infer

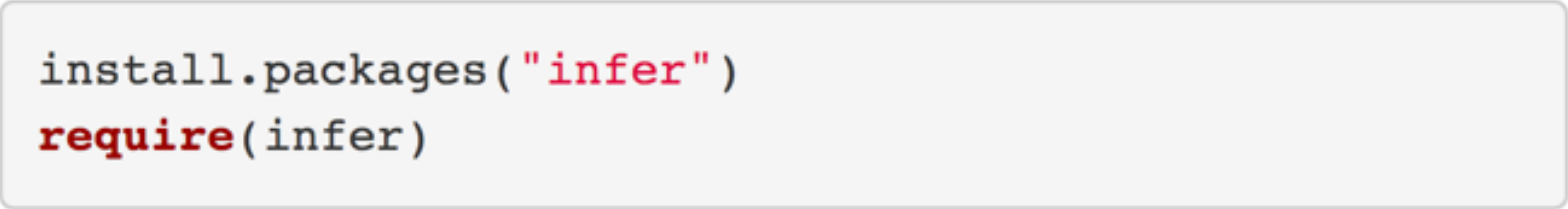
an R package for tidy statistical inference

Andrew Bray

infer.netlify.com

How to kick the tires on infer

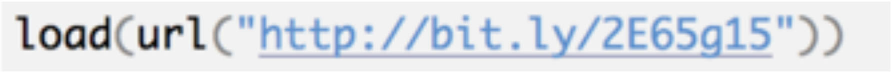
- If you have R installed on your computer, you can download and install the infer package:



```
install.packages("infer")
require(infer)
```

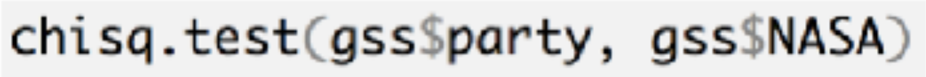
- The package website provides documentation and example vignettes: infer.netlify.com

- GSS data available with

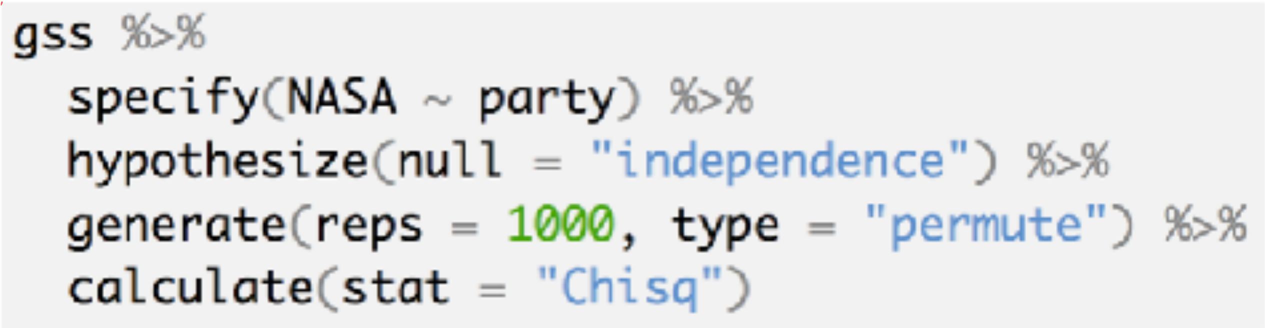


```
load(url("http://bit.ly/2E65g15"))
```

The goal of this presentation



```r
chisq.test(gss$party, gss$NASA)
```



```r
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq")
```

Competing goals in Intro

- Instill principles of statistics
- Train effective tool users
- Empower students to answer statistical questions

4-way Tug of War

Less Volume, More Creativity

Aim for an R toolkit that is

- **small**: fewer commands/templates is better
- **coherent**: commands should be as similar as possible
- **powerful**: can do what needs doing

