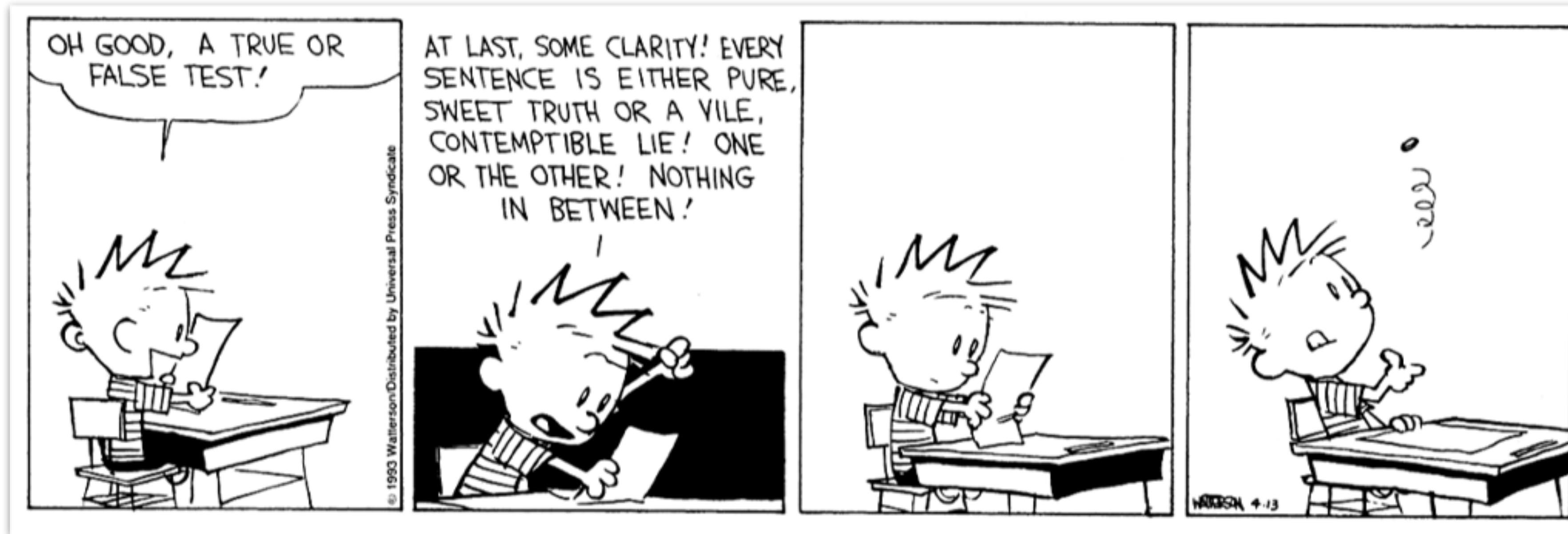


# Power Analysis 2 ... Model comparison



02/14/2025

# **Logistics**

# **Project proposal**

# Project proposal

IT'S TIME FOR A...

GROUP ASSIGNMENT!!

**Members of teams  
will all get the  
same grade!**



Didn't attend  
any group  
meetings



Doesn't  
understand  
the material



Gave the  
presentation  
but obviously  
didn't know  
what he was  
even saying

**maximum 3 team  
members**



Who is  
this guy

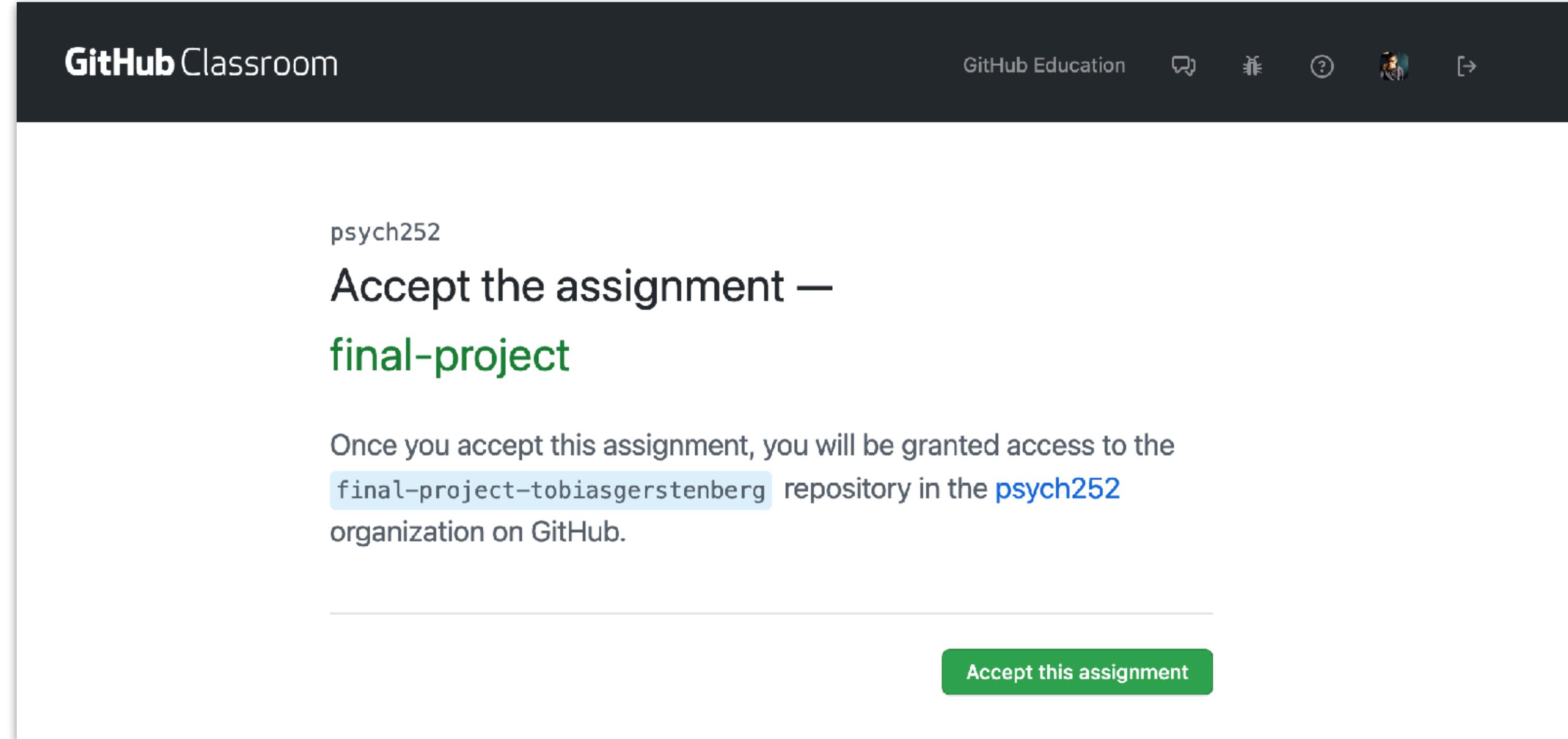


"You can  
use my  
printer"

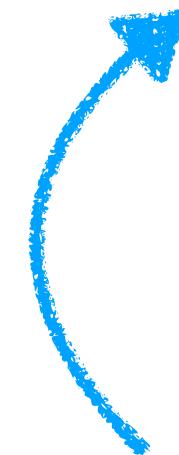


Did all the  
research, wrote  
paper, composed  
presentation

# Project proposal



The screenshot shows a GitHub Classroom interface. At the top, it says "GitHub Classroom". To the right are links for "GitHub Education", a user profile icon, and other navigation options. Below this, the text "psych252" is displayed. Underneath, there's a heading "Accept the assignment — final-project". A descriptive text follows: "Once you accept this assignment, you will be granted access to the [final-project-tobiasgerstenberg](#) repository in the [psych252](#) organization on GitHub." At the bottom right of the dialog is a green button labeled "Accept this assignment".



**if you work as a team, have one person accept the assignment and the others can then join that repository**

<https://tinyurl.com/psych252project25>

# Project proposal

The screenshot shows a GitHub repository page for 'psych252 / final-projects'. The repository is private, has 5 commits, 1 branch, 0 releases, and 1 contributor (tobiasgerstenberg). The commit history shows initial commits for various files like code/R, data, figures, papers, presentation, writeup, .gitignore, and README.md. Below the repository details, there's a 'Final project' section with a 'General points' subsection containing guidelines for folder and file names, relative paths, and organization. There's also a 'Repository structure' section.

**Final project**

Starter code for your final project.

**General points**

- for folder and file names:
  - don't use white space in either folder or filenames, use an underscore "\_" instead
  - (almost always) use lower case only
- always use relative paths in your code
  - for example, to save a figure from an R script inside the `code/R/` folder the path should be `"../../figures/figure_name.pdf"`
- keep your folder structure organized
  - we recommend adhering to the folder structure in this repository
  - more complex projects may have additional folders such as `videos/`, `papers/`, ...
- note: some of the folders are empty except for a `.keep` file
  - the `.keep` file is just there to make sure that github includes the otherwise empty folder
  - feel free to delete the `.keep` file once you've added another file to that folder

**Repository structure**

- each team will have their own private github repository
- all work on your final project should happen within this repository
- you can get **github** help in section
- post on Ed Discussion in case you experience any problems getting set up (use **final\_project** tag)

# Project proposal

# RMarkdown template



The screenshot shows the RStudio interface with the following details:

- Project Area:** Shows a project named "final-projects" containing files: "project\_proposal.Rmd", "project\_report.html", "project\_report.Rmd", "psych252-final-project.Rproj", and "references".
- Code Editor:** The file "project\_proposal.Rmd" is open, showing R Markdown code. The code includes metadata (title, subtitle, author, date, urlcolor) and instructions for the project proposal. It specifies the output format as PDF and contains a section for research questions.
- Console:** The R console shows the command "R 4.1.2" and the current working directory as the path to the project folder.
- Environment:** The environment pane shows that the environment is empty.

psych252/final-projects/code/R/project\_proposal.Rmd

Upload the  
pdf to canvas

# Project proposal

The project proposal is due on  
**Thursday, February 20<sup>th</sup> at 8pm**

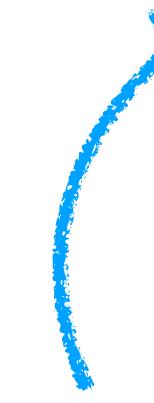
# Project proposal

W22-PSYCH-252-01 > Files > final\_project > proposal > 2021

Search for files Q 0 items selected + Folder Upload ⋮

| Name ▲               | Date Created | Date Modified | Modified By      | Size   | ⋮ |
|----------------------|--------------|---------------|------------------|--------|---|
| anjie_cao.pdf        | 2:59pm       | 2:59pm        | Tobias Gerste... | 236 KB |   |
| danyang_fan.pdf      | 2:59pm       | 2:59pm        | Tobias Gerste... | 259 KB |   |
| hannah_marshall.pdf  | 2:59pm       | 2:59pm        | Tobias Gerste... | 238 KB |   |
| madi_jamie_catie.pdf | 2:59pm       | 2:59pm        | Tobias Gerste... | 247 KB |   |

8% of 5.2 GB used All My Files

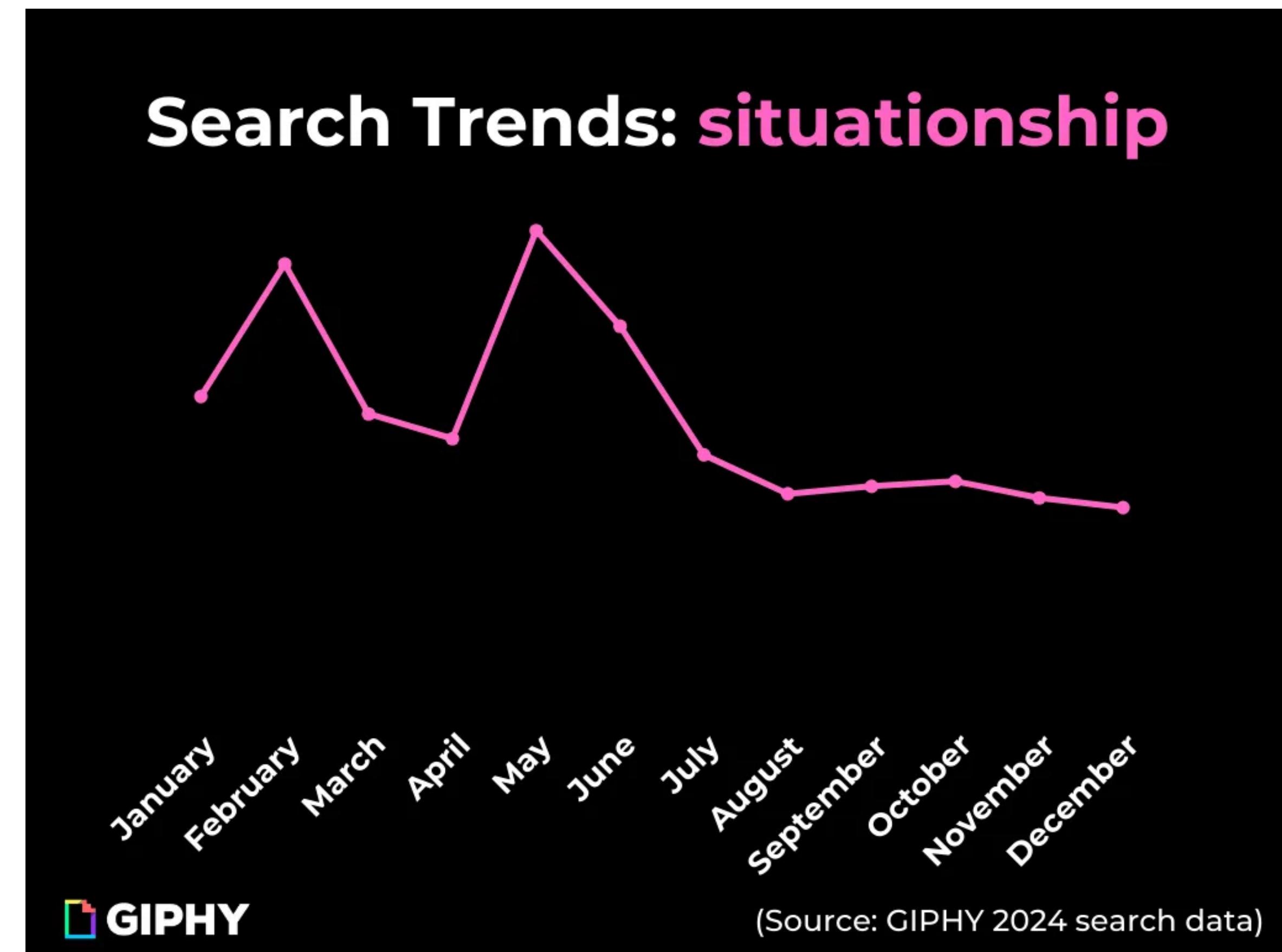


examples of project proposals from prior years

# **Things that came up**

# Digital Traces

Google Maps searches for flower shops spike nearly 120% leading up to February 14. ... In the U.S., February 13 marks the day people most use Google Maps to search for “romantic restaurants.”





