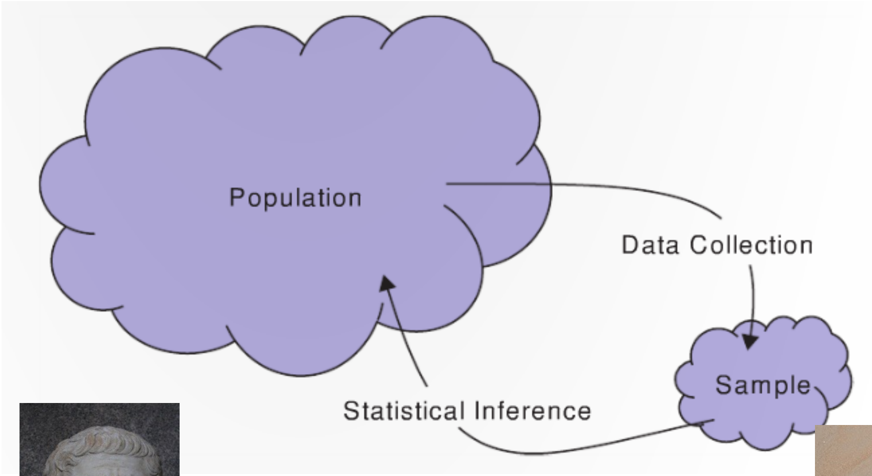
data wrangling
visualization

What we have covered so far…

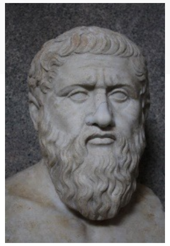
1	Sep 1	Course overview, introduction to R, descriptive statistics
2	Sep 6-8	Review of central statistical concepts and exploratory analysis using R

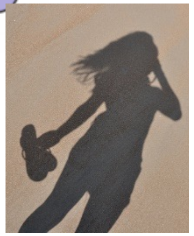


Parameters and statistics commonly used symbols



$$\bar{x} = \frac{\Sigma_i^n x_i}{n}$$





	Population parameter (Plato)	Sample statistic (shadow)
Mean	μ	x̄
Standard deviation	σ	s
Proportion	π	p̂
Correlation	ρ	r
Regression slope	β	b

Base R

Basics of R

```
> my_vec <- c(5, 28, 19)
> inds_less_than_10 <- 10
```

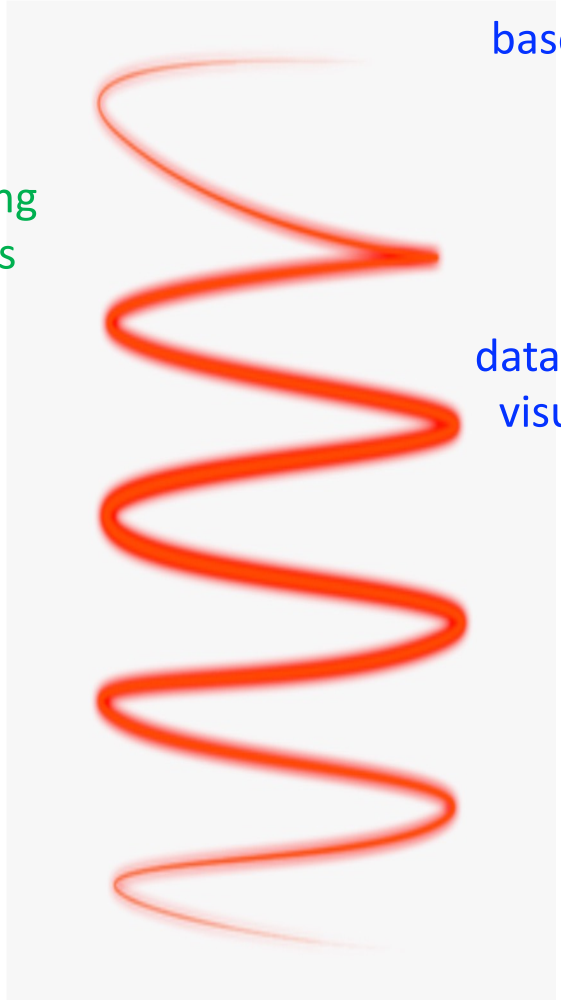
How to plot data in base R

```
> drinks_table <- table(profiles$drinks)
> barplot(drinks_table)
> pie(drinks_table)
> hist(profiles$height)
```