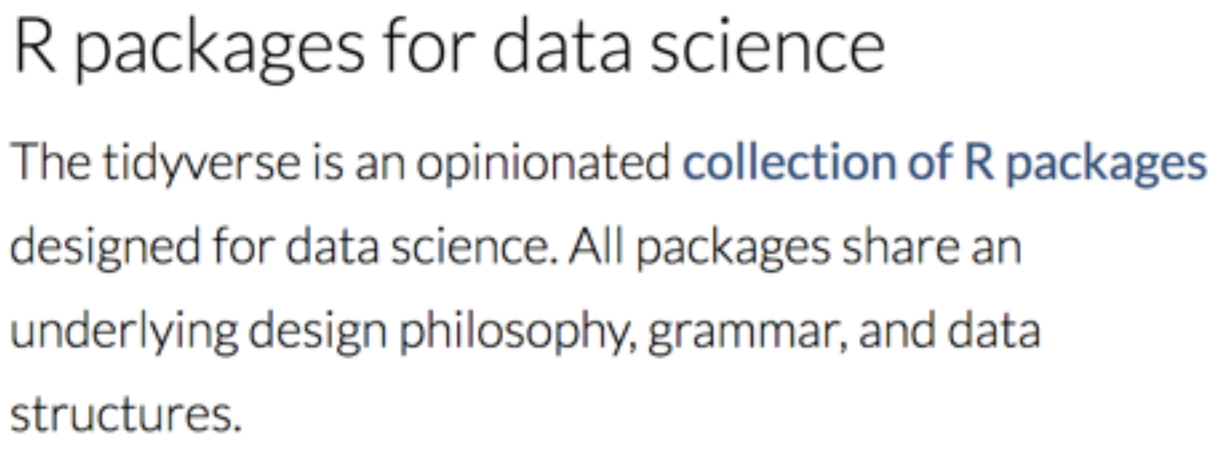
> Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away.
>
> — Antoine de Saint-Exupery (writer, poet, pioneering aviator)

infer

Inspired by the *Less Volume, More Creativity* philosophy, an R package for statistical inference that

- Conforms to the Tidy Tools Manifesto
- Unifies computation and approximation

Tidyverse



R packages for data science

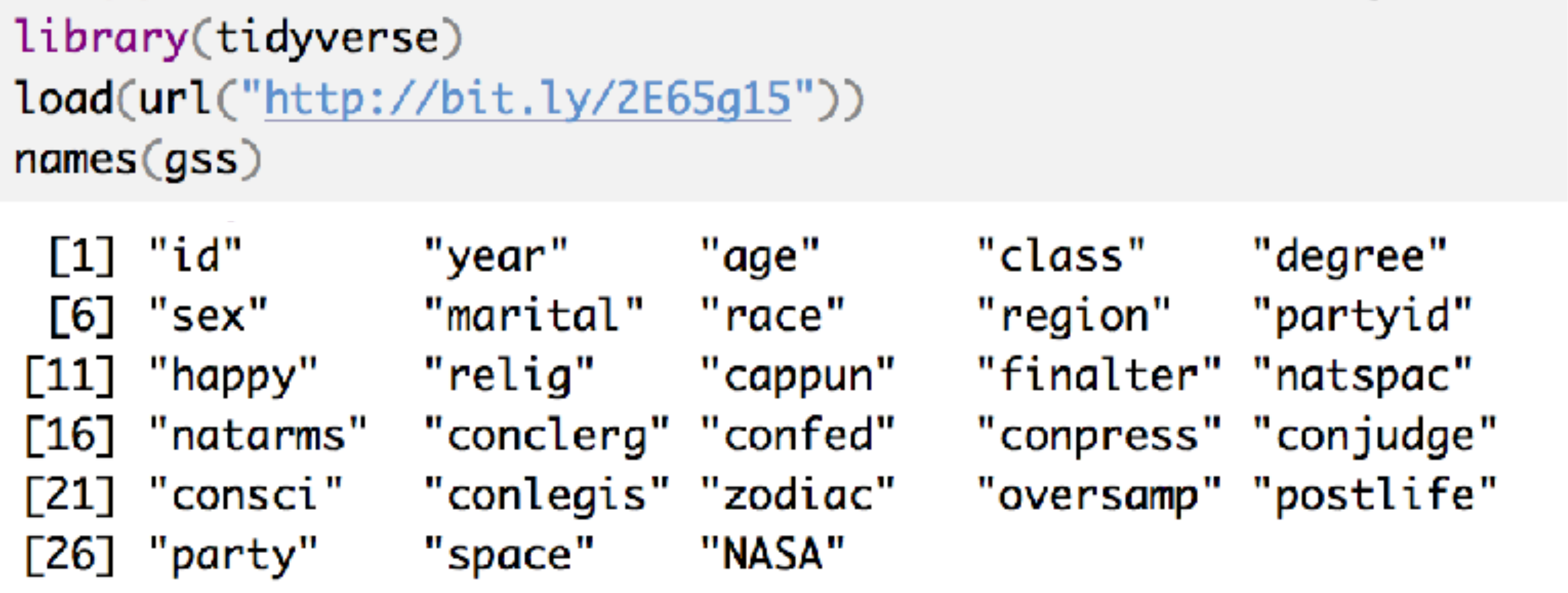
The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Design: compose functions