# Plan for today

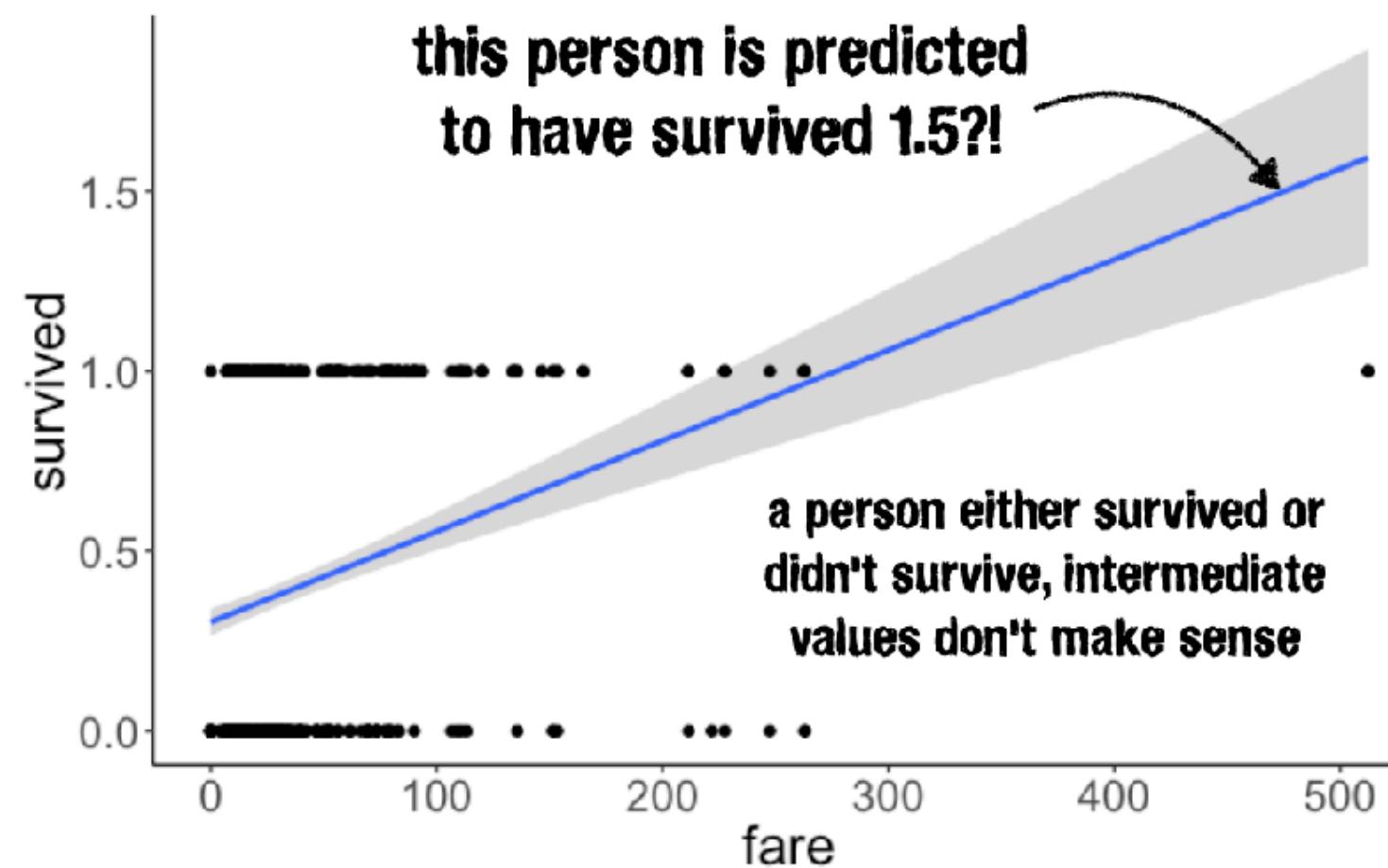
- Quick recap
- Power analysis
  - Making decisions
  - Calculating power
  - Effect sizes
  - Determining sample size
- Simulating a power analysis
  - Demonstration in RStudio
- Model comparison
  - Cross-validation
  - AIC and BIC

# Quick recap

# Quick recap: Logistic regression

Is there a relationship between fare and survived?

```
fit.lm = lm(formula = survived ~ 1 + fare, data = df.titanic)
```



22

## Fitting a logistic regression in R

```
1 fit.glm = glm(formula = survived ~ 1 + fare,
2                      family = "binomial",
3                      data = df.titanic)
4
5 fit.glm %>% summary()
```

```
Call:
glm(formula = survived ~ 1 + fare, family = "binomial", data = df.titanic)

Deviance Residuals:
    Min      10     Median      30      Max
-2.4906 -0.8878 -0.8531  1.3429  1.5942

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.941330  0.095129 -9.895 < 2e-16 ***
fare         0.015197  0.002232  6.810 9.79e-12 ***

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1186.7 on 890 degrees of freedom
Residual deviance: 1117.6 on 889 degrees of freedom
AIC: 1121.6

Number of Fisher Scoring iterations: 4
```

27

## Logit transform

$$\pi_i = b_0 + b_1 \cdot X_i + e_i \quad \text{predict the probability of Y}$$

$$\pi_i = P(Y_i = 1)$$

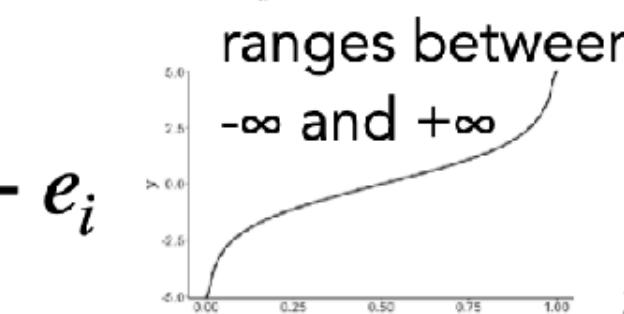
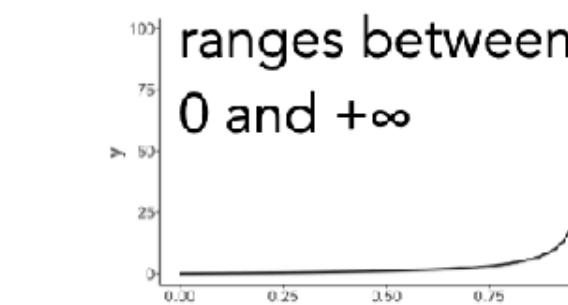
### Step 1: Calculate the "odds"

$$\frac{P(Y_i = 1)}{P(Y_i = 0)} = \frac{\pi_i}{1 - \pi_i}$$

### Step 2: Take the (natural) log

$$\ln\left(\frac{\pi_i}{1 - \pi_i}\right) = b_0 + b_1 \cdot X_i + e_i$$

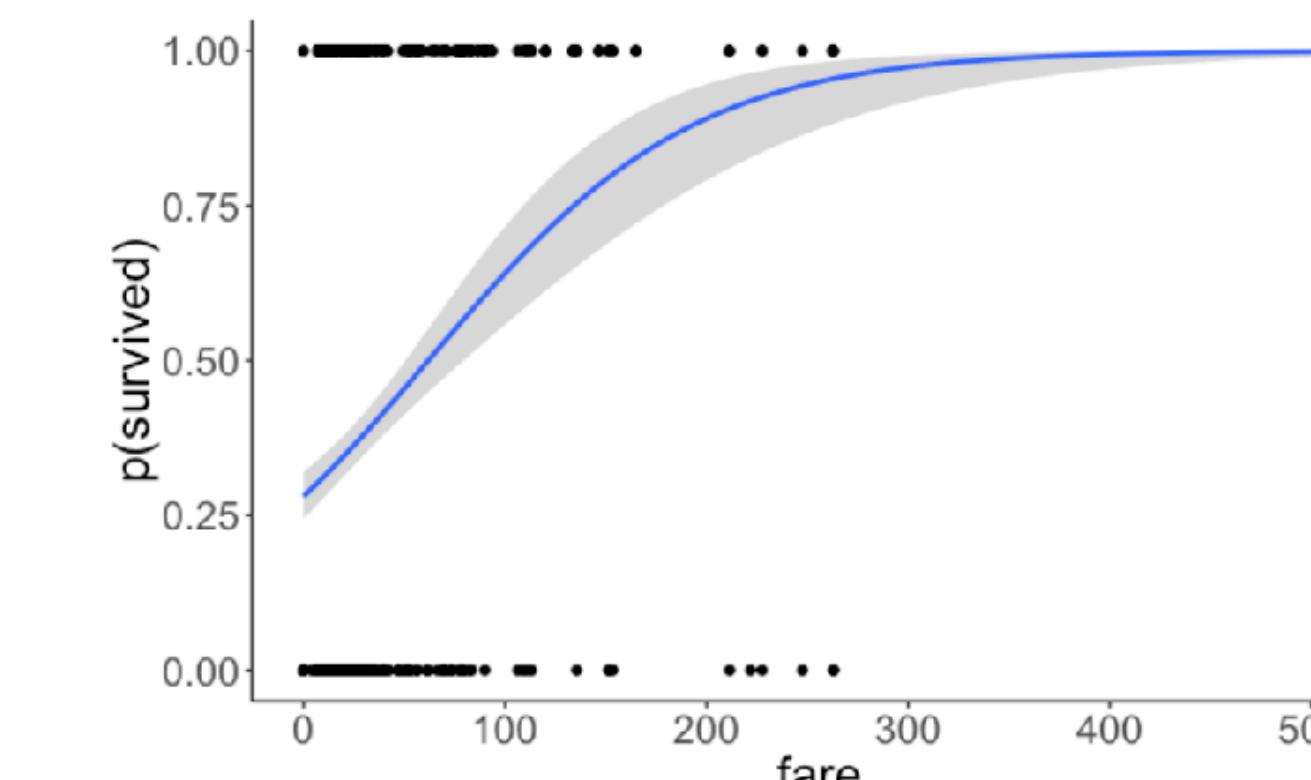
we need to transform the dependent variable so that it can take any value between  $-\infty$  and  $+\infty$  (we can then transform it back into a probability later)



25

## Visualize the model's predictions

```
1 ggplot(data = df.titanic,
2           mapping = aes(x = fare,
3                           y = survived)) +
4   geom_smooth(method = "glm",
5               method.args = list(family = "binomial")) +
6   geom_point() +
7   labs(y = "p(survived)")
```



28

15

# Quick recap: Logistic regression

```
1 fit.glm3 = glm(formula = survived ~ 1 + sex * fare,  
2                      family = "binomial",  
3                      data = df.titanic)  
4  
5 fit.glm3 %>%  
6   summary()
```

Call:  
`glm(formula = survived ~ 1 + sex * fare, family = "binomial",  
 data = df.titanic)`

Deviance Residuals:

| Min     | 1Q      | Median  | 3Q     | Max    |
|---------|---------|---------|--------|--------|
| -2.6280 | -0.6279 | -0.5991 | 0.8172 | 1.9288 |

Coefficients:

|              | Estimate  | Std. Error | z value | Pr(> z )     |
|--------------|-----------|------------|---------|--------------|
| (Intercept)  | 0.408428  | 0.189999   | 2.150   | 0.031584 *   |
| sexmale      | -2.099345 | 0.230291   | -9.116  | < 2e-16 ***  |
| fare         | 0.019878  | 0.005372   | 3.701   | 0.000215 *** |
| sexmale:fare | -0.011617 | 0.005934   | -1.958  | 0.050252 .   |

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

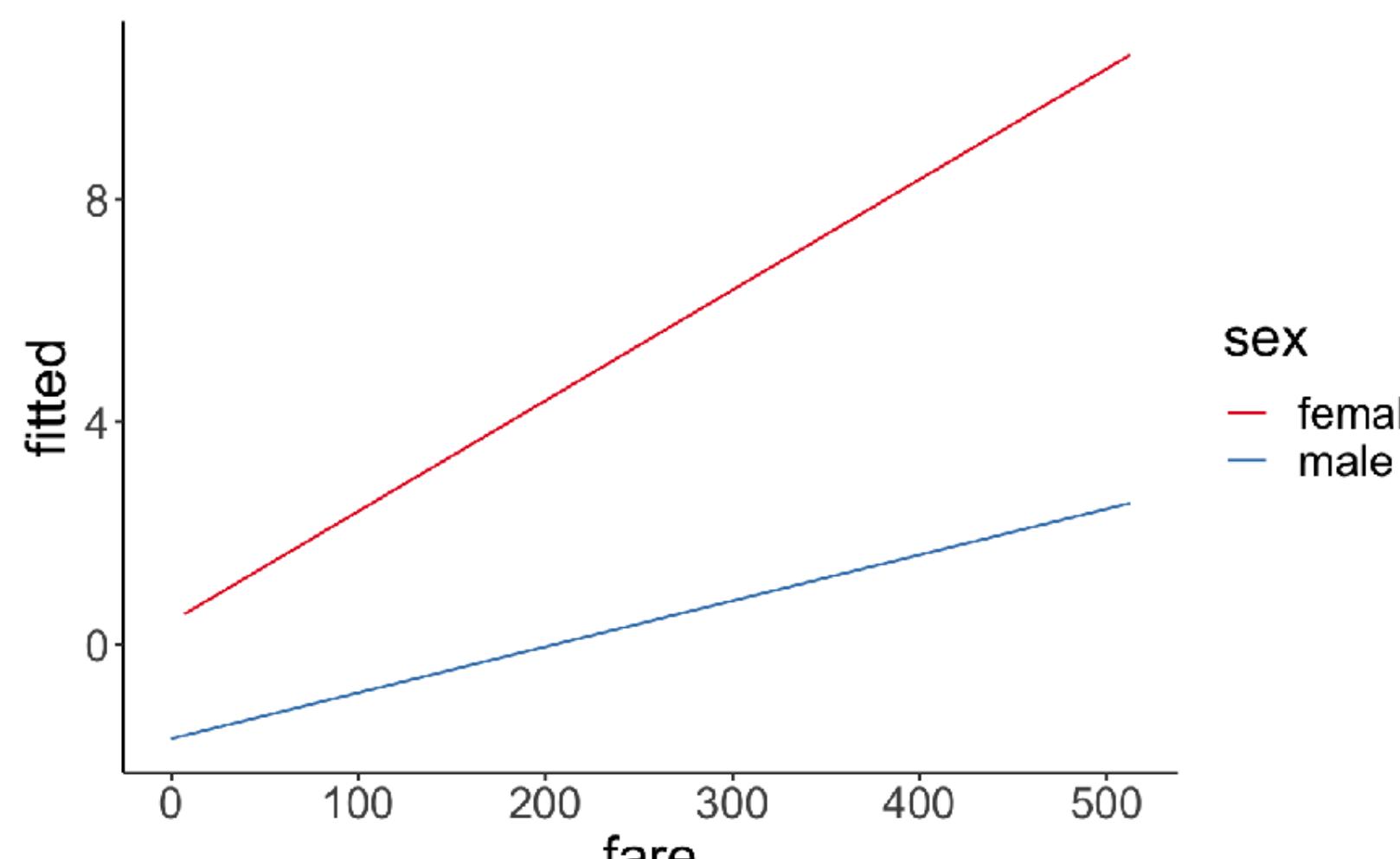
(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1186.66 on 890 degrees of freedom

Residual deviance: 879.85 on 887 degrees of freedom

AIC: 887.85

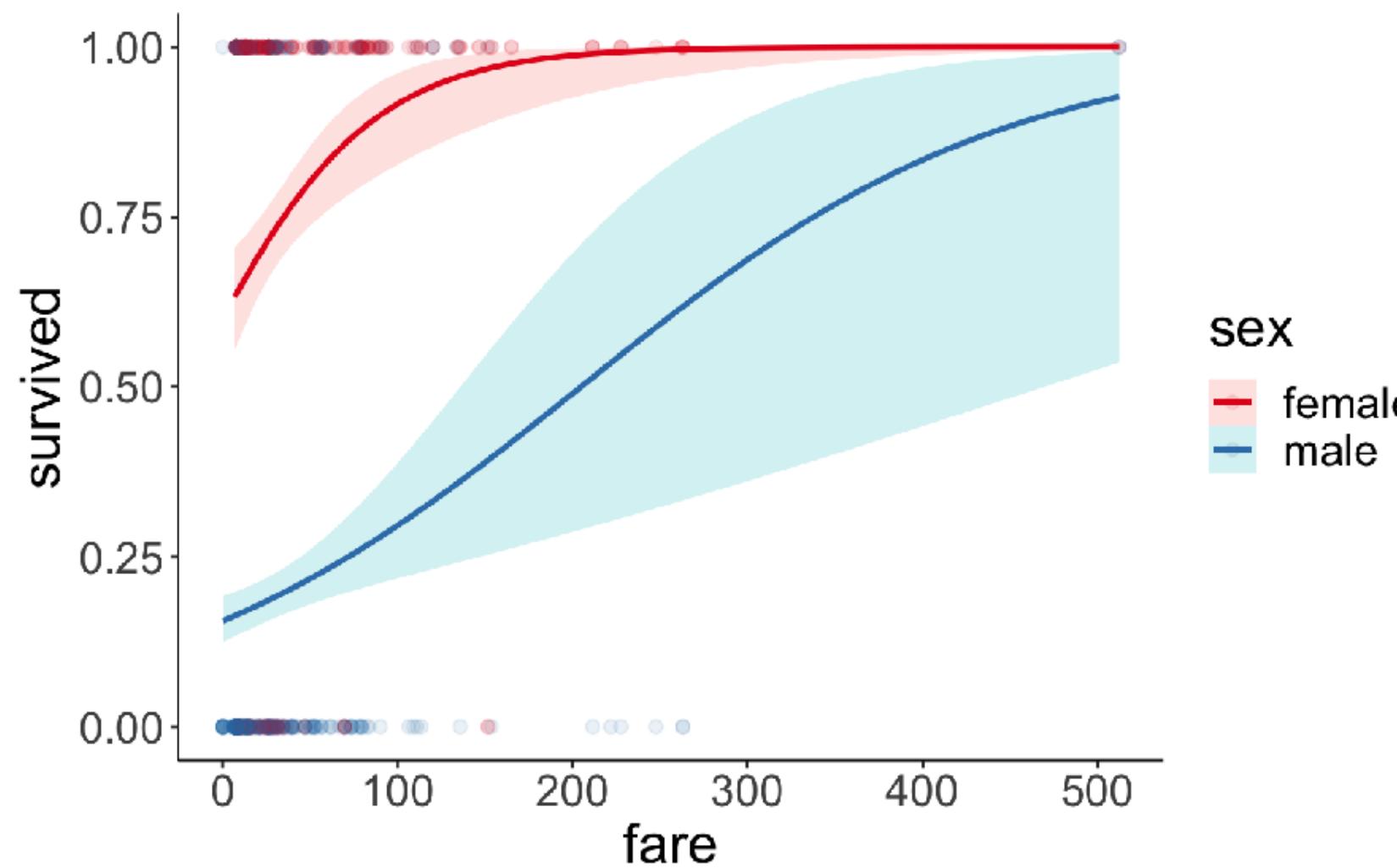
Number of Fisher Scoring iterations: 5



lines in log odds  
space

# Quick recap: Logistic regression

```
1 fit.glm3 = glm(formula = survived ~ 1 + sex * fare,  
2                         family = "binomial",  
3                         data = df.titanic)  
4  
5 fit.glm3 %>%  
6   summary()
```



after inverse logit  
transformation

Call:  
`glm(formula = survived ~ 1 + sex * fare, family = "binomial",  
 data = df.titanic)`