For loops

```
my_results <- NULL
for (i in 1:100) {
	my_results[i] <- i^2
}
```

What we have covered so far…



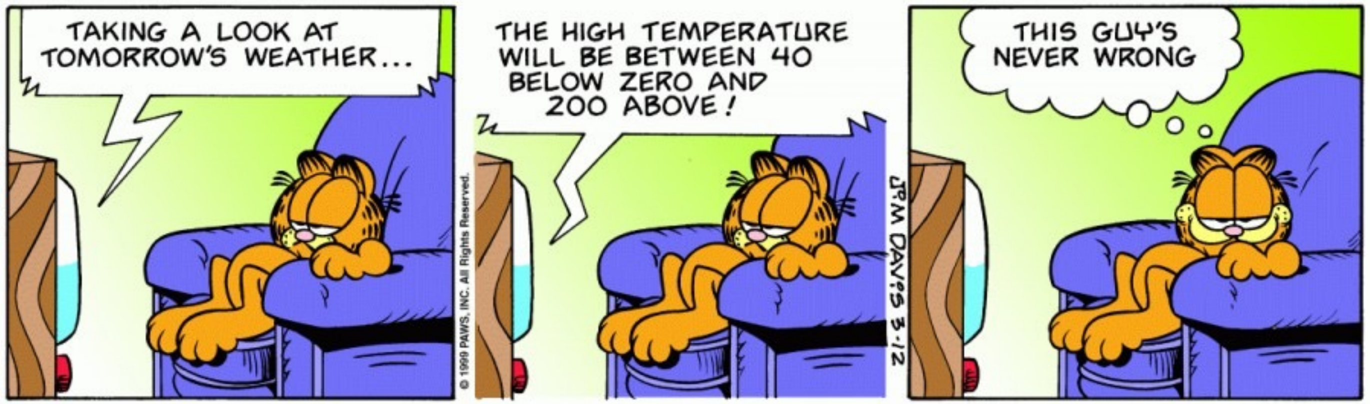
3	Sep 13-15	Confidence Intervals and the bootstrap
4	Sep 20-22	Review of hypothesis tests and permutation tests in R
5	Sep 27-29	Parametric, non-parametric and theories of hypothesis testing

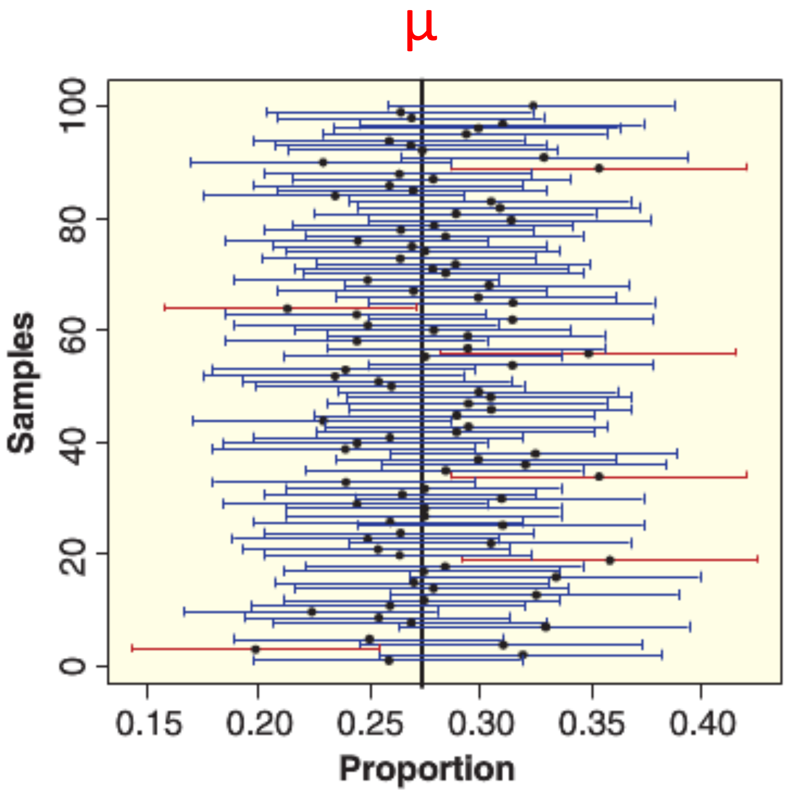
Probability and confidence intervals

Probability functions; e.g., rnorm, pnorm, dnorm, qnorm

Confidence intervals:

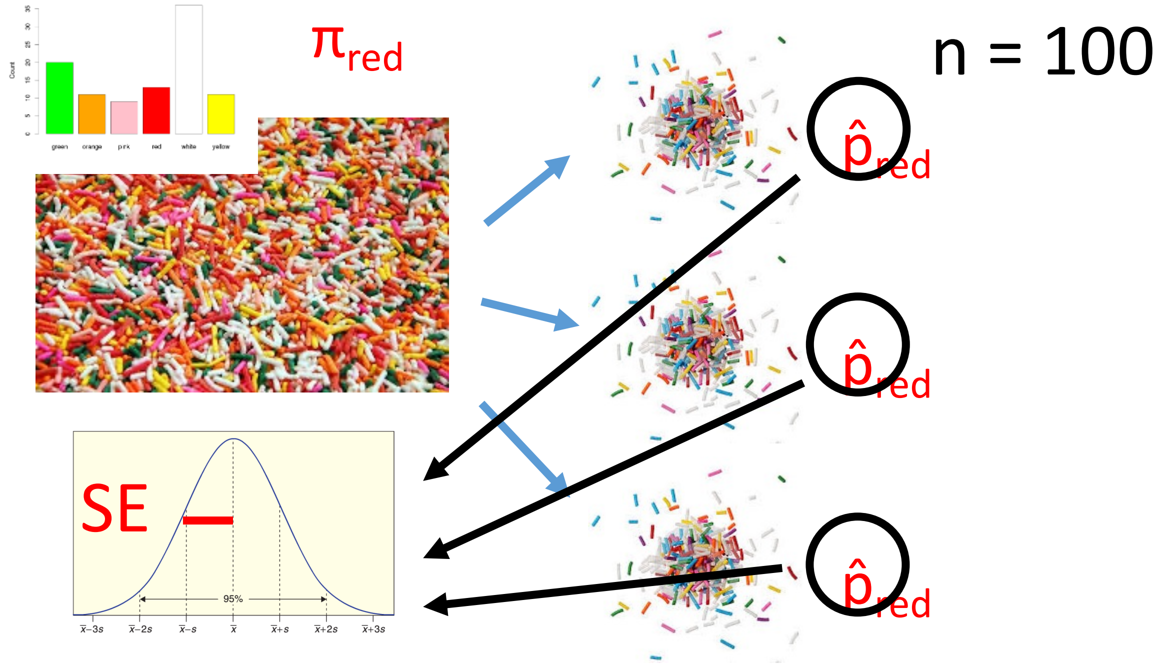
$CI_{95} = \text{stat} \pm 2 \cdot SE$





Sampling and bootstrap distributions

Sampling distribution



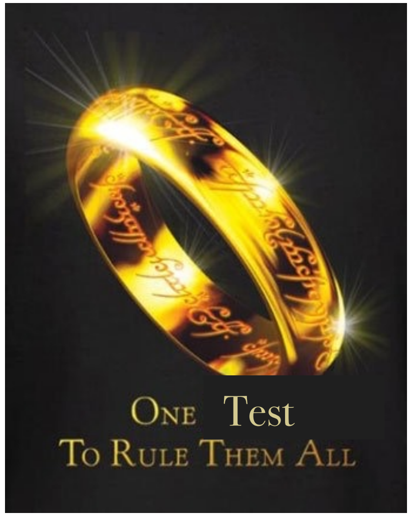
Sampling distribution!