Grammar: write for humans

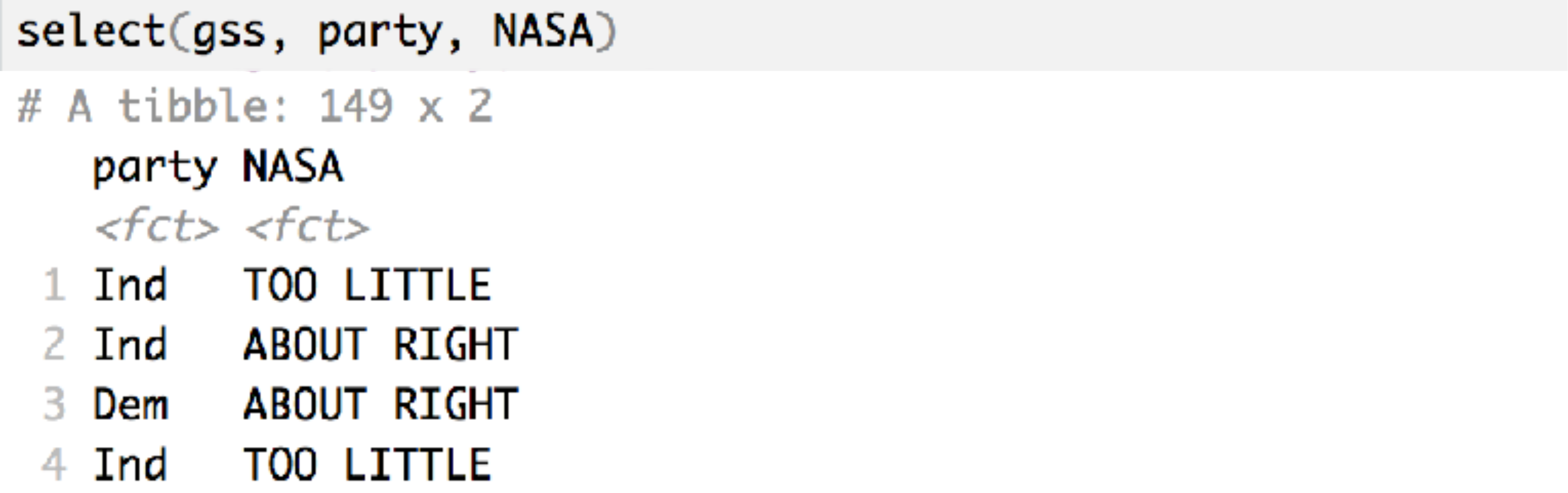
Data Structures: dataframes

Case study: Is funding for space exploration a partisan issue?



```
library(tidyverse)
load(url("http://bit.ly/2E65g15"))
names(gss)
```

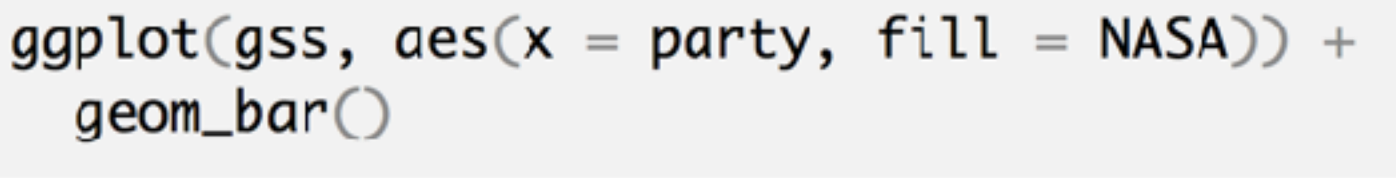
```
 [1] "id"       "year"     "age"      "class"    "degree"  
 [6] "sex"      "marital"  "race"     "region"   "partyid" 
[11] "happy"    "relig"    "cappun"   "finalter" "natspac" 
[16] "natarms"  "conclerg" "confed"   "conpress" "conjudge"
[21] "consci"   "conlegis" "zodiac"   "oversamp" "postlife"
[26] "party"    "space"    "NASA"    
```