Deviance Residuals:

| Min     | 1Q      | Median  | 3Q     | Max    |
|---------|---------|---------|--------|--------|
| -2.6280 | -0.6279 | -0.5991 | 0.8172 | 1.9288 |

Coefficients:

|              | Estimate  | Std. Error | z value | Pr(> z )     |
|--------------|-----------|------------|---------|--------------|
| (Intercept)  | 0.408428  | 0.189999   | 2.150   | 0.031584 *   |
| sexmale      | -2.099345 | 0.230291   | -9.116  | < 2e-16 ***  |
| fare         | 0.019878  | 0.005372   | 3.701   | 0.000215 *** |
| sexmale:fare | -0.011617 | 0.005934   | -1.958  | 0.050252 .   |
| ---          |           |            |         |              |

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1186.66 on 890 degrees of freedom  
Residual deviance: 879.85 on 887 degrees of freedom  
AIC: 887.85

Number of Fisher Scoring iterations: 5

# Quick recap: Interpreting the model output

Let's consider a binary predictor

```
1 fit.glm2 = glm(formula = survived ~ sex,
2                  family = "binomial",
3                  data = df.titanic)
4
5 fit.glm2 %>% summary()
```

```
Call:
glm(formula = survived ~ sex, family = "binomial", data = df.titanic)

Deviance Residuals:
    Min      1Q  Median      3Q     Max 
-1.6462 -0.6471 -0.6471  0.7725  1.8256 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)  1.0566    0.1290   8.191 2.58e-16 ***
sexmale     -2.5137    0.1672 -15.036 < 2e-15 ***
---
Signif. codes:  0 '*****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1186.7 on 890 degrees of freedom
Residual deviance: 917.8 on 889 degrees of freedom
AIC: 921.8

Number of Fisher Scoring Iterations: 4
```

**sex was significantly associated with survival**

35

**Was the probability of survival different between female and male passengers on the Titanic?**



Let's consider a binary predictor

$$\ln\left(\frac{p(\text{survived})_i}{1 - p(\text{survived})_i}\right) = b_0 + b_1 \cdot \text{sex}_i$$

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)  1.0566    0.1290   8.191 2.58e-16 ***
sexmale     -2.5137    0.1672 -15.036 < 2e-15 ***
```

| sex    | survived | n   | p    | p(survived   sex) |
|--------|----------|-----|------|-------------------|
| female | 0        | 81  | 0.09 | 0.26              |
| female | 1        | 233 | 0.26 | 0.74              |
| male   | 0        | 468 | 0.53 | 0.81              |
| male   | 1        | 109 | 0.12 | 0.19              |

**if sex == 0:**

$$\ln\left(\frac{\widehat{p(\text{survived})}_i}{1 - \widehat{p(\text{survived})}_i}\right) = b_0$$

$$p(\text{survived})_i = \frac{e^{b_0}}{1 + e^{b_0}} = 0.74$$

36

Now let's go back to a continuous predictor

$$\ln\left(\frac{p(\text{survived})_i}{1 - p(\text{survived})_i}\right) = b_0 + b_1 \cdot \text{fare}_i$$

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) -0.941330  0.095129 -9.895 < 2e-16 ***
fare         0.015197  0.002232  6.810 9.79e-12 ***
```

| fare | prediction | p(survival) |
|------|------------|-------------|
| 0    | -0.94      | 0.28        |
| 10   | -0.79      | 0.31        |
| 50   | -0.18      | 0.45        |
| 100  | 0.58       | 0.64        |
| 500  | 6.66       | 1.00        |

$$\ln\left(\frac{\widehat{p(\text{survived})}}{1 - \widehat{p(\text{survived})}}\right) = -0.94 + 0.015 \cdot 10$$

$$p(\text{survived})_i = \frac{e^{-0.94+0.015 \cdot 10}}{1 + e^{-0.94+0.015 \cdot 10}} = 0.31$$

38

18

# Quick recap: Fitting and reporting models

## Simulating a logistic regression

```

1 # make example reproducible
2 set.seed(1)
3
4 # set parameters
5 sample_size = 1000
6 b0 = 0
7 b1 = 1
8
9 # generate data
10 df.data = tibble(
11   x = rnorm(n = sample_size),
12   y = b0 + b1 * x,
13   p = inv.logit(y) %>%
14     mutate(response = rbinom(n(), size = 1, p = p))
15
16 # fit model
17 fit = glm(formula = response ~ 1 + x,
18            family = "binomial",
19            data = df.data)
20
21 # model summary
22 fit %>% summary()

```

set some parameters  
 linear model (y is in log odds)  
 transform into probability  
 randomly draw response  
 fit a logistic regression  
 summarize the result

45

## Assessing the model fit

actual value      predicted value

$$\text{log-likelihood} = \sum_{i=1}^n [Y_i \cdot \ln(P(Y_i)) + (1 - Y_i) \cdot \ln(1 - P(Y_i))]$$

- calculate the probability of the observed response
- take the log of these probabilities
- sum them up to get the log-likelihood of the data (given the model)

| response | p(Y = 1) | p(Y = response) | log(p(Y = response)) |
|----------|----------|-----------------|----------------------|
| 1        | 0.34     | 0.34            | -1.07                |
| 0        | 0.53     | 0.47            | -0.75                |
| 1        | 0.30     | 0.30            | -1.20                |
| 1        | 0.81     | 0.81            | -0.22                |
| 1        | 0.56     | 0.56            | -0.58                |
| 0        | 0.30     | 0.70            | -0.36                |
| 1        | 0.90     | 0.60            | -0.52                |
| 1        | 0.55     | 0.65            | -0.43                |
| 1        | 0.52     | 0.62            | -0.48                |
| 0        | 0.41     | 0.59            | -0.54                |

47

## Testing hypotheses

`joint_tests()`

```

1 glm(formula = survived ~ 1 + sex + fare,
2   family = "binomial",
3   data = df.titanic) %>%
4   joint_tests()

```

| model term | df1 | df2 | F.ratio | p.value |
|------------|-----|-----|---------|---------|
| sex        | 1   | Inf | 201.881 | <.0001  |
| fare       | 1   | Inf | 23.869  | <.0001  |

not quite the same value

`anova(test = "LRT")`

```

1 # fit compact model
2 fit.compact = glm(formula = survived ~ 1 + fare,
3                    family = "binomial",
4                    data = df.titanic)
5
6 # fit augmented model
7 fit.augmented = glm(formula = survived ~ 1 + sex + fare,
8                      family = "binomial",
9                      data = df.titanic)
10
11 # likelihood ratio test
12 anova(fit.compact, fit.augmented, test = "LRT")

```

Analysis of Deviance Table

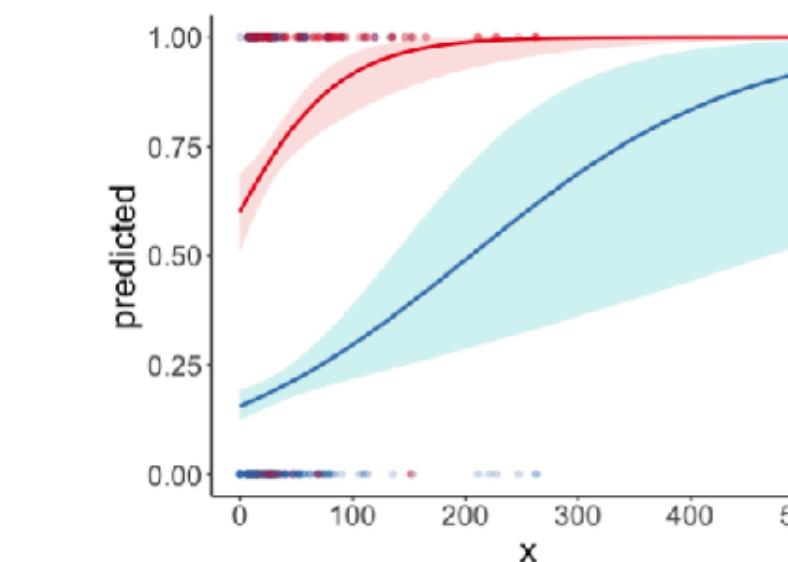
| Model:         | survived ~ 1 + fare | Model:    | survived ~ 1 + sex + fare |
|----------------|---------------------|-----------|---------------------------|
| Resid. Df      | 889                 | Resid. Df | 888                       |
| Deviance       | 1117.57             | Deviance  | 884.31                    |
|                |                     |           | 233.26 < 2.2e+16 ***      |
| Signif. codes: | 0 ****              | 0.001 *** | 0.01 **                   |
|                | 0.05 *              | 0.1 .     | 1                         |

<https://stats.stackexchange.com/questions/400101/using-emmeans-with-clmm-to-look-at-joint-effects>

51

## Reporting results

- Visualize the data
- Show a table with the regression results
- Report significance of different factors
- Interpreting parameter estimates is tricky -- probably best to report probabilities for a few example cases



| #   | x   | Predicted | SE   | 95% CI       |
|-----|-----|-----------|------|--------------|
| 1   | 0   | 0.50      | 0.12 | [0.15, 0.69] |
| 100 | 100 | 0.32      | 0.42 | [0.83, 0.96] |
| 200 | 200 | 0.39      | 0.95 | [0.53, 1.00] |
| 300 | 300 | 1.00      | 1.78 | [0.97, 1.00] |
| 400 | 400 | 1.00      | 2.02 | [0.99, 1.00] |
| 500 | 500 | 1.00      | 2.53 | [1.00, 1.00] |

| #   | x   | Predicted | SE   | 95% CI       |
|-----|-----|-----------|------|--------------|
| 1   | 0   | 0.15      | 0.12 | [0.12, 0.19] |
| 100 | 100 | 0.30      | 0.21 | [0.22, 0.39] |
| 200 | 200 | 0.49      | 0.24 | [0.73, 0.79] |
| 300 | 300 | 0.59      | 0.63 | [0.25, 0.90] |
| 400 | 400 | 0.83      | 0.94 | [0.44, 0.97] |
| 500 | 500 | 0.92      | 1.13 | [0.53, 0.99] |

54      19

# Assessing the model fit

actual value

predicted value

$$\text{log-likelihood} = \sum_{i=1}^n [Y_i \cdot \ln(P(Y_i)) + (1 - Y_i) \cdot \ln(1 - P(Y_i))]$$

- calculate the probability of the observed response
- take the log of these probabilities
- sum them up to get the log-likelihood of the data (given the model)

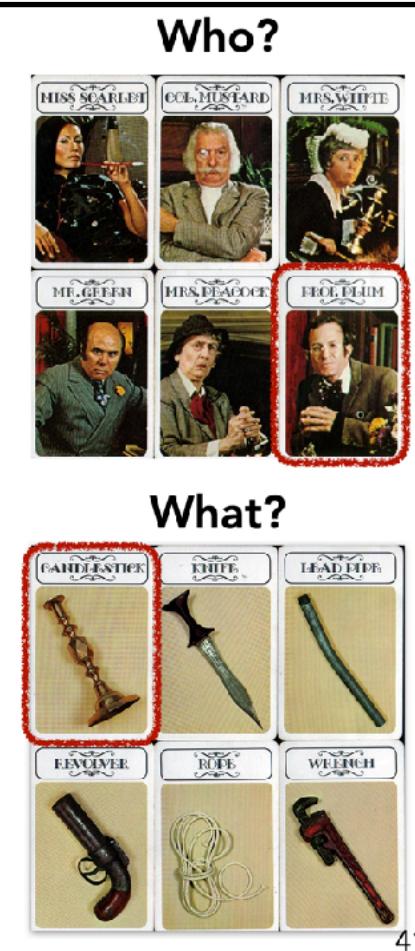
## Clue guide to probability

- joint probability:
- if A and B are independent then
- Definition:  $p(A, B) = p(A) \cdot p(B)$

$p(\text{Prof Plum, candle stick}) =$

$p(\text{Prof Plum}) \cdot p(\text{candle stick}) =$

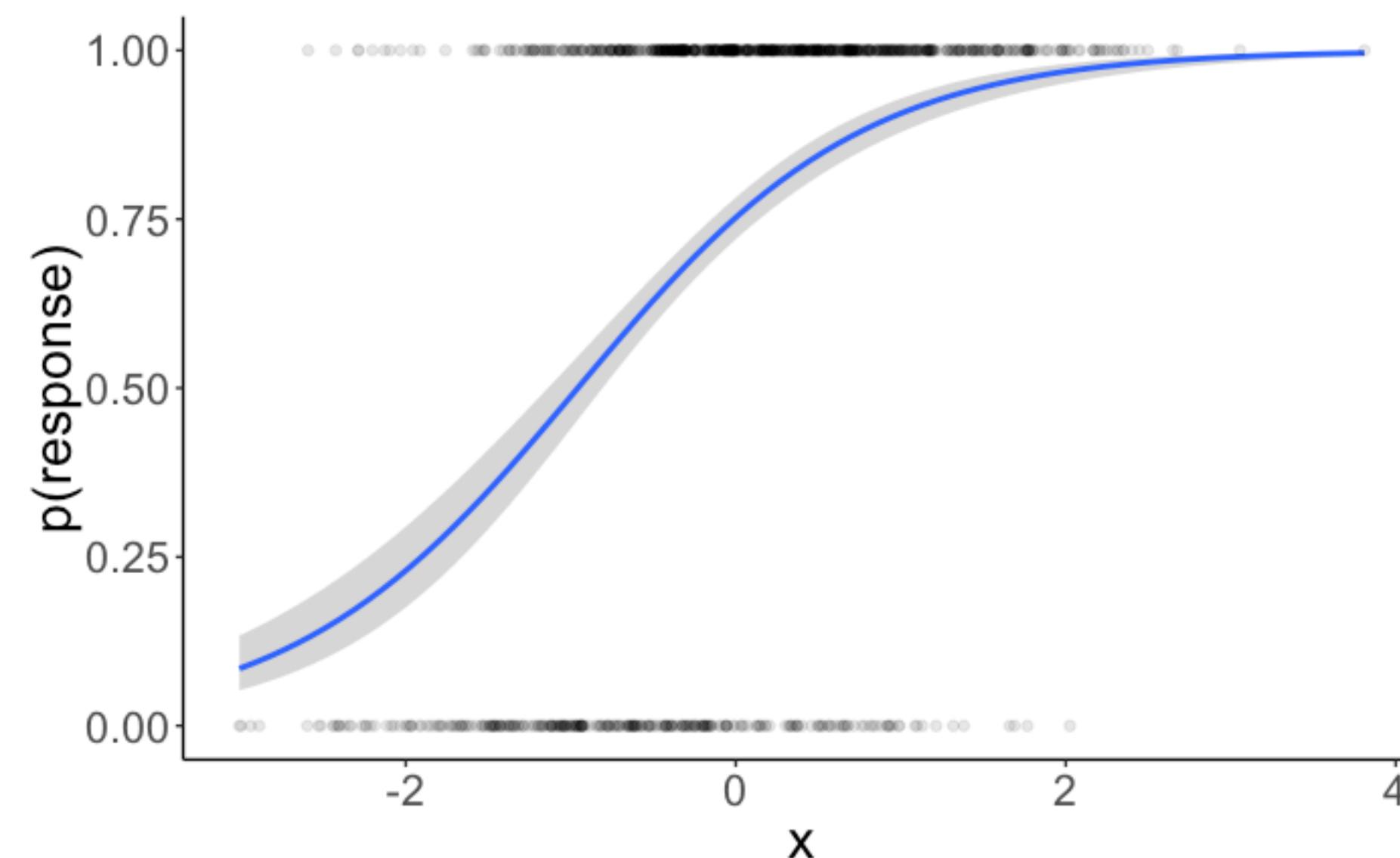
$$\frac{1}{6} \cdot \frac{1}{6} = \frac{1}{36}$$



| response | $p(Y = 1)$ | $p(Y = \text{response})$ | $\log(p(Y = \text{response}))$ |
|----------|------------|--------------------------|--------------------------------|
| 1        | 0.34       | 0.34                     | -1.07                          |
| 0        | 0.53       | 0.47                     | -0.75                          |
| 1        | 0.30       | 0.30                     | -1.20                          |
| 1        | 0.81       | 0.81                     | -0.22                          |
| 1        | 0.56       | 0.56                     | -0.58                          |
| 0        | 0.30       | 0.70                     | -0.36                          |
| 1        | 0.60       | 0.60                     | -0.52                          |
| 1        | 0.65       | 0.65                     | -0.43                          |
| 1        | 0.62       | 0.62                     | -0.48                          |
| 0        | 0.41       | 0.59                     | -0.54                          |