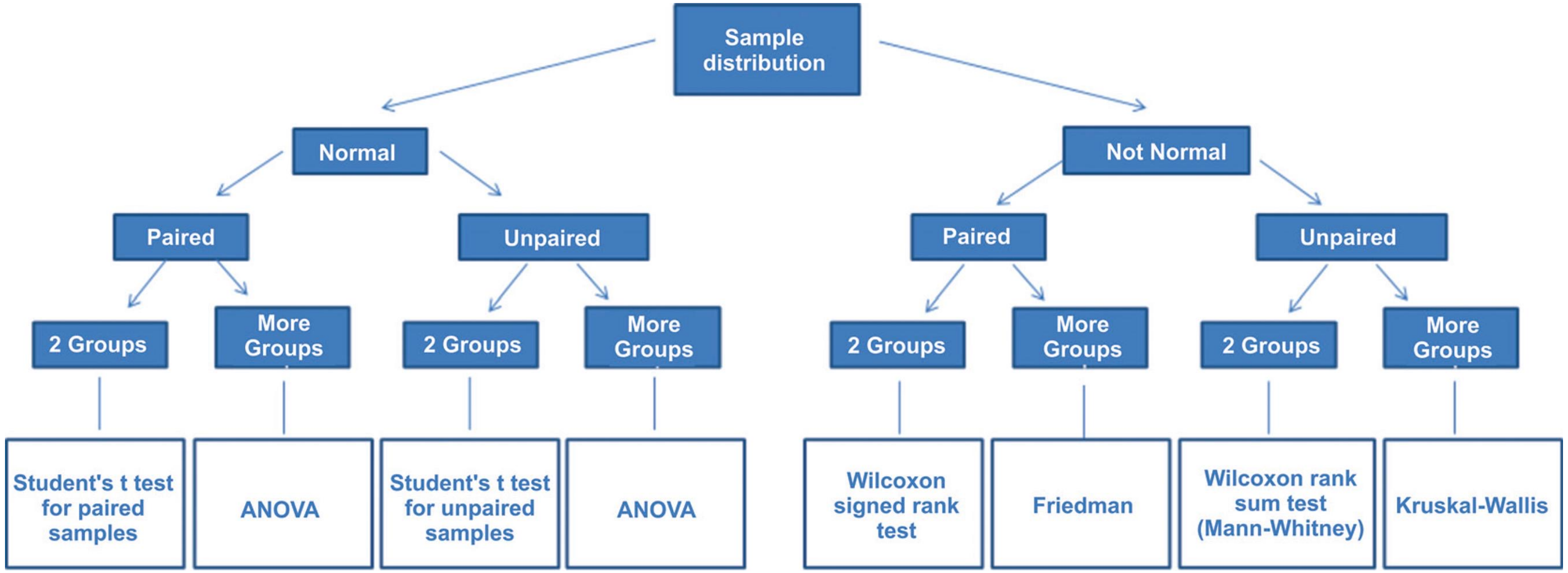
Bootstrap distribution

Sample with replacement from our original sample to mimic a sampling distribution

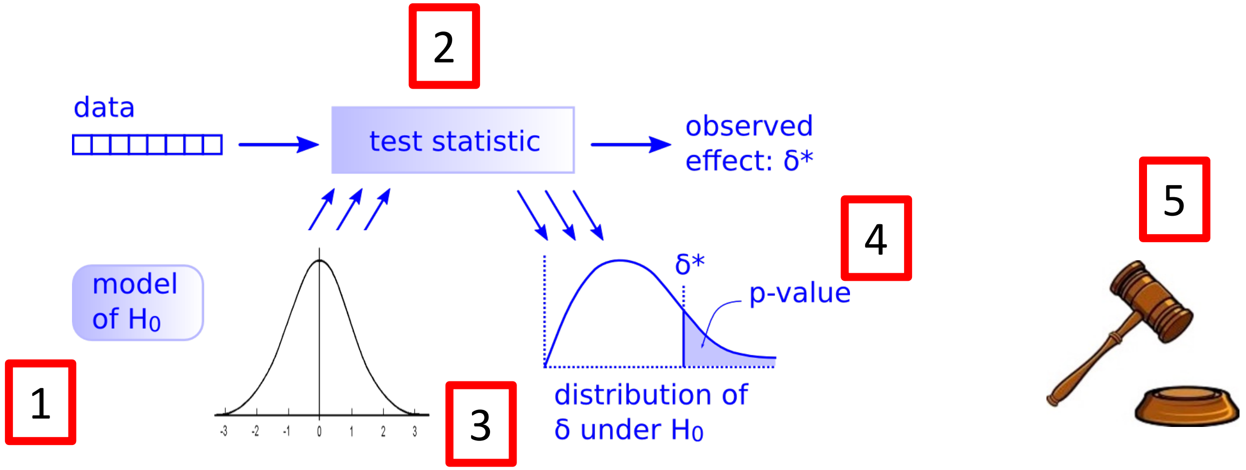
$$CI_{95} = \text{stat} \pm 2 \cdot SE^*$$

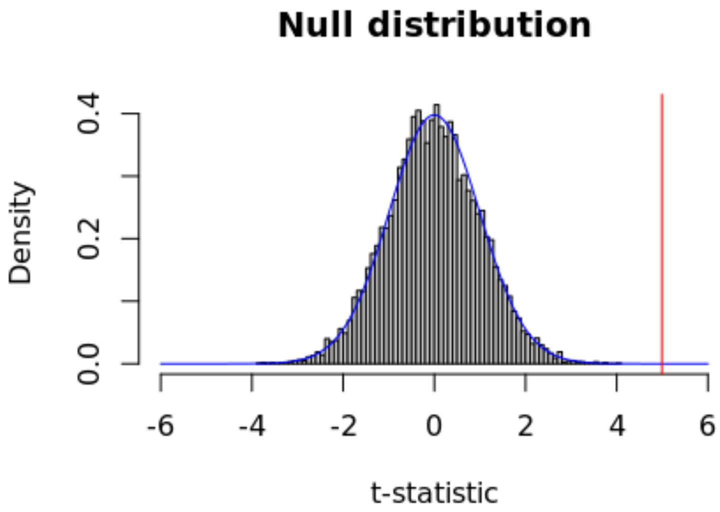
Hypothesis tests





Just need to follow 5 steps!