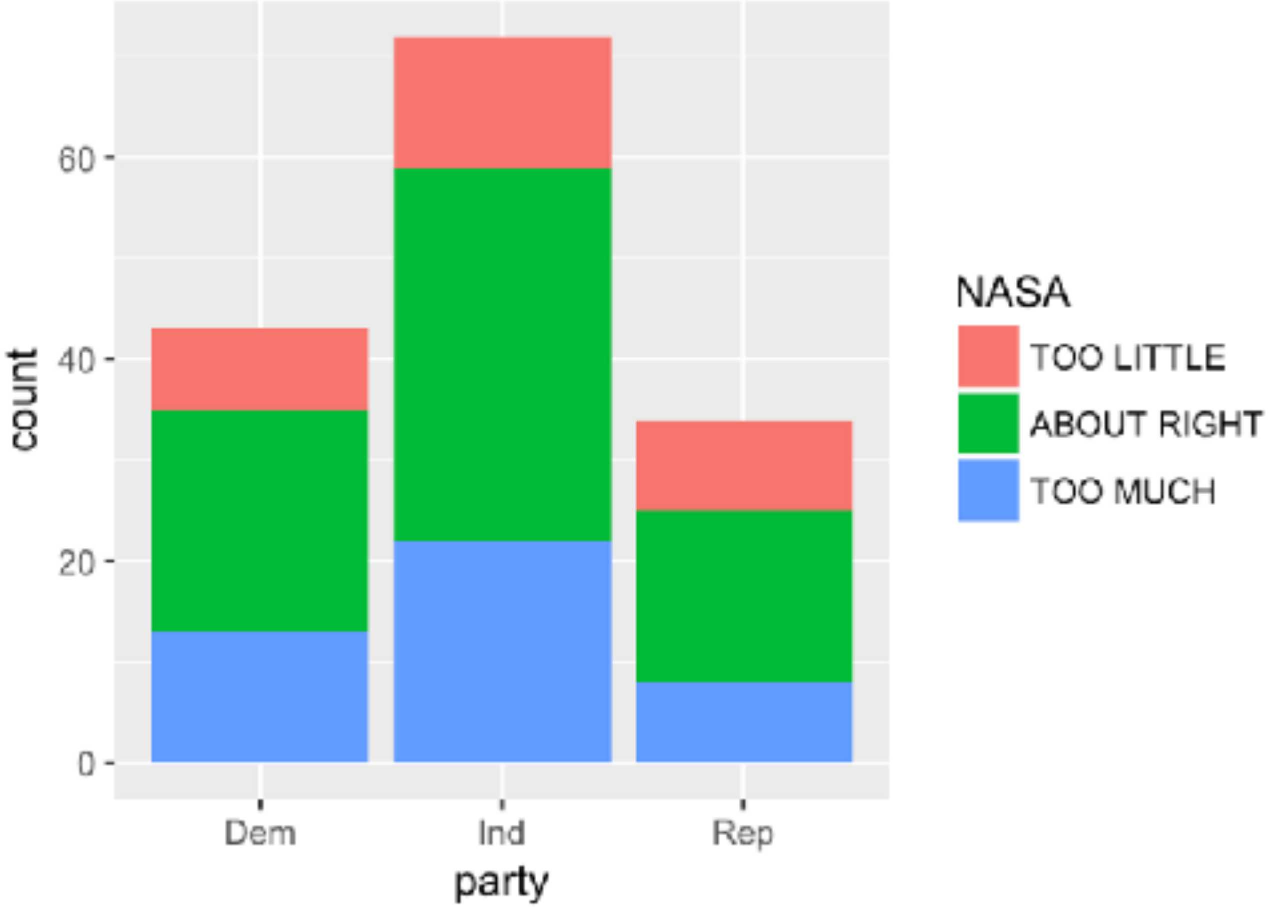
```
select(gss, party, NASA)
```

```
# A tibble: 149 x 2
  party NASA       
  <fct> <fct>      
1 Ind   TOO LITTLE 
2 Ind   ABOUT RIGHT
3 Dem   ABOUT RIGHT
4 Ind   TOO LITTLE 
```