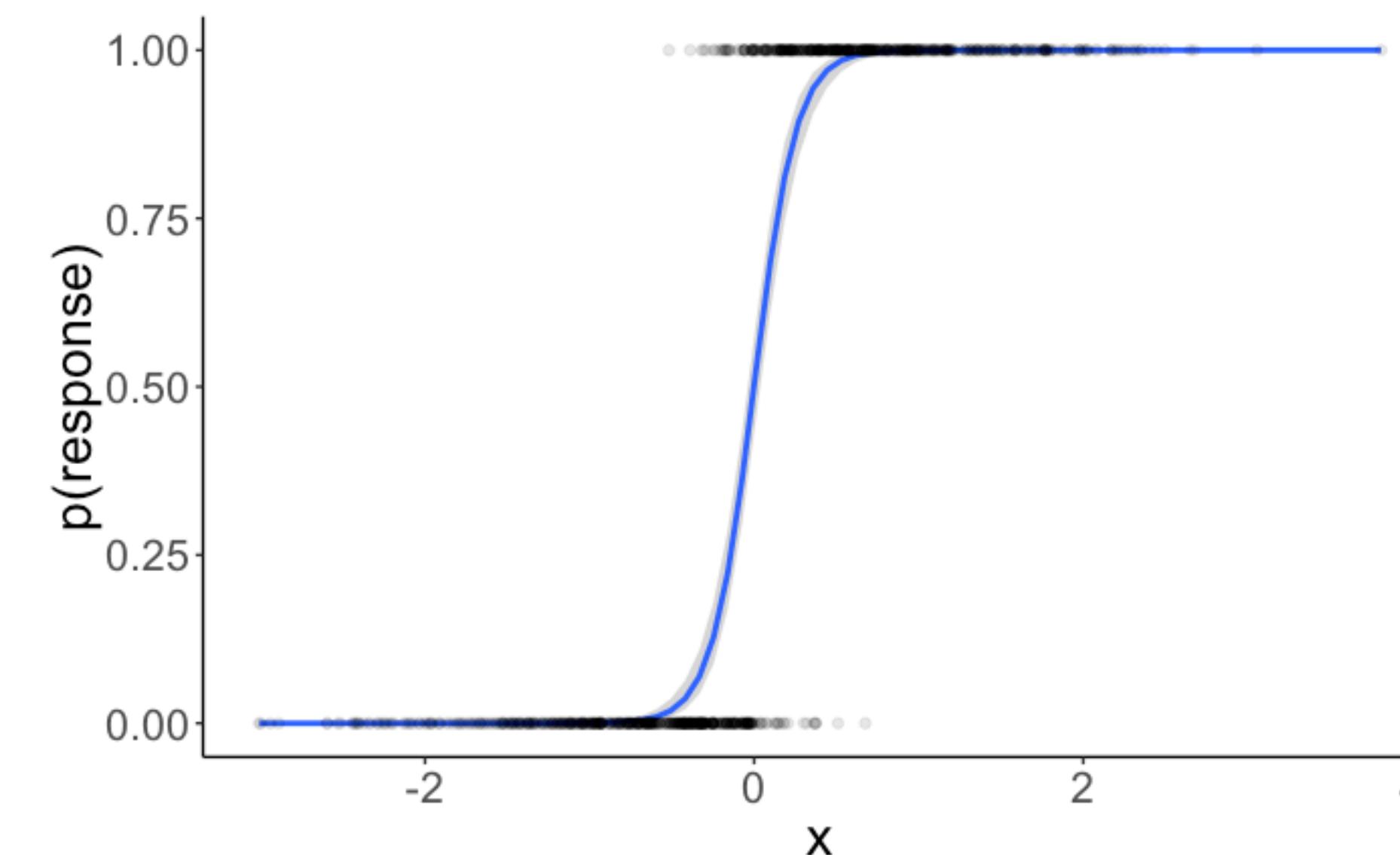
# Assessing the model fit

doesn't predict the response very well



| logLik  | AIC    | BIC     |
|---------|--------|---------|
| -501.65 | 1007.3 | 1017.12 |

predicts the response much better



| logLik  | AIC    | BIC    |
|---------|--------|--------|
| -156.37 | 316.74 | 326.55 |

# Testing hypotheses

aka checking  
whether it's **worth it**

```
1 # fit compact model
2 fit.compact = glm(formula = survived ~ 1 + fare,
3                         family = "binomial",
4                         data = df.titanic)
5
6 # fit augmented model
7 fit.augmented = glm(formula = survived ~ 1 + sex + fare,
8                         family = "binomial",
9                         data = df.titanic)
10
11 # likelihood ratio test
12 anova(fit.compact, fit.augmented, test = "LRT")
```

we need to specify that we  
want a likelihood ratio test

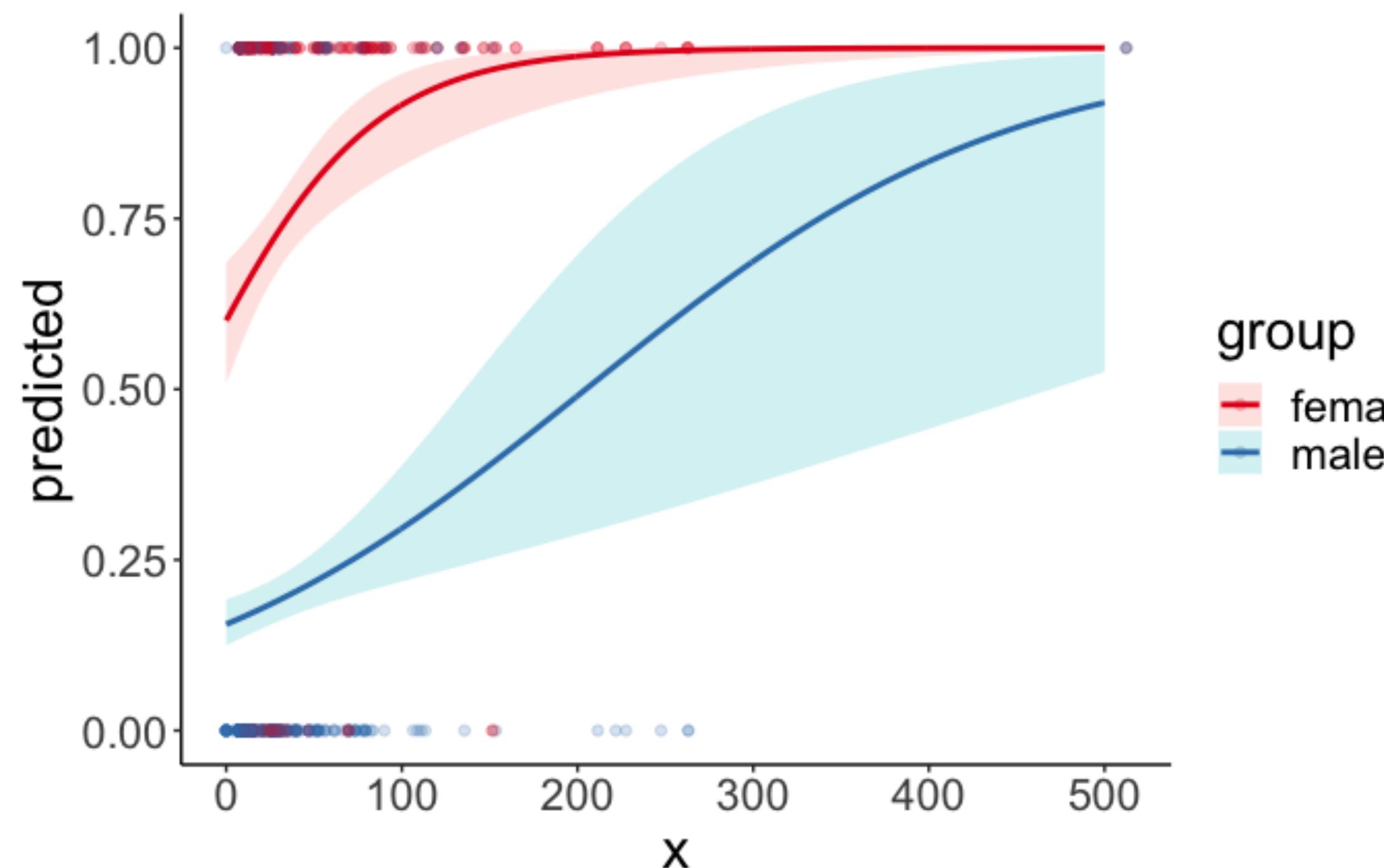
```
Analysis of Deviance Table

Model 1: survived ~ 1 + fare
Model 2: survived ~ 1 + sex + fare
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       889    1117.57
2       888    884.31  1     233.26 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Reporting results



- Visualize the data
- Show a table with the regression results (log-odds)
- Report significance of different factors
- Interpreting parameter estimates is tricky -- probably best to report probabilities for a few example cases



| # Predicted values of survived |           |           |              |              |
|--------------------------------|-----------|-----------|--------------|--------------|
| #                              | x         | Predicted | SE           | 95% CI       |
| # x = fare                     |           |           |              |              |
| 1                              | 0         | 0.60      | 0.19         | [0.51, 0.69] |
| 2                              | 100       | 0.92      | 0.42         | [0.83, 0.96] |
| 3                              | 200       | 0.99      | 0.95         | [0.93, 1.00] |
| 4                              | 300       | 1.00      | 1.48         | [0.97, 1.00] |
| 5                              | 400       | 1.00      | 2.02         | [0.99, 1.00] |
| 6                              | 500       | 1.00      | 2.55         | [1.00, 1.00] |
| # sex = female                 |           |           |              |              |
| x                              | Predicted | SE        | 95% CI       |              |
| 0                              | 0.60      | 0.19      | [0.51, 0.69] |              |
| 100                            | 0.92      | 0.42      | [0.83, 0.96] |              |
| 200                            | 0.99      | 0.95      | [0.93, 1.00] |              |
| 300                            | 1.00      | 1.48      | [0.97, 1.00] |              |
| 400                            | 1.00      | 2.02      | [0.99, 1.00] |              |
| 500                            | 1.00      | 2.55      | [1.00, 1.00] |              |
| # sex = male                   |           |           |              |              |
| x                              | Predicted | SE        | 95% CI       |              |
| 0                              | 0.16      | 0.13      | [0.12, 0.19] |              |
| 100                            | 0.30      | 0.21      | [0.22, 0.39] |              |
| 200                            | 0.49      | 0.44      | [0.29, 0.70] |              |
| 300                            | 0.69      | 0.69      | [0.36, 0.90] |              |
| 400                            | 0.83      | 0.94      | [0.44, 0.97] |              |
| 500                            | 0.92      | 1.19      | [0.53, 0.99] |              |

# Assumptions

- linearity (between predictors and log odds)
- independence
- no multi-collinearity
- model fails to converge when there is  
**complete separation:**
  - if outcome variable can be perfectly predicted by a (combination of) predictor(s)

# Different kinds of generalized models

## Different linking functions

```
binomial(link = "logit")  
  
gaussian(link = "identity")  
  
Gamma(link = "inverse")  
  
inverse.gaussian(link = "1/mu^2")  
  
poisson(link = "log")  
  
quasi(link = "identity", variance = "constant")  
  
quasibinomial(link = "logit")  
  