Randomization/permutation tests

Create a null distribution through computational simulations/shuffling

- rbinom(), sample(), etc.

H_0: $\pi = 0.5$
H_A: $\pi > 0.5$

H_0: $\mu_T - \mu_C = 0$
H_A: $\mu_T - \mu_C > 0$

H_0: $\mu_i = \mu_j \;_{...}= \ldots \mu_k$
H_A: $\mu_i \neq \mu_j$ for some i, j

Data	1 Sample	2 Samples	> 2 Samples
Categorical data	H_0: $\pi = p_0$ H_A: $\pi \neq p_0$ Flip "coins" rbinom()	H_0: $\pi_1 = \pi_2$ H_A: $\pi_1 \neq \pi_2$ Flip "coins" rbinom()	H_0: $\pi_1 = p_1$, $\pi_2 = p_2$, ... , $\pi_k = p_k$ H_A: At least one p_i is different than specified Flip coins rmultinom()
Quantitative data	H_0: $\mu = v_0$ H_A: $\mu \neq v_0$ resample sample(... , replace = TRUE)	H_0: $\mu_1 = \mu_2$ H_A: $\mu_1 \neq \mu_2$ Shuffle data sample()	H_0: $\mu_1 = \mu_2 = ... = \mu_k$ H_A: At least one μ_i is different Shuffle data sample()

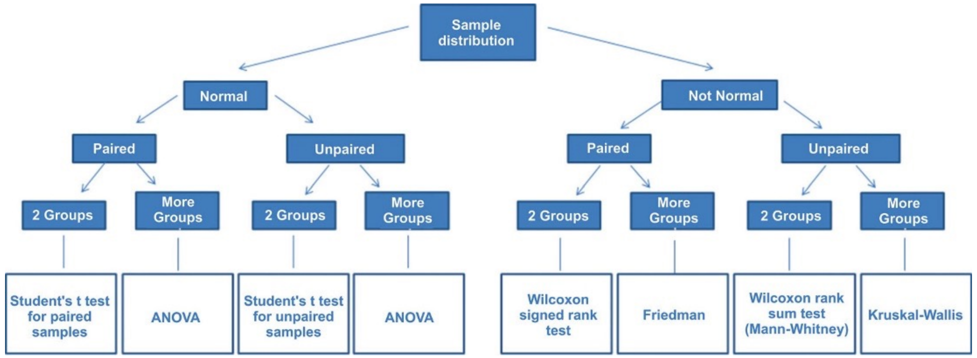
Parametric tests

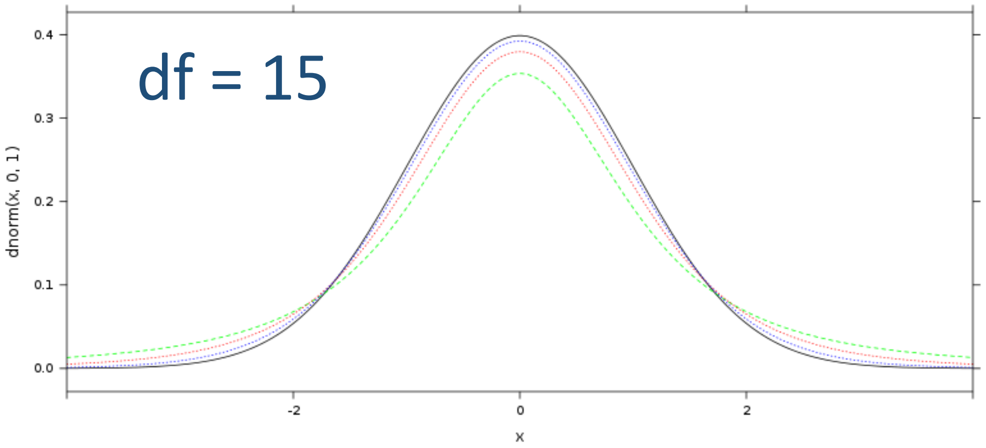
Use mathematical density functions for the null distribution