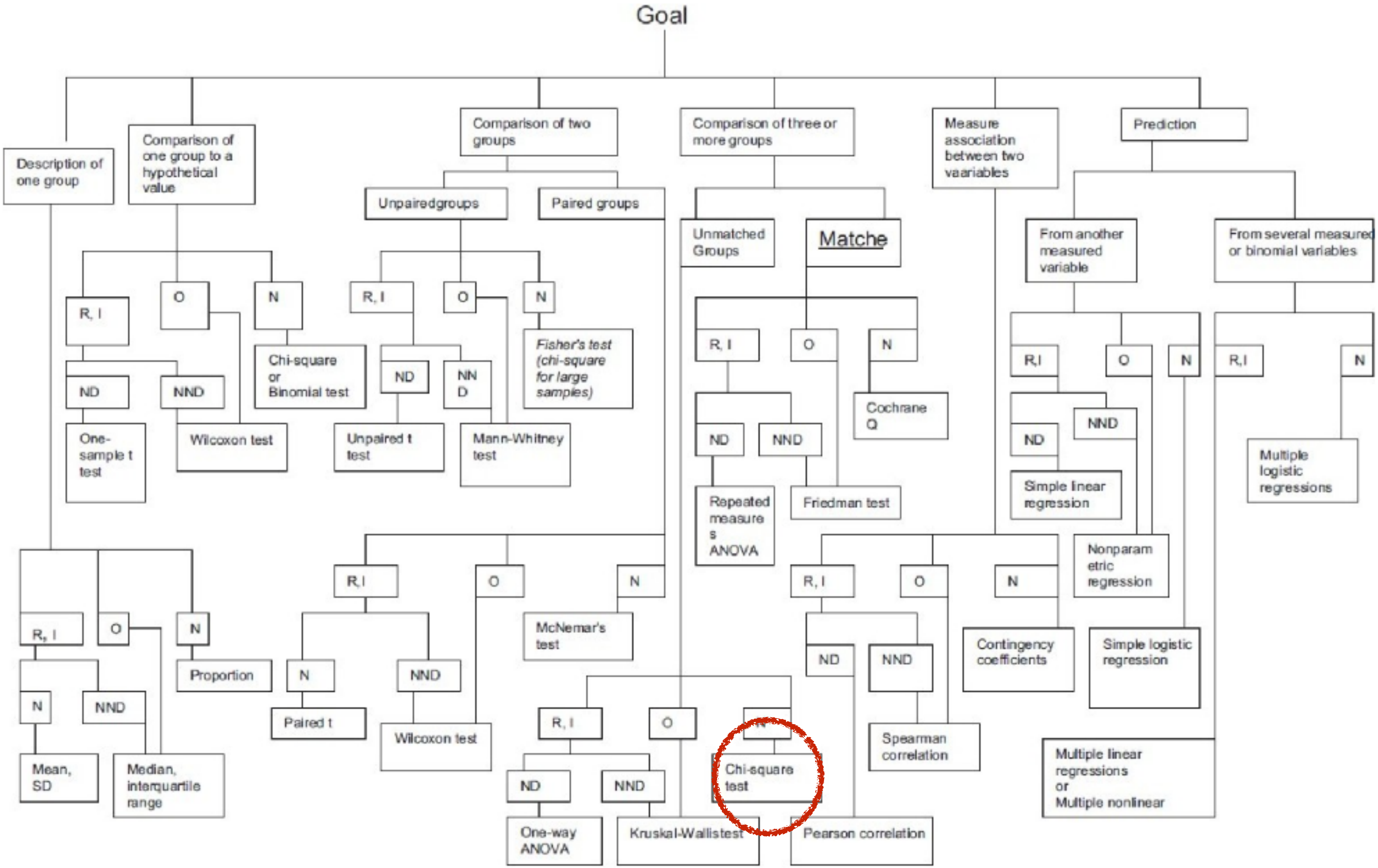
Case study: Is funding for space exploration a partisan issue?



```r
ggplot(gss, aes(x = party, fill = NASA)) +
  geom_bar()
```



Test to see if the structure that we see is *significant.*



Goal

Description of one group

Comparison of one group to a hypothetical value

Comparison of two groups

Comparison of three or more groups

Measure association between two vaariables

Prediction

Unpairedgroups

Paired groups

Unmatched Groups

Matche

From another measured variable

From several measured or binomial variables

R, I

O

N

Chi-square or Binomial test

ND

NND

One-sample t test

Wilcoxon test

R, I

O

N

ND

NN D

Fisher's test (chi-square for large samples)