quasipoisson(link = "log")
```

**apply different transformations to the  
dependent variable**

# **Power analysis**

# Making decisions

## Type I Error



## Type II Error



$H_0$ : Not pregnant.     $H_1$ : Pregnant.

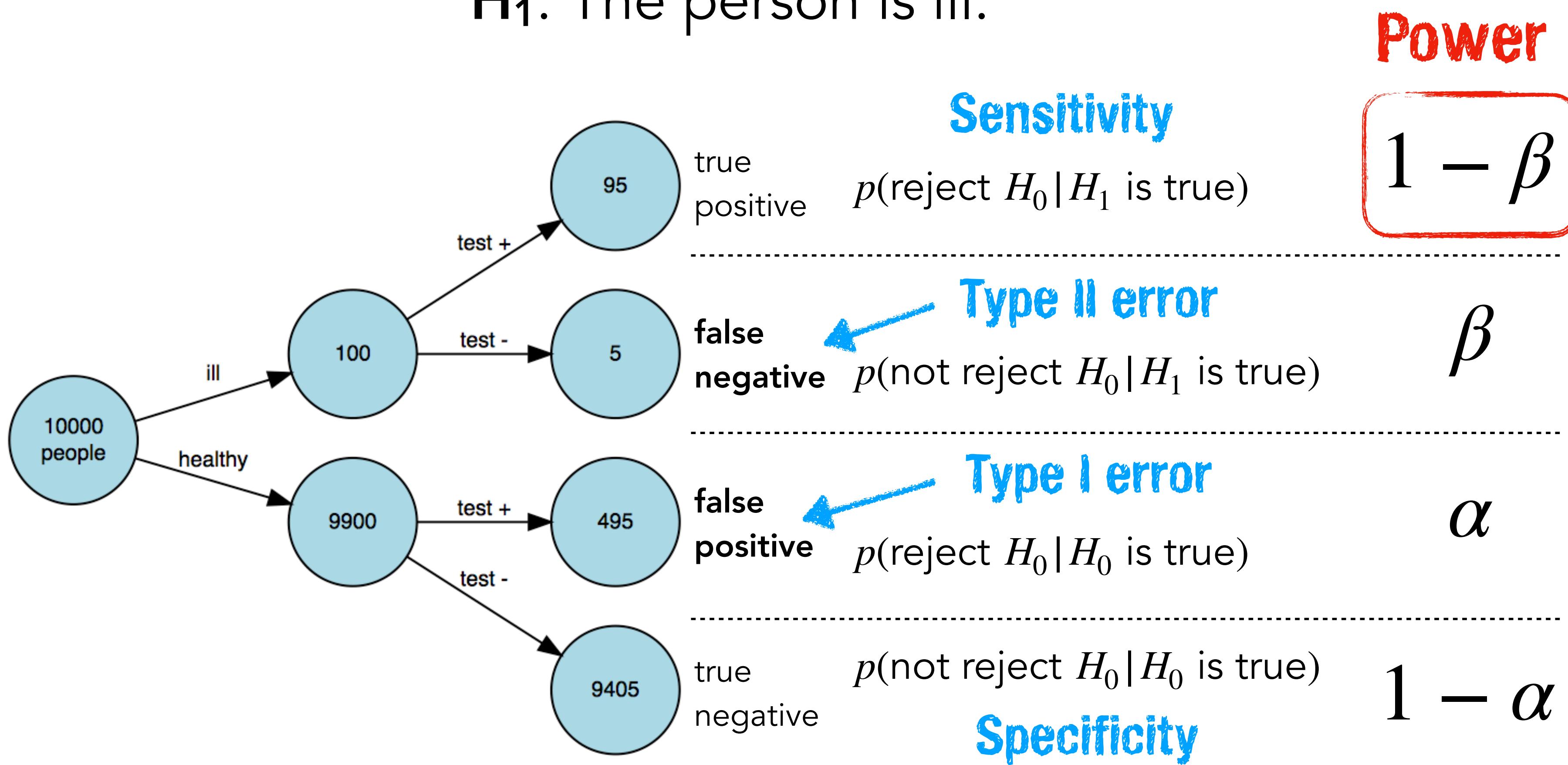
**Type I Error:** Falsely rejecting the null hypothesis (even though it is true).

**Type II Error:** Failing to reject the null hypothesis (even though it is false).

# Clue guide to probability

$H_0$ : The person is healthy.

$H_1$ : The person is ill.



# What affects power?

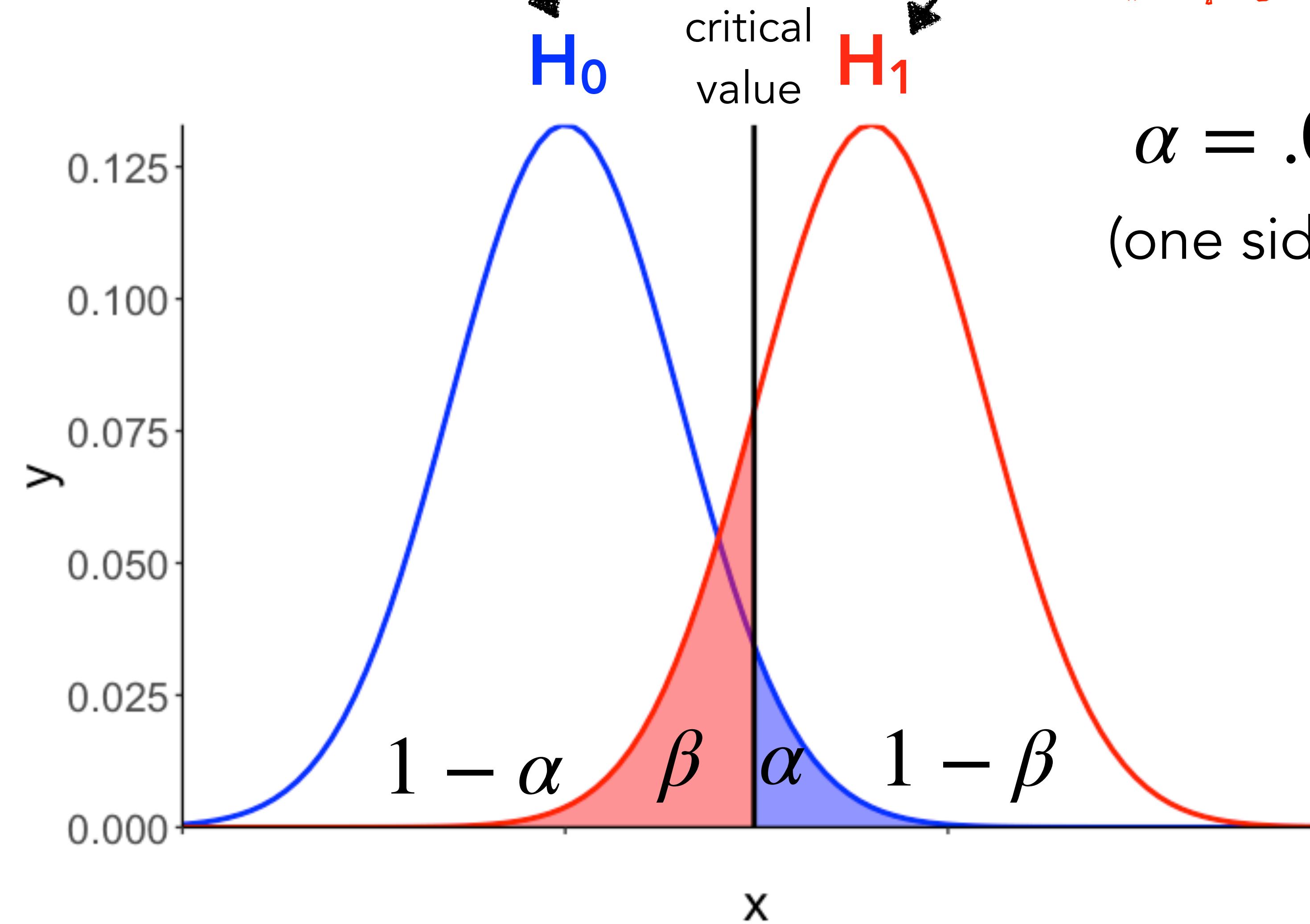
sampling distribution  
if  $H_0$  is true

$H_0$

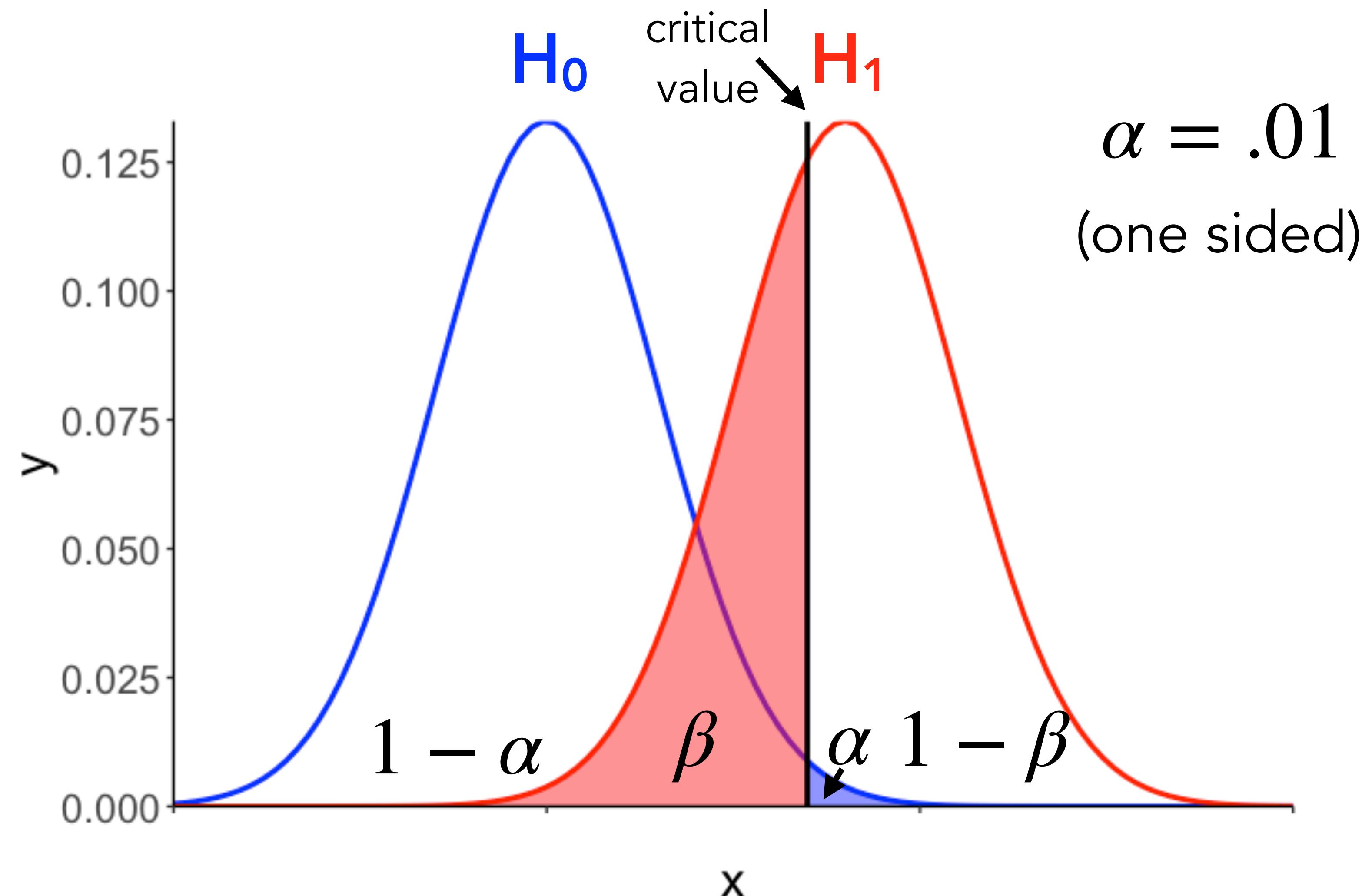
sampling distribution  
if  $H_1$  is true

$H_1$

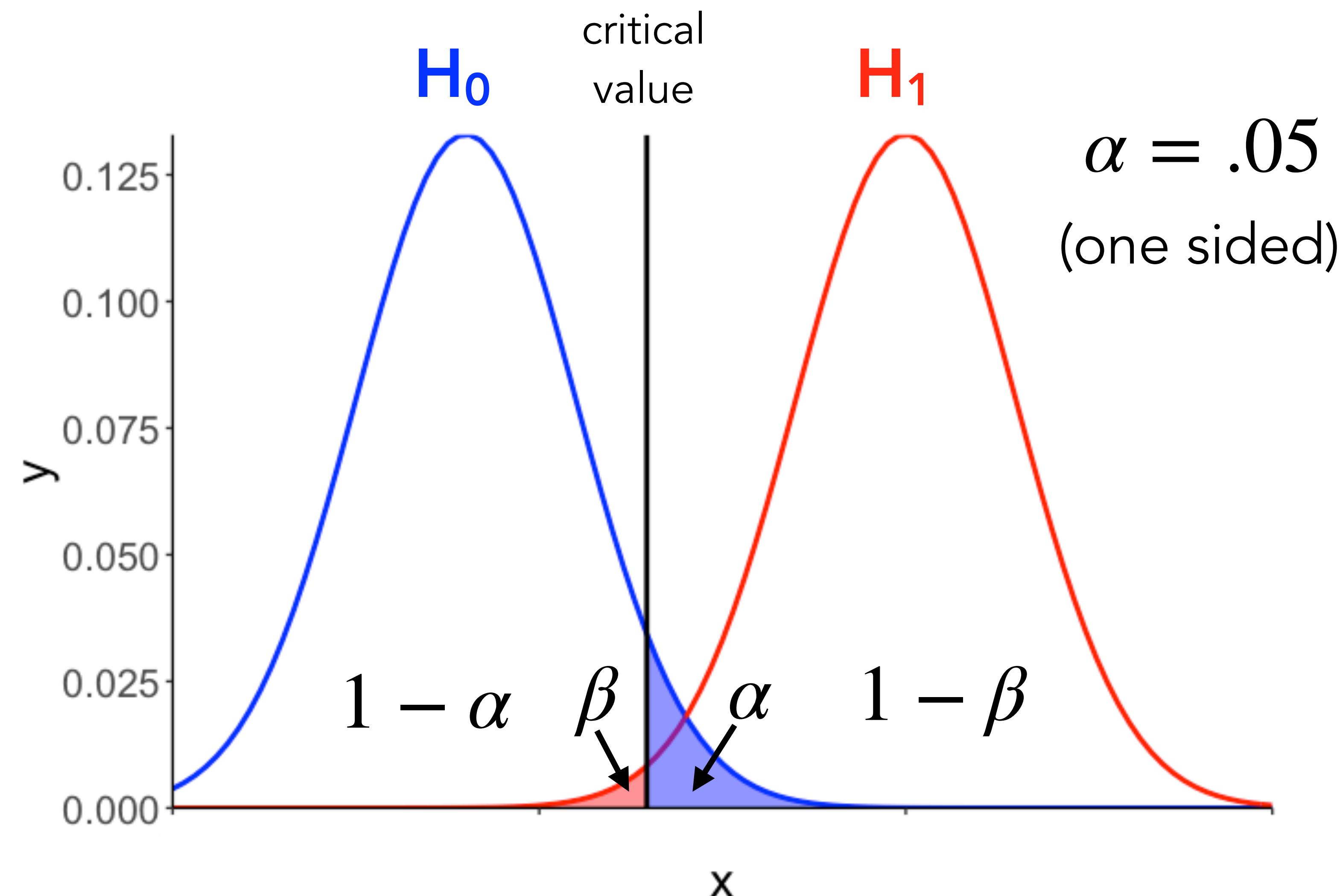
$\alpha = .05$   
(one sided)



# What affects power?

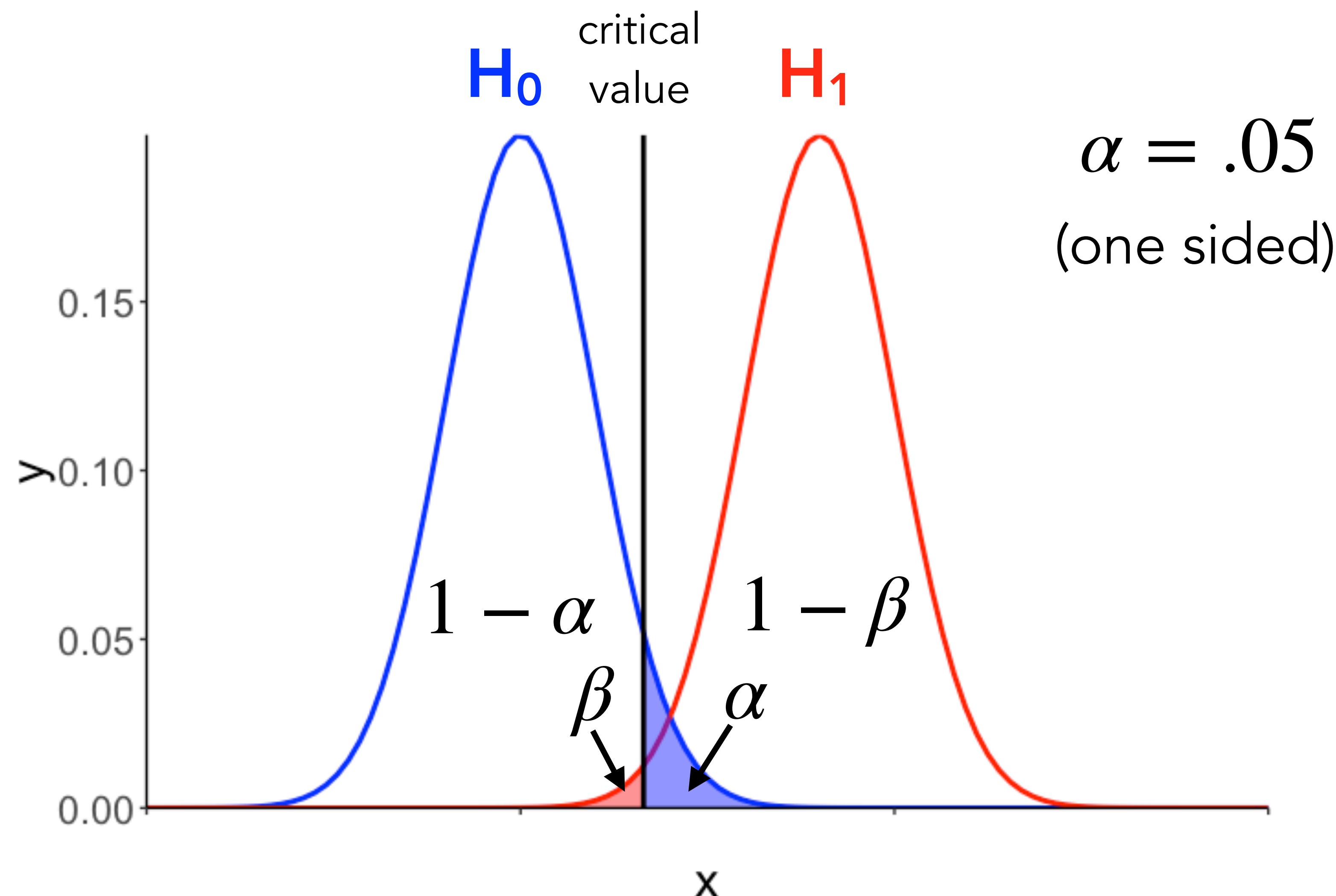


# What affects power?



distance between means affects power

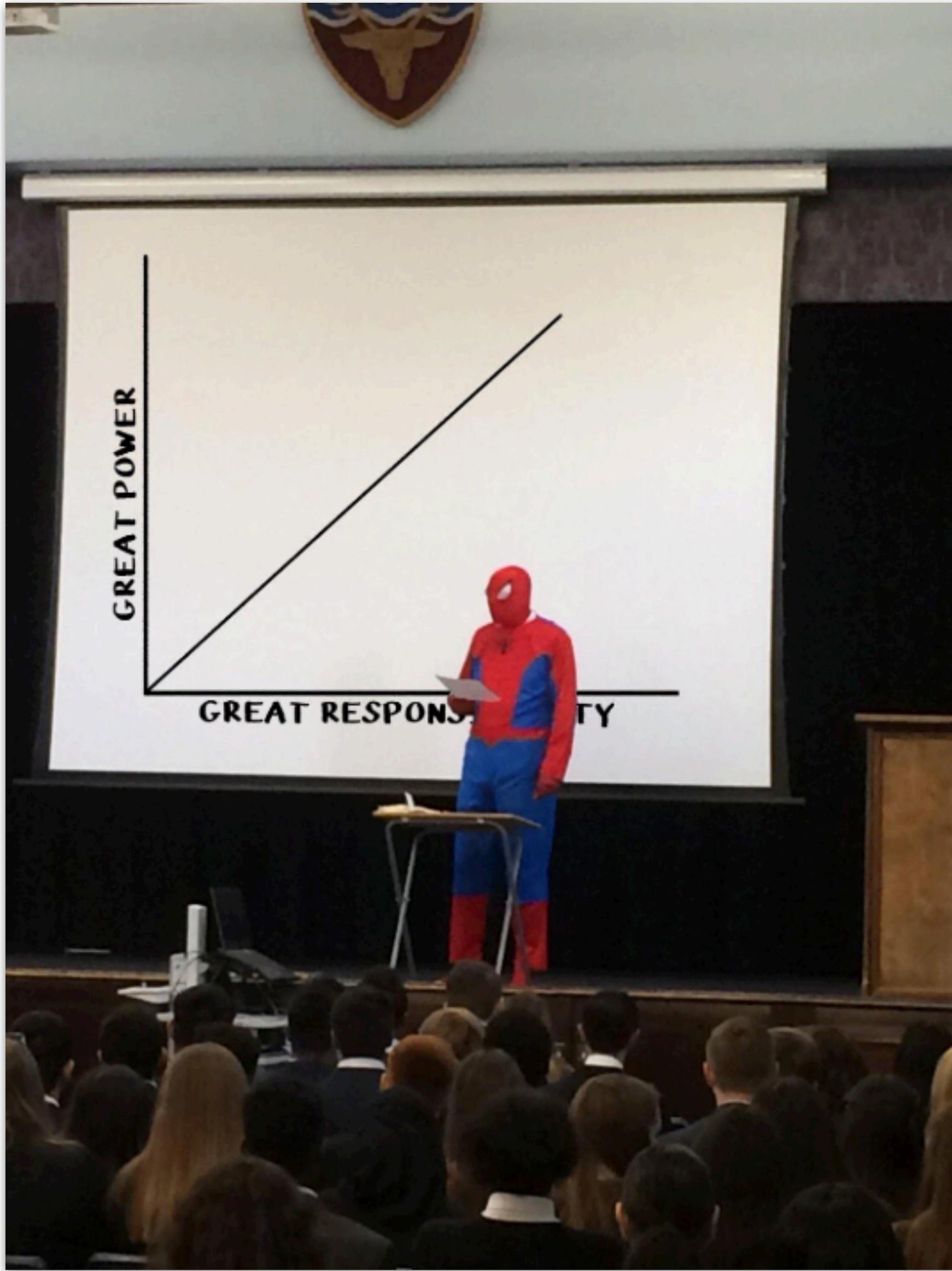
# What affects power?



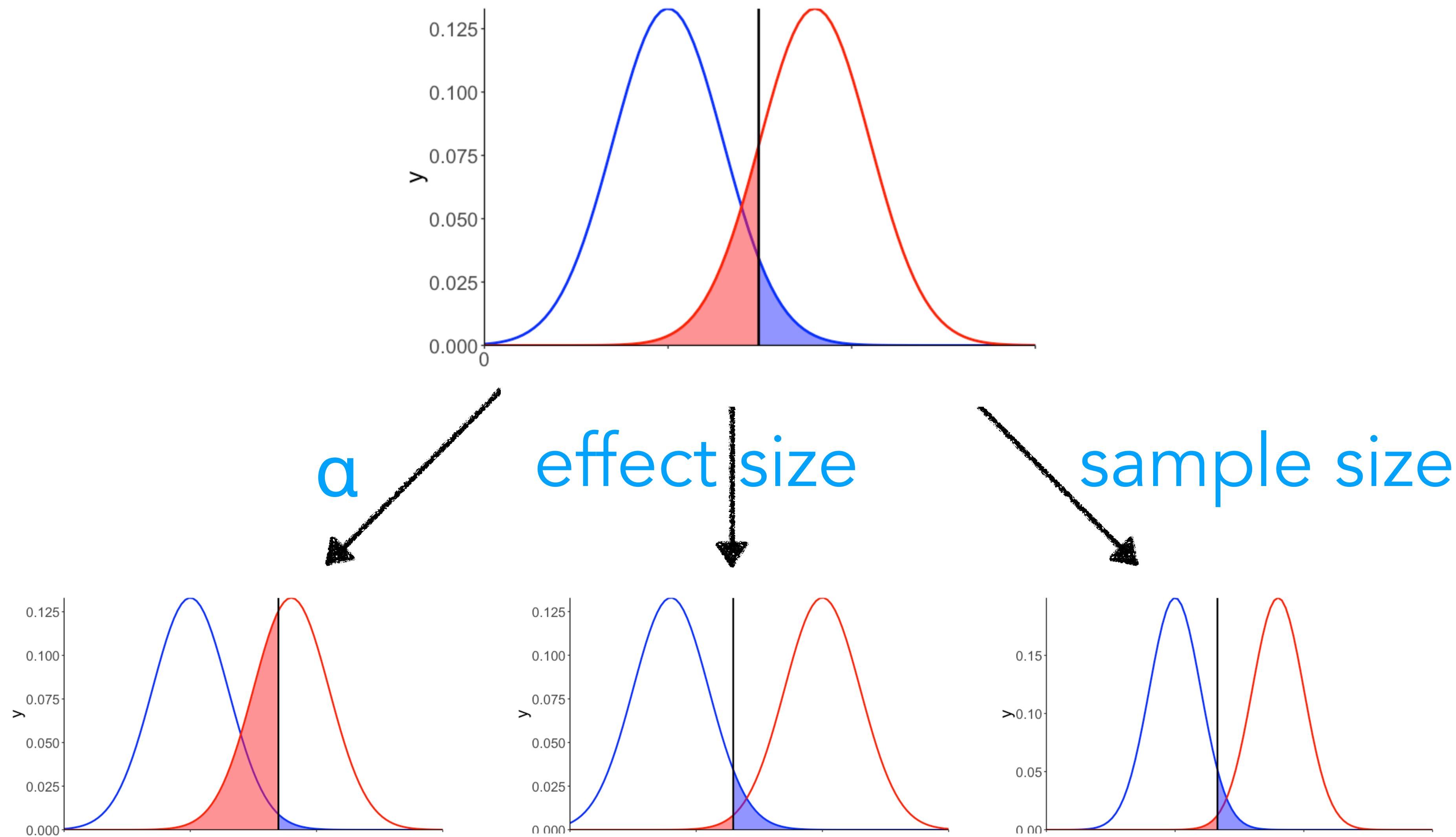
variance affects power

# Calculating power

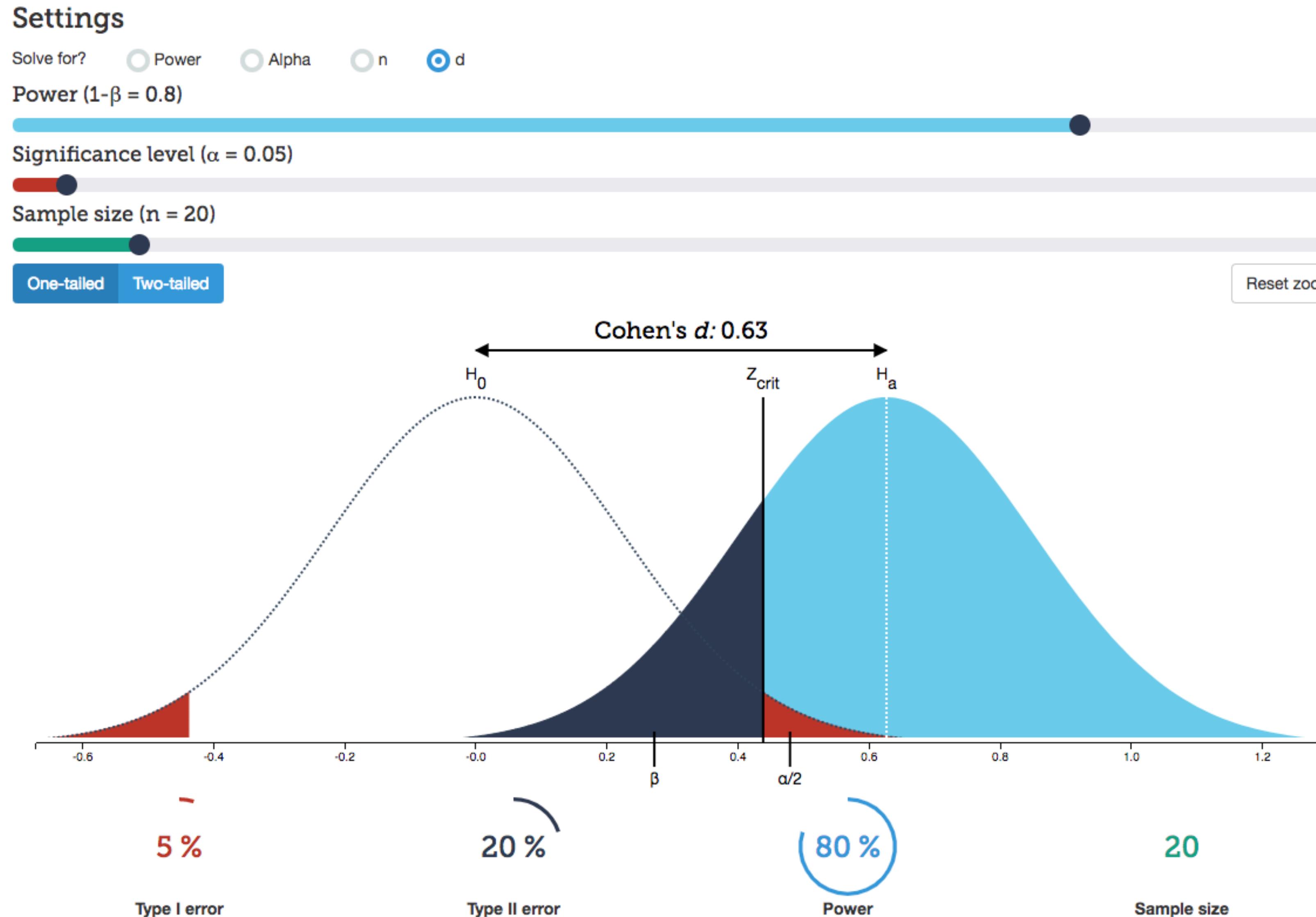
# With great power comes ...



# The knobs we can turn to affect power



# Visualization demo



The **power** of a binary hypothesis test is the probability that the test rejects the null hypothesis ( $H_0$ ) when a **specific** alternative hypothesis ( $H_1$ ) is true.

---

$H_0$ : Students and non-students have the same balance.

**Model C**

$$Y_i = \beta_0 + \epsilon_i$$

$$\beta_1 = 0$$

$H_1$ : Students and non-students have different balances.

**Model A**

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

$$\beta_1 \neq 0$$

We cannot calculate power in this case.  
We need a specific alternative hypothesis!

The **power** of a binary hypothesis test is the probability that the test rejects the null hypothesis ( $H_0$ ) when a **specific** alternative hypothesis ( $H_1$ ) is true.

---

$H_0$ : Students and non-students have the same balance.

**Model C**

$$Y_i = \beta_0 + \epsilon_i$$

$$\beta_1 = 0$$

$H_1$ : Students and non-students have different balances.

**Model A**

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

$$\beta_1 = 300$$

We can calculate power in this case (since we have a specific alternative hypothesis)!

# **Effect sizes**

# Effect sizes

- a p-value tells us whether we can reject the  $H_0$
- effect sizes is a measure of the strength of the actual effect

**Why can't we just use p-values  
as a measure of the effect size?**

$$F = \frac{\text{PRE}/(\text{PA} - \text{PC})}{(1 - \text{PRE})/(n - \text{PA})}$$

PRE = proportional reduction in error

PA = # parameters in the augmented model

PC = # parameters in the compact model

n = sample size

any PRE will become significant if n gets large enough

**statistical vs. practical significance**

# Effect sizes

**PRE** = proportional reduction in error

**Compact model**

SSE(C)

**Augmented model**

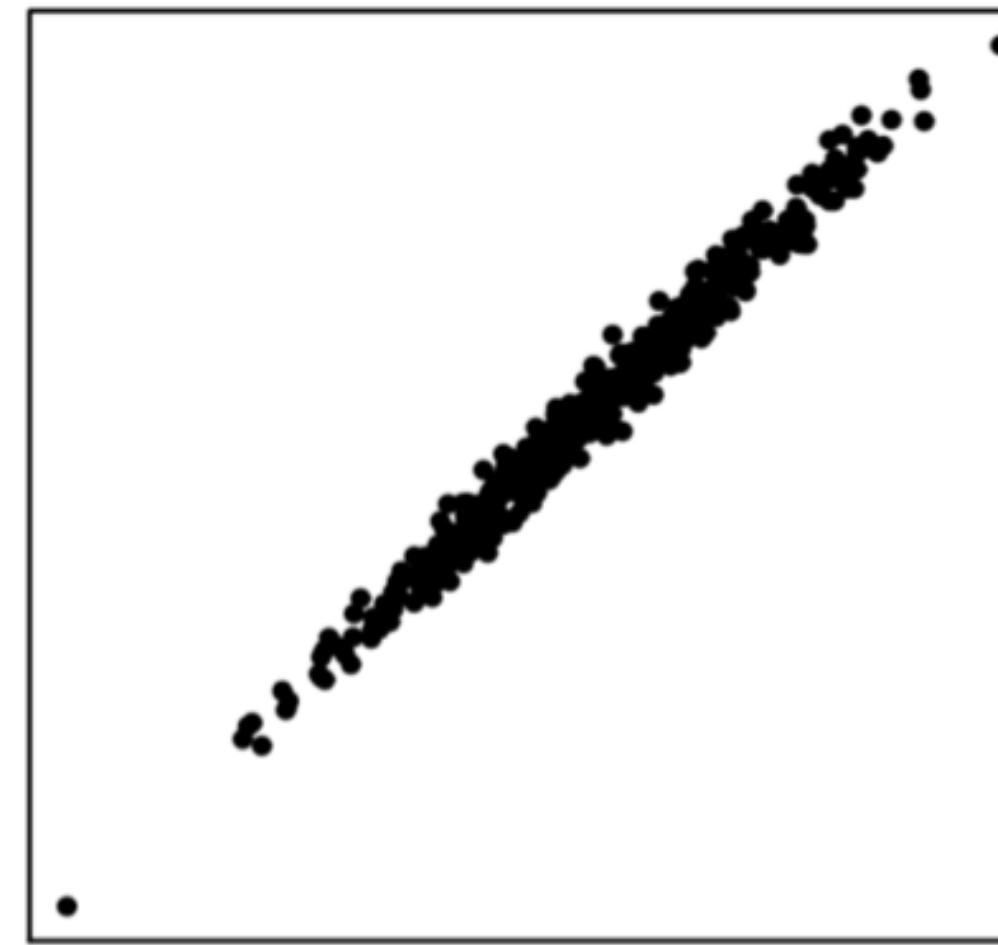
SSE(A)

$$\text{PRE} = 1 - \frac{\text{SSE}(A)}{\text{SSE}(C)}$$

SSE = sum of squared errors

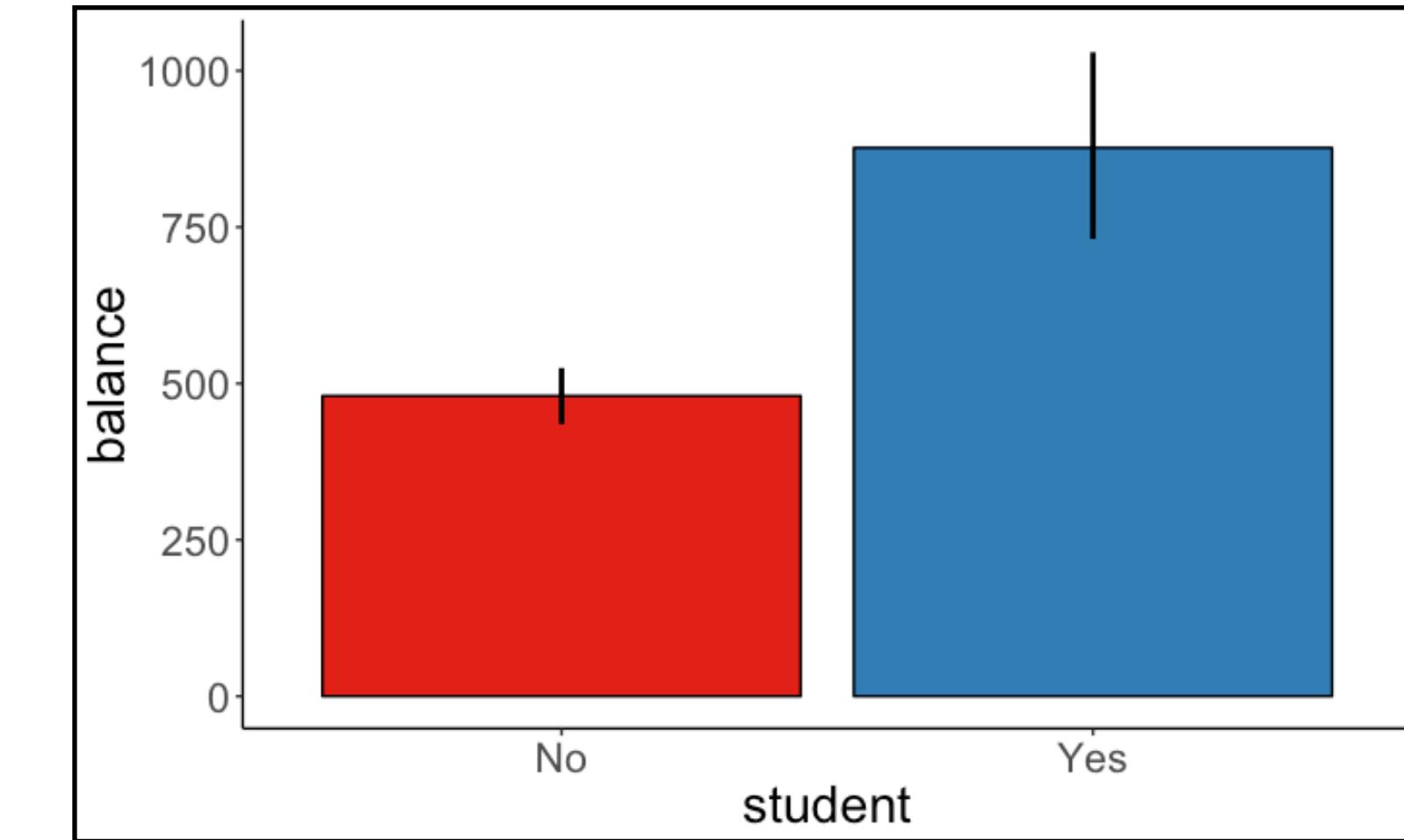
# Common effect sizes

## Relationships between variables



$r$  correlation

## Differences between groups



Cohen's  $d$

# Correlation

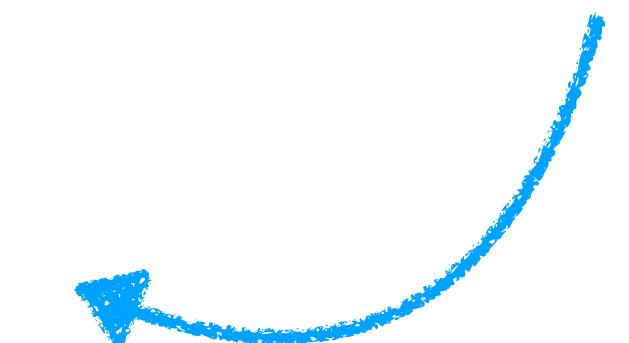
## Pearson correlation

$$r(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \cdot \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}$$

Cohen's guidelines for the social sciences

| Effect size | $r$ |
|-------------|-----|
| Small       | 0.1 |
| Medium      | 0.3 |
| Large       | 0.5 |

depends very  
much on the  
domain



# Cohen's $d$

- standardized difference between two means

absolute difference between means

$$d = \frac{|\bar{y}_1 - \bar{y}_2|}{s_p}$$

pooled standard variation

$$s_p = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}}$$

| <b>Effect size</b> | <b><math>d</math></b> |
|--------------------|-----------------------|
| Very small         | 0.01                  |
| Small              | 0.20                  |
| Medium             | 0.50                  |
| Large              | 0.80                  |
| Very large         | 1.20                  |
| Huge               | 2.0                   |

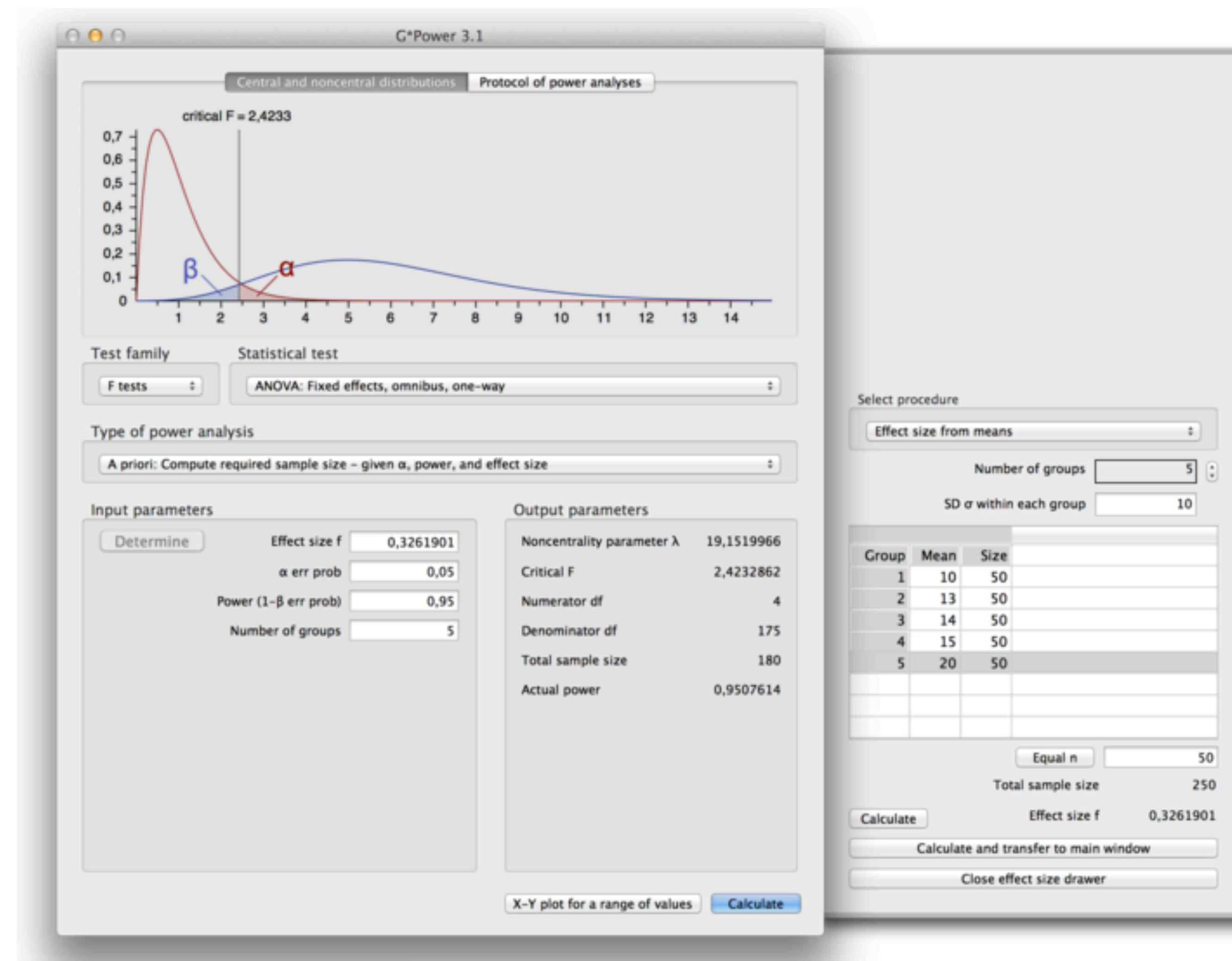
Difference between two means in pooled standard deviation

# Determining sample size

**How many participants do I need to run to have a  
good chance of detecting a true effect?**

# G\*Power 3.1: Alternative software for power calculations

Option 1

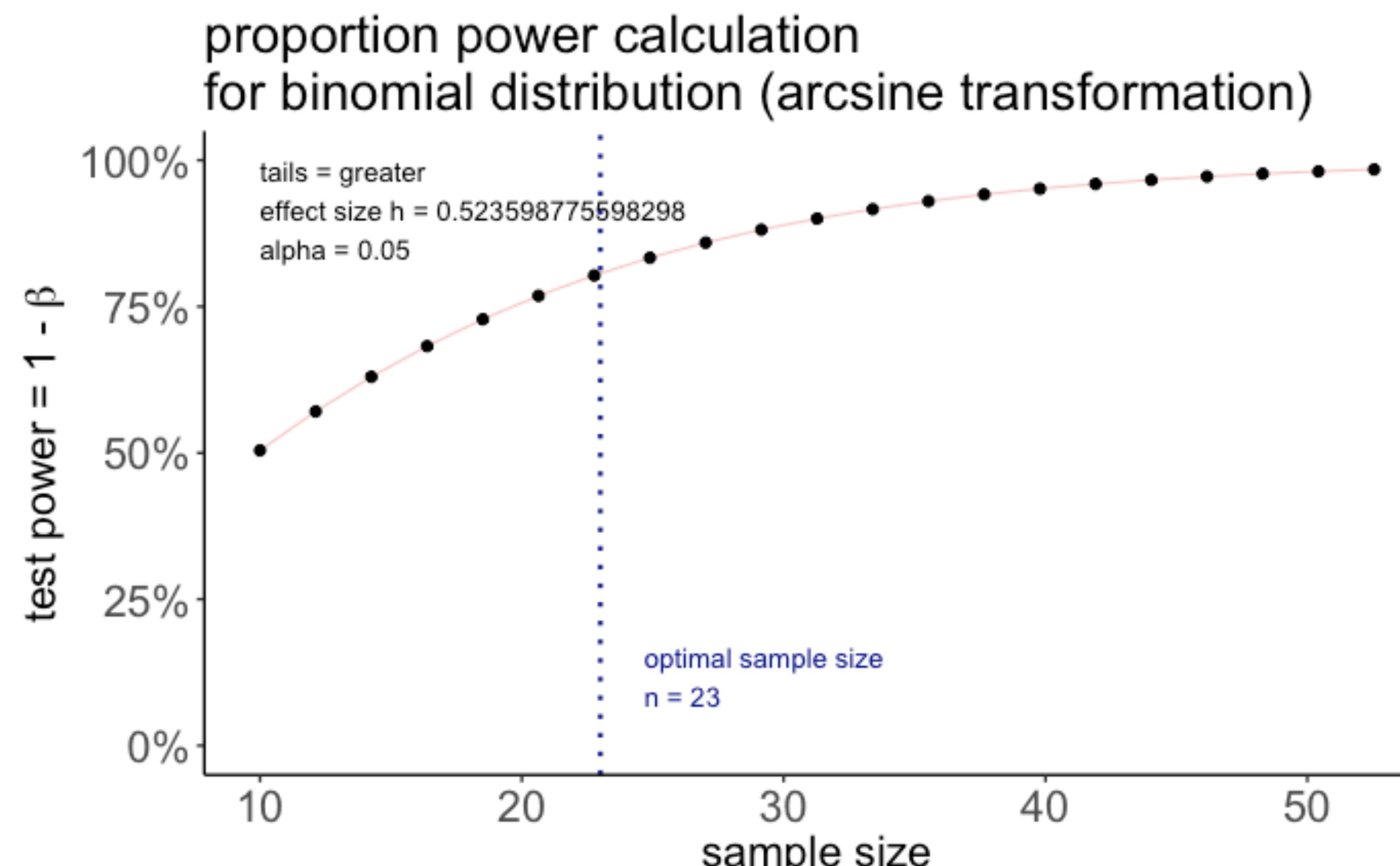


<http://www.gpower.hhu.de/>

# "pwr" package in R

Option 2