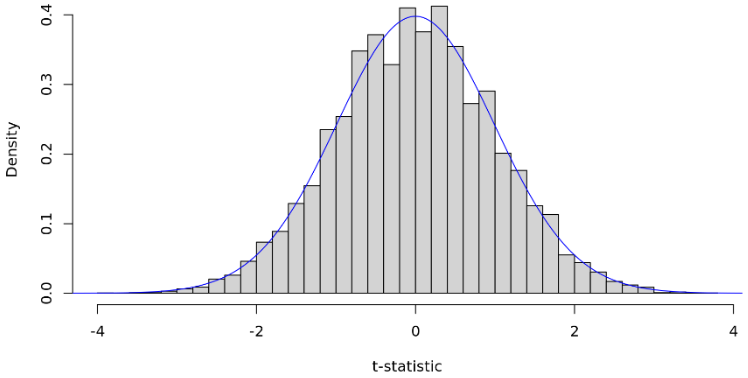
H_0: $\mu_T - \mu_C = 0$

H_A: $\mu_T - \mu_C > 0$

$$t = \frac{\bar{x}_t - \bar{x}_c}{\sqrt{\frac{s_t^2}{n_t} + \frac{s_c^2}{n_c}}}$$







Data	1 Sample	2 Samples	> 2 Samples
Categorical data	H_0: $\pi = p_0$ H_A: $\pi \neq p_0$ <u>z-test</u> $z = \frac{\hat{p}-p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$	H_0: $\pi_1 = \pi_2$ H_A: $\pi_1 \neq \pi_2$ <u>z-test or a chi-square</u> $z = \frac{\hat{p_1} - \hat{p_2}}{\sqrt{\frac{\hat{p_1}(1-\hat{p_1})}{n_1} + \frac{\hat{p_2}(1-\hat{p_2})}{n_2}}}$	H_0: $\pi_1 = p_1$, $\pi_2 = p_2$, … , $\pi_k = p_k$ H_A: At least one p_i is different than specified <u>chi-square test</u> $\chi^2 = \sum_{i=1}^{k} \frac{(Observed_i - Expected_i)^2}{Expected_i}$
Quantitative data	H_0: $\mu = v_0$ H_A: $\mu \neq v_0$ <u>One sample t-test</u> $t = \frac{\bar{x}-v_0}{s/\sqrt{n}}$	H_0: $\mu_1 = \mu_2$ H_A: $\mu_1 \neq \mu_2$ <u>Two sample t-test</u> $t = \frac{\bar{x}_1-\bar{x}_2}{\sqrt{\frac{s_1^2}{n_1}+\frac{s_2^2}{n_2}}}$	H_0: $\mu_1 = \mu_2 = \ldots = \mu_k$ H_A: At least one μ_i is different <u>Analysis of Variance</u> $F = \frac{\frac{1}{K-1}\sum_{i=1}^{K} n_i(\bar{x}_i - \bar{x}_{tot})^2}{\frac{1}{N-K}\sum_{i=1}^{K}\sum_{j=1}^{n_i}(x_{ij} - \bar{x}_i)^2}$ $df_1 = k$, $df_2 = n - k$

Data	1 Sample	2 Samples
Categorical Data	$SE = \sqrt{\frac{\pi(1-\pi)}{n}}$ $\hat{p} \pm z^* \cdot \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$	$SE = \sqrt{\frac{\pi_1(1-\pi_1)}{n_1} + \frac{\pi_2(1-\pi_2)}{n_2}}$ $\hat{p_1} - \hat{p_2} \pm z^* \sqrt{\frac{\hat{p_1}(1-\hat{p_1})}{n_1} + \frac{\hat{p_2}(1-\hat{p_2})}{n_2}}$
Quantitative Data	$SE = \frac{s}{\sqrt{n}}$ $\overline{x} \pm t^* \frac{s}{\sqrt{n}}$	$SE = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$ $(\overline{x_1} - \overline{x_2}) \pm t^* \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$

Theories of hypothesis testing

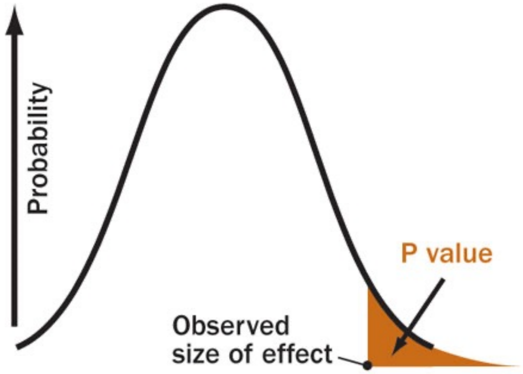
Fisher (1890-1962)