Unpaired t test

Mann-Whitney test

R, I

O

N

ND

NND

Cochrane Q

Repeated measures ANOVA

Friedman test

R,I

O

N

NND

ND

Simple linear regression

Nonparametric regression

R,I

N

Multiple logistic regressions

R, I

O

N

ND

NND

Mean, SD

Median, interquartile range

Proportion

R,I

O

N

N

NND

Paired t

Wilcoxon test

McNemar's test

R, I

O

N

ND

NND

One-way ANOVA

Kruskal-Wallistest

Chi-square test

R, I

O

N

ND

NND

Spearman correlation

Pearson correlation

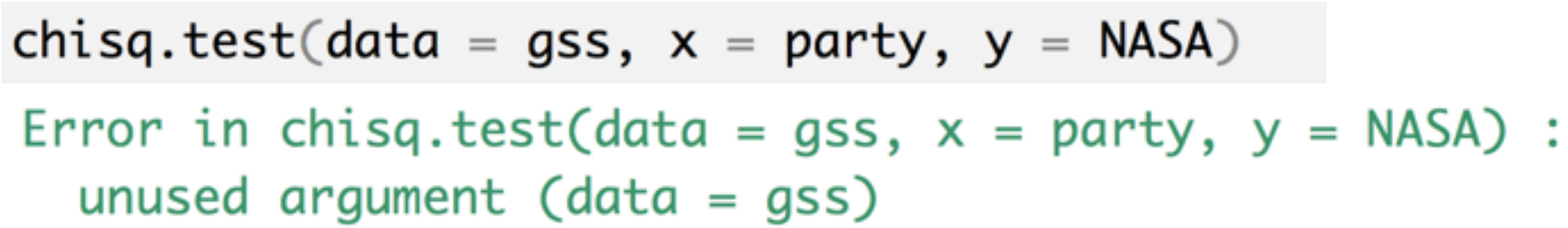
Contingency coefficients

Simple logistic regression

Multiple linear regressions or Multiple nonlinear

Journal of Pharmacological Negative Results, DOI: 10.4103/0976-9234.75708

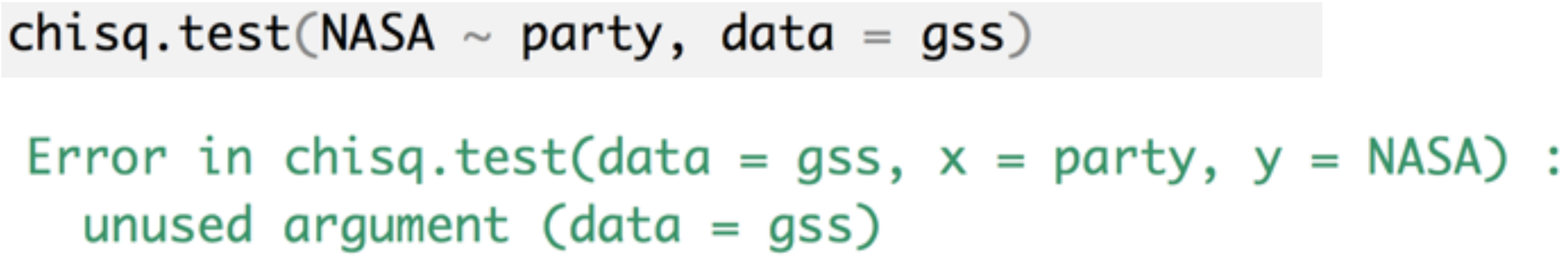
Optimistic effort I



```
chisq.test(data = gss, x = party, y = NASA)
```

```
Error in chisq.test(data = gss, x = party, y = NASA) :
  unused argument (data = gss)
```

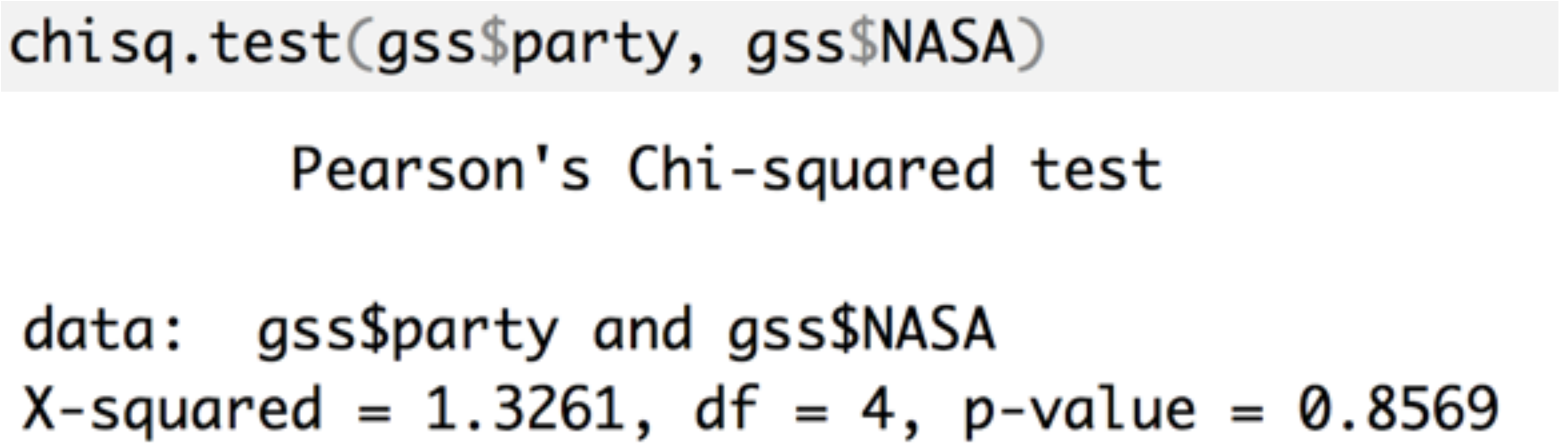
. . . optimistic effort II



```
chisq.test(NASA ~ party, data = gss)
```

```
Error in chisq.test(data = gss, x = party, y = NASA) :
  unused argument (data = gss)
```