```
1 library("pwr")
2 pwr.p.test(h = ES.h(p1 = 0.75, p2 = 0.50),
3              sig.level = 0.05,
4              power = 0.80,
5              alternative = "greater") %>%
6 plot()
```



### Power simulation recipe

- assume:
  - $\alpha$ ,  $n$ , effect size
- simulate a large number of data sets of size  $n$  with the specified effect size
- for each data set, run a statistical test to calculate the p-value
- determine the probability of rejecting the  $H_0$  (given that  $H_1$  is true)

# **Simulating a power analysis**

## Power simulation recipe

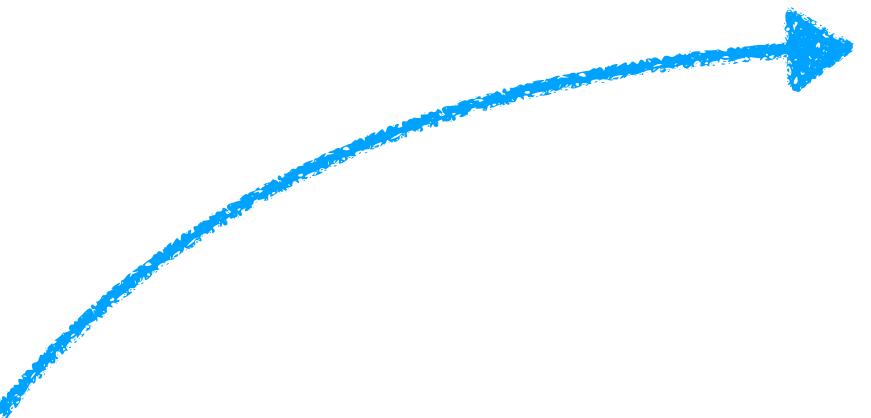
- assume:
  - $\alpha$ ,  $n$ , effect size
- simulate a large number of data sets of size  $n$  with the specified effect size
- for each data set, run a statistical test to calculate the p-value
- determine the probability of rejecting the  $H_0$  (given that  $H_1$  is true)

# Let's simulate ...