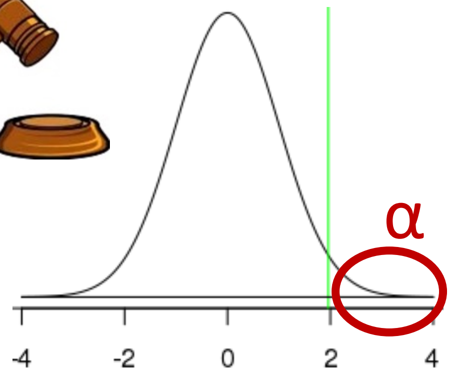
Neyman (1894-1981)

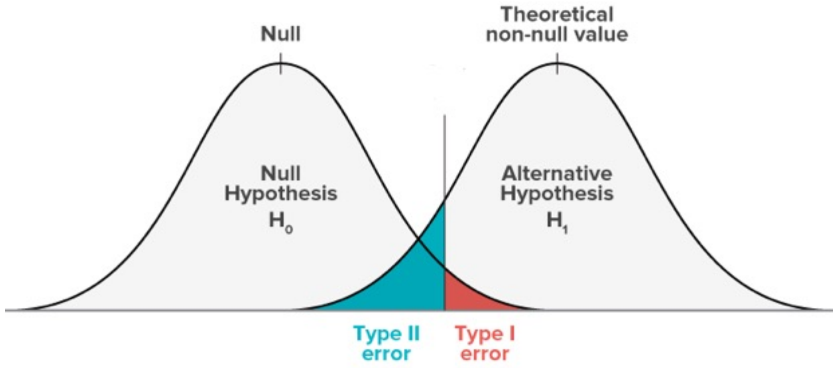
Pearson (1895-1980)

p-value a strength of evidence

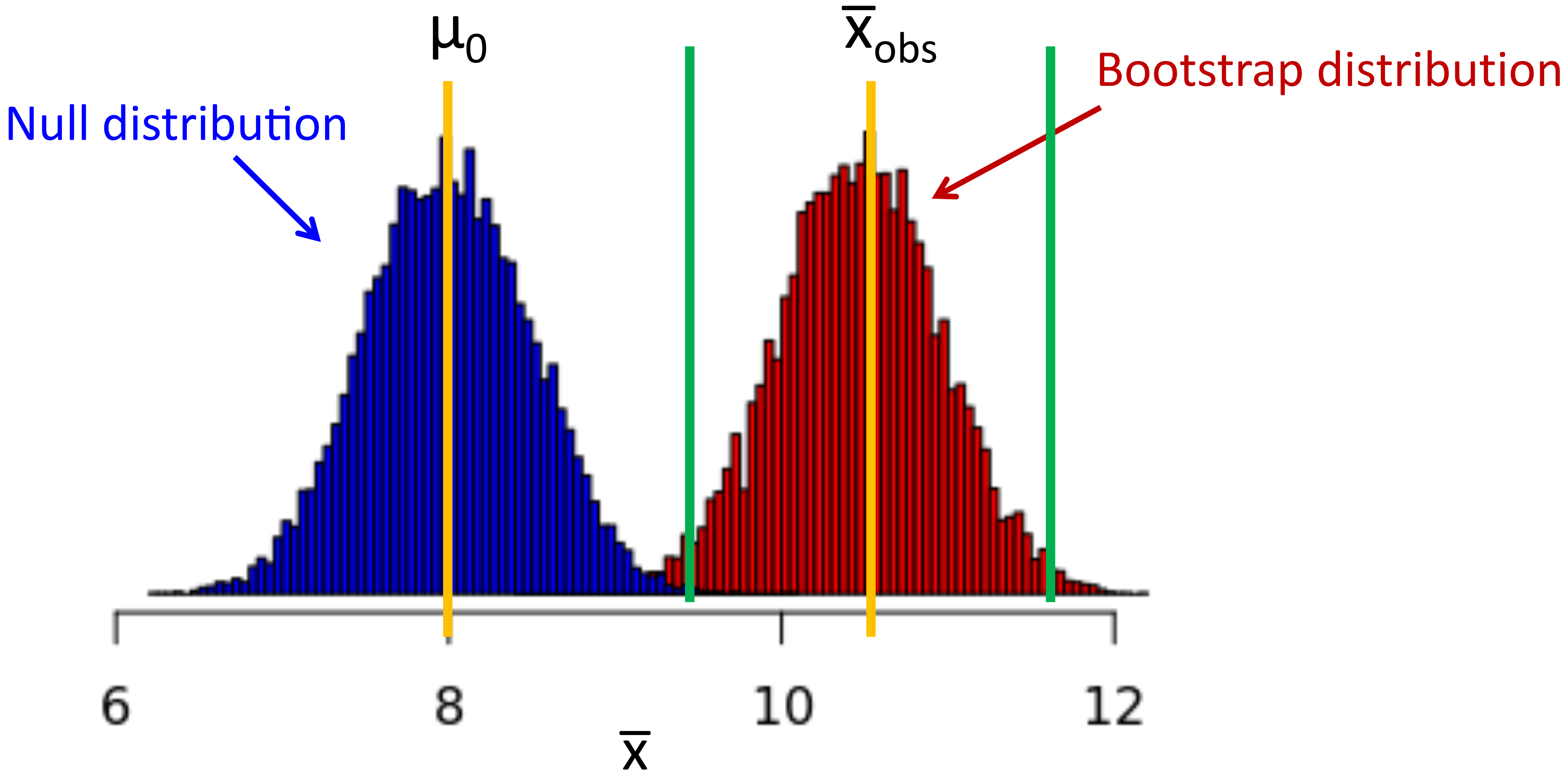
Use p-value to make a decision







Relationship between null and bootstrap distributions



Data manipulation with dplyr

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

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

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

age	body_type	diet
22	a little extra	strictly anything
35	average	mostly other
38	thin	anything
23	thin	vegetarian
29	athletic	*NA*
29	average	mostly anything

```
film_results <- movies |>
    filter(title_type == "Feature Film") |>
    select(critics_score, audience_score, genre) |>
     mutate(audience_prefers =
             audience_score - critics_score) |>
     group_by(genre) |>
      summarize(mean_audience_prefers =
              mean(audience_prefers)) |>
      arrange(desc(mean_audience_prefers))

head(film_results )
```

Grammar of graphics with ggplot

A Frame: Coordinate system on which data is placed

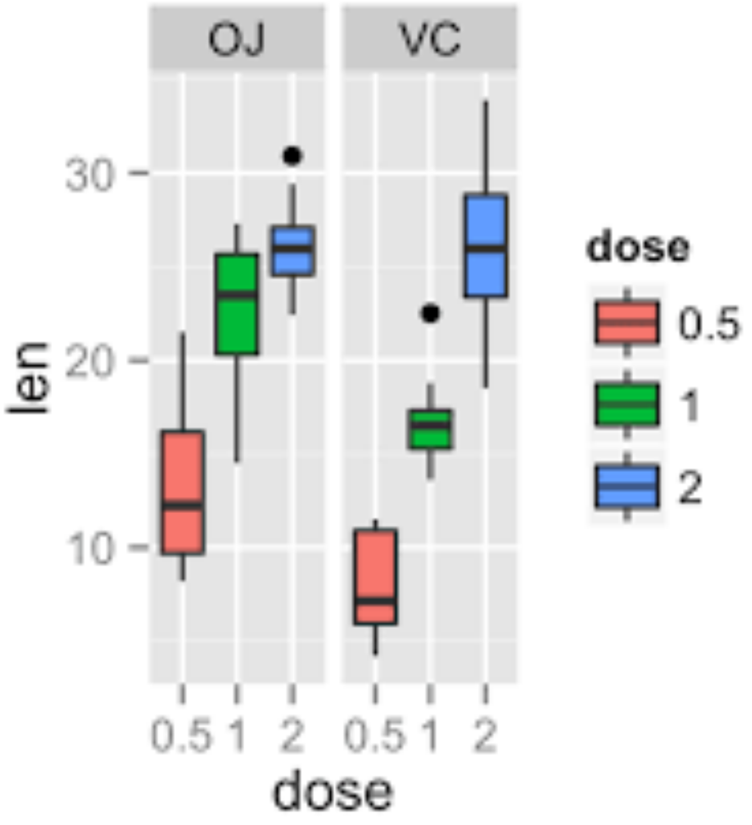
Glyphs: basic graphic unit representing cases or statistics

Scales and guides: shows how to interpret axes and other properties of the glyphs

Facets: allows for multiple side-by-side graphs based on a categorical variable

Layers: allows for more than one types of data to be mapped onto the same figure

Theme: contains finer points of display (e.g., font size, background color, etc.)



Questions