...after looking at the help file

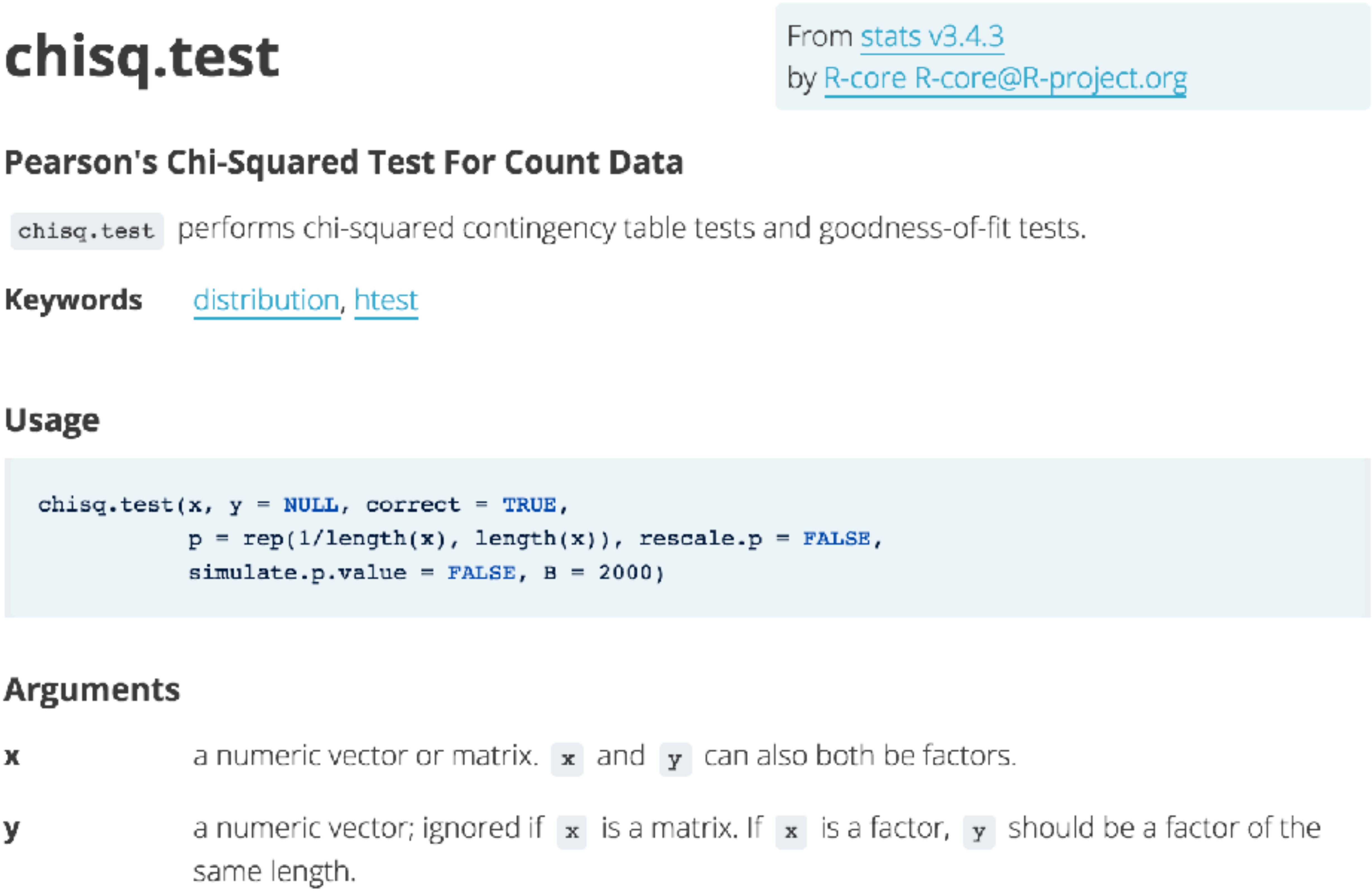


```
chisq.test(gss$party, gss$NASA)
```

```

	Pearson's Chi-squared test

data:  gss$party and gss$NASA
X-squared = 1.3261, df = 4, p-value = 0.8569
```



chisq.test

From stats v3.4.3
by R-core R-core@R-project.org

Pearson's Chi-Squared Test For Count Data

`chisq.test` performs chi-squared contingency table tests and goodness-of-fit tests.