```
1 # make reproducible
2 set.seed(1)
3
4 # number of simulations
5 n_simulations = 5
6
7 # run simulation
8 expand_grid(n = seq(10, 40, 2),
9             simulation = 1:n_simulations,
10            p = 0.75) %>%
11 mutate(index = 1:n(),
12         .before = n) %>%
13 group_by(index, n, p, simulation) %>%
14 mutate(response = rbinom(n = 1,
15                         size = n,
16                         prob = p),
17         p.value = binom.test(x = response,
18                               n = n,
19                               p = 0.5,
20                               alternative = "two.sided")$p.value) %>%
21 group_by(n, p) %>%
22 summarize(power = sum(p.value < 0.05) / n())
```

# Let's simulate ...

```
1 # make reproducible
2 set.seed(1)
3
4 # number of simulations
5 n_simulations = 5
6
7 # run simulation
8 expand_grid(n = seq(10, 40, 2),
9             simulation = 1:n_simulations,
10            p = 0.75) %>%
11 mutate(index = 1:n(),
12         .before = n) %>%
```



| index | n  | simulation | p    |
|-------|----|------------|------|
| 1     | 10 | 1          | 0.75 |
| 2     | 10 | 2          | 0.75 |
| 3     | 10 | 3          | 0.75 |
| 4     | 10 | 4          | 0.75 |
| 5     | 10 | 5          | 0.75 |
| 6     | 12 | 1          | 0.75 |
| 7     | 12 | 2          | 0.75 |
| 8     | 12 | 3          | 0.75 |
| 9     | 12 | 4          | 0.75 |
| 10    | 12 | 5          | 0.75 |

# Let's simulate ...

```
1 # make reproducible
2 set.seed(1)
3
4 # number of simulations
5 n_simulations = 5
6
7 # run simulation
8 expand_grid(n = seq(10, 40, 2),
9             simulation = 1:n_simulations,
10            p = 0.75) %>%
11 mutate(index = 1:n(),
12         .before = n) %>%
13 group_by(index, n, p, simulation) %>%
14 mutate(response = rbinom(n = 1,
15                           size = n,
16                           prob = p),
```

| index | n  | simulation | p    | response |
|-------|----|------------|------|----------|
| 1     | 10 | 1          | 0.75 | 8        |
| 2     | 10 | 2          | 0.75 | 7        |
| 3     | 10 | 3          | 0.75 | 8        |
| 4     | 10 | 4          | 0.75 | 8        |
| 5     | 10 | 5          | 0.75 | 7        |
| 6     | 12 | 1          | 0.75 | 10       |
| 7     | 12 | 2          | 0.75 | 8        |
| 8     | 12 | 3          | 0.75 | 11       |
| 9     | 12 | 4          | 0.75 | 10       |
| 10    | 12 | 5          | 0.75 | 11       |



# Let's simulate ...