Keywords distribution, htest

Usage

```
chisq.test(x, y = NULL, correct = TRUE,
           p = rep(1/length(x), length(x)), rescale.p = FALSE,
           simulate.p.value = FALSE, B = 2000)
```

Arguments

x a numeric vector or matrix. `x` and `y` can also both be factors.

y a numeric vector; ignored if `x` is a matrix. If `x` is a factor, `y` should be a factor of the same length.

infer

Inspired by the *Less Volume, More Creativity* philosophy, an R package for statistical inference that

- Conforms to the Tidy Tools Manifesto
- Unifies computation and approximation

Two Paradigms

Mathematical Approximation

- Chi-squared
- Student t
- Normal

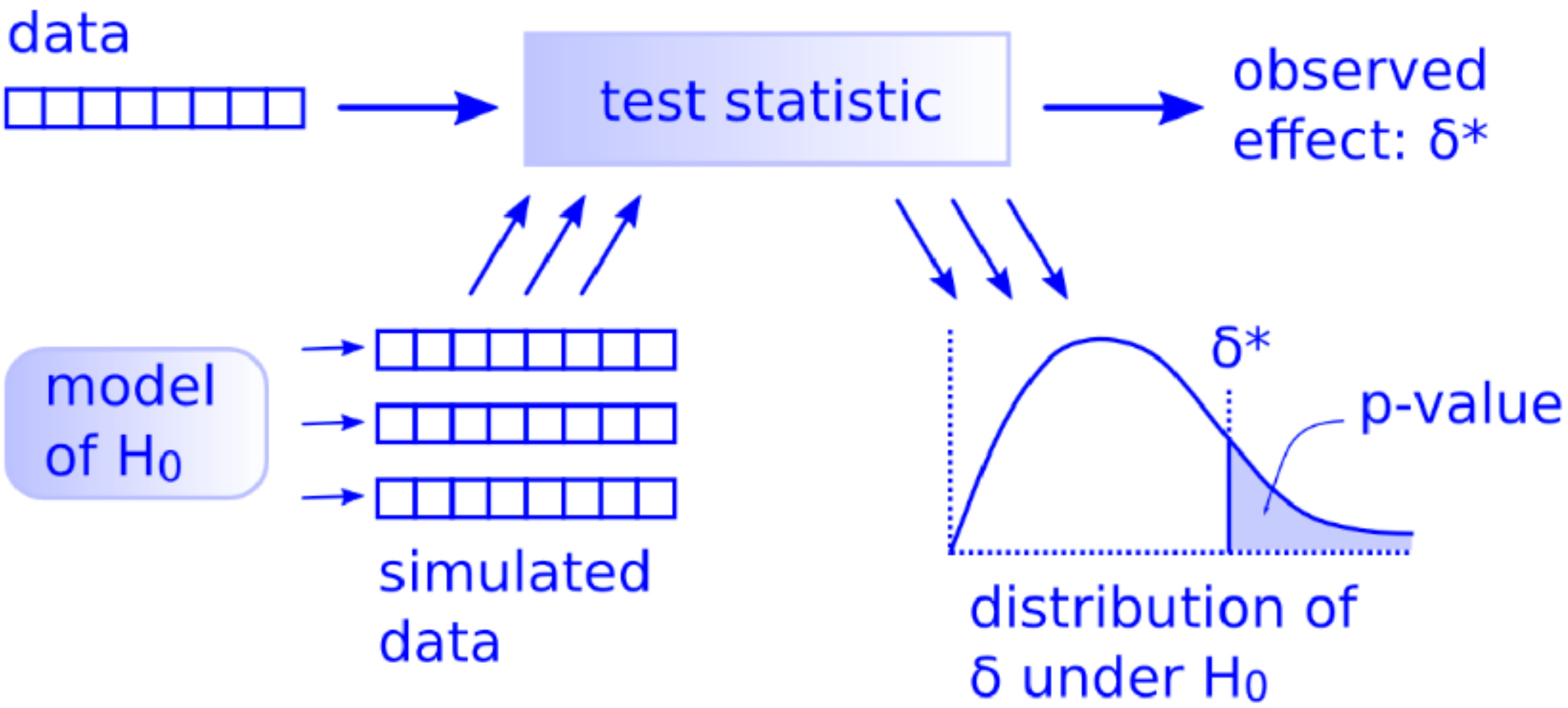
Computational

- Permutation
- Bootstrap
- Simulation

Allen Downey