```
1 # make reproducible
2 set.seed(1)
3
4 # number of simulations
5 n_simulations = 5
6
7 # run simulation
8 expand_grid(n = seq(10, 40, 2),
9             simulation = 1:n_simulations,
10            p = 0.75) %>%
11 mutate(index = 1:n(),
12         .before = n) %>%
13 group_by(index, n, p, simulation) %>%
14 mutate(response = rbinom(n = 1,
15                         size = n,
16                         prob = p),
17         p.value = binom.test(x = response,
18                               n = n,
19                               p = 0.5,
20                               alternative = "two.sided")$p.value) %>%
```

| index | n  | simulation | p    | response | p.value |
|-------|----|------------|------|----------|---------|
| 1     | 10 | 1          | 0.75 | 8        | 0.11    |
| 2     | 10 | 2          | 0.75 | 7        | 0.34    |
| 3     | 10 | 3          | 0.75 | 8        | 0.11    |
| 4     | 10 | 4          | 0.75 | 8        | 0.11    |
| 5     | 10 | 5          | 0.75 | 7        | 0.34    |
| 6     | 12 | 1          | 0.75 | 10       | 0.04    |
| 7     | 12 | 2          | 0.75 | 8        | 0.39    |



# Let's simulate ...

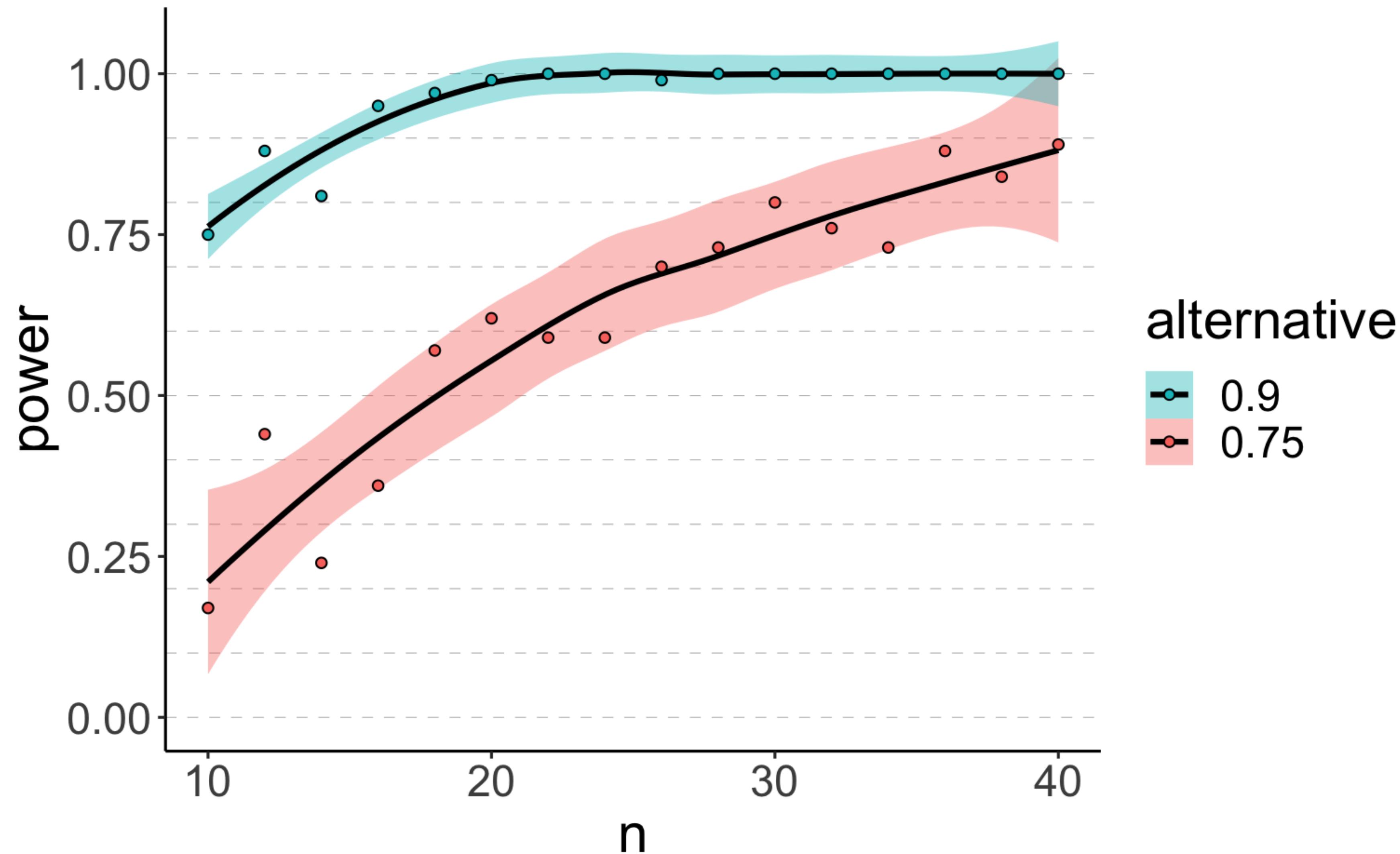
```
1 # make reproducible
2 set.seed(1)
3
4 # number of simulations
5 n_simulations = 5
6
7 # run simulation
8 expand_grid(n = seq(10, 40, 2),
9             simulation = 1:n_simulations,
10            p = 0.75) %>%
11 mutate(index = 1:n(),
12         .before = n) %>%
13 group_by(index, n, p, simulation) %>%
14 mutate(response = rbinom(n = 1,
15                           size = n,
16                           prob = p),
17         p.value = binom.test(x = response,
18                               n = n,
19                               p = 0.5,
20                               alternative = "two.sided")$p.value) %>%
21 group_by(n, p) %>%
22 summarize(power = sum(p.value < 0.05) / n())
```

| n  | p    | power |
|----|------|-------|
| 10 | 0.75 | 0.2   |
| 12 | 0.75 | 0.2   |
| 14 | 0.75 | 0.4   |
| 16 | 0.75 | 0.2   |
| 18 | 0.75 | 0.6   |
| 20 | 0.75 | 0.8   |
| 22 | 0.75 | 0.6   |
| 24 | 0.75 | 0.4   |
| 26 | 0.75 | 0.6   |
| 28 | 0.75 | 0.8   |
| 30 | 0.75 | 0.8   |



# Let's simulate ...

in this example, I looked at the power for two different alternative hypotheses



# Let's simulate ...

- here, I've used a simple example (Binomial test)
- but: we can use the same recipe for any statistical test that we are planning on running

## Power simulation recipe

- assume:
  - $\alpha$ ,  $n$ , effect size
- simulate a large number of data sets of size  $n$  with the specified effect size
- for each data set, run a statistical test to calculate the p-value for a given  $\alpha$
- determine the probability of rejecting the  $H_0$  (given that  $H_1$  is true)

# Let's simulate

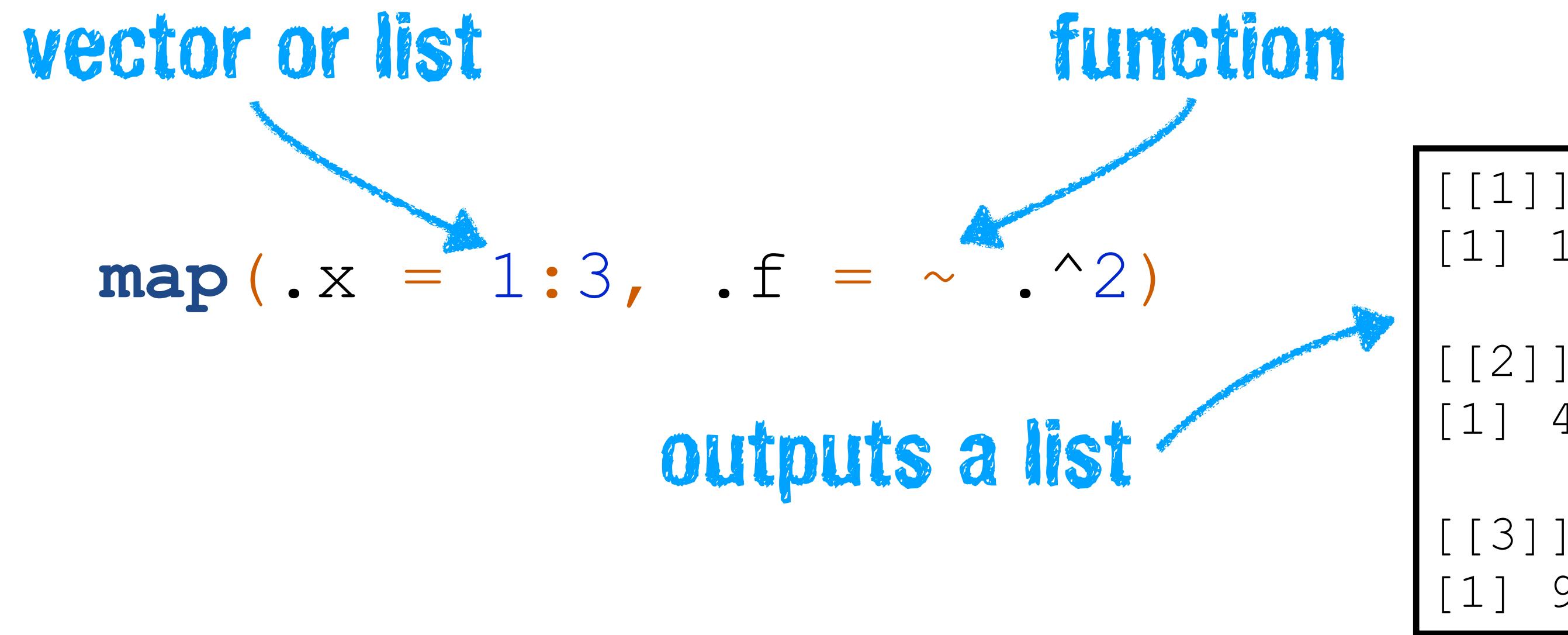
```
library("purrr")
```



automatically loaded with  
**library("tidyverse")**

**map ( )**

# **map ( )**



- **map**(list, function) applies a function to each element of the list
- it's a unified version of the many different **apply**() functions in base R
- you already know a cousin of **map**(): **replicate**()
- use **map**(), don't write `for () {}` loops!
- it's extremely powerful in combination with data frames

# map ()

same same but different

```
map (.x = 1:3, .f = ~ .x^2)
```

```
map (1:3, ~ .x^2)
```

```
map (1:3, ~ .^2)
```

```
map (.x = 1:3, .f = function (.x) .x^2)
```

```
# using a function
```

```
square = function (x) {x^2}  
map (1:3, square)
```

INTERACTIVE COURSE

# Foundations of Functional Programming with purrr

[Start Course For Free](#)

[Bookmark](#)

⌚ 4 hours ⏷ 13 Videos ↗ 44 Exercises 📃 7,769 Participants 🏆 3,750 XP

## Course Description

Lists can be difficult to both understand and manipulate, but they can pack a ton of information and are very powerful. In this course, you will learn to easily extract, summarize, and manipulate lists and how to export the data to your desired object, be it another list, a vector, or even something else! Throughout the course, you will work with the `purrr` package and a variety of datasets from the `reprex` package, including data from Star Wars and Wes Anderson films and data collected about GitHub users and GitHub repos. Following this course, your list skills will be purrrfect!

### 1 Simplifying Iteration and Lists With purrr FREE

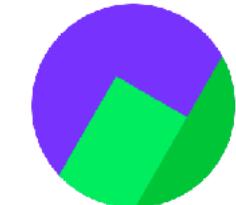
0%

Iteration is a powerful way to make the computer do the work for you. It can also be an area of coding where it is easy to make lots of typos and simple mistakes. The `purrr` package helps simplify iteration so you can focus on the next step, instead of finding typos.

- |   |        |
|---|--------|
| ▶ The power of iteration                | 50 xp  |
| ◀/▶ Introduction to iteration           | 100 xp |
| ◀/▶ Iteration with purrr                | 100 xp |
| ◀/▶ More iteration with for loops       | 100 xp |
| ◀/▶ More iteration with purrr           | 100 xp |
| ▶ Subsetting lists; it's not that hard! | 50 xp  |
| ◀/▶ Subsetting lists                    | 100 xp |
| ◀/▶ Subsetting list elements            | 100 xp |

This course is part of these tracks:

Intermediate Tidyverse Toolbox



DataCamp Content Creator

Course Instructor

This instructor prefers to remain anonymous.

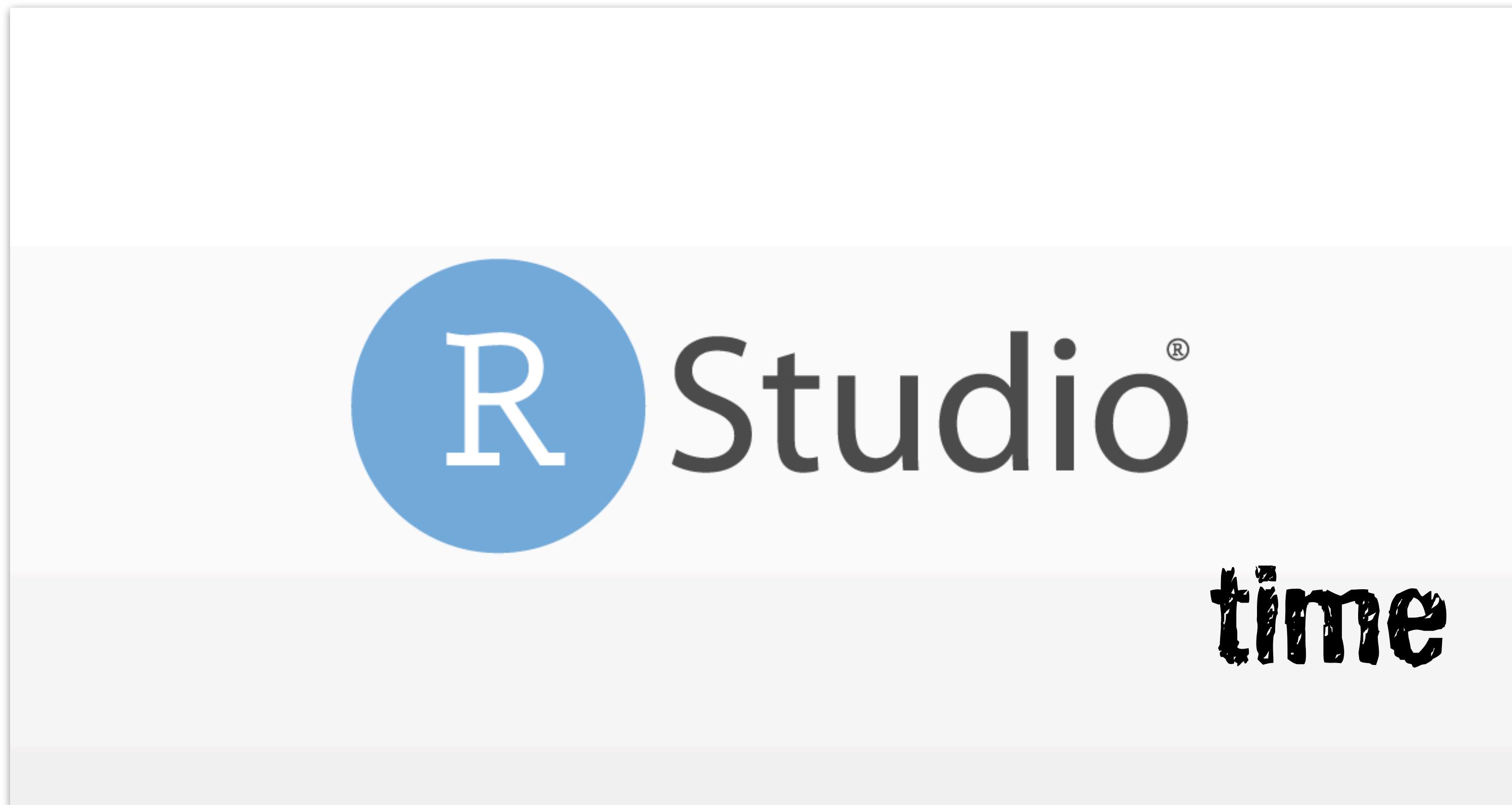
[See More](#)

## COLLABORATOR(S)

- |  |               |
|--|---------------|
|  | Chester Ismay |
|  | Becca Robins  |

<https://www.datacamp.com/courses-foundations-of-functional-programming-with-purrr>

# Simulating a power analysis



# Plan for today

- Quick recap
- Power analysis
  - Making decisions
  - Calculating power
  - Effect sizes
  - Determining sample size
- Simulating a power analysis
  - Demonstration in RStudio
  -



0%

much too slow

0%

a little too slow

0%

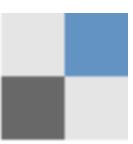
just right

0%

a little too fast

0%

much too fast



Start the presentation to see live content. For screen share software, share the entire screen. Get help at [pollev.com/app](https://pollev.com/app)



Thank you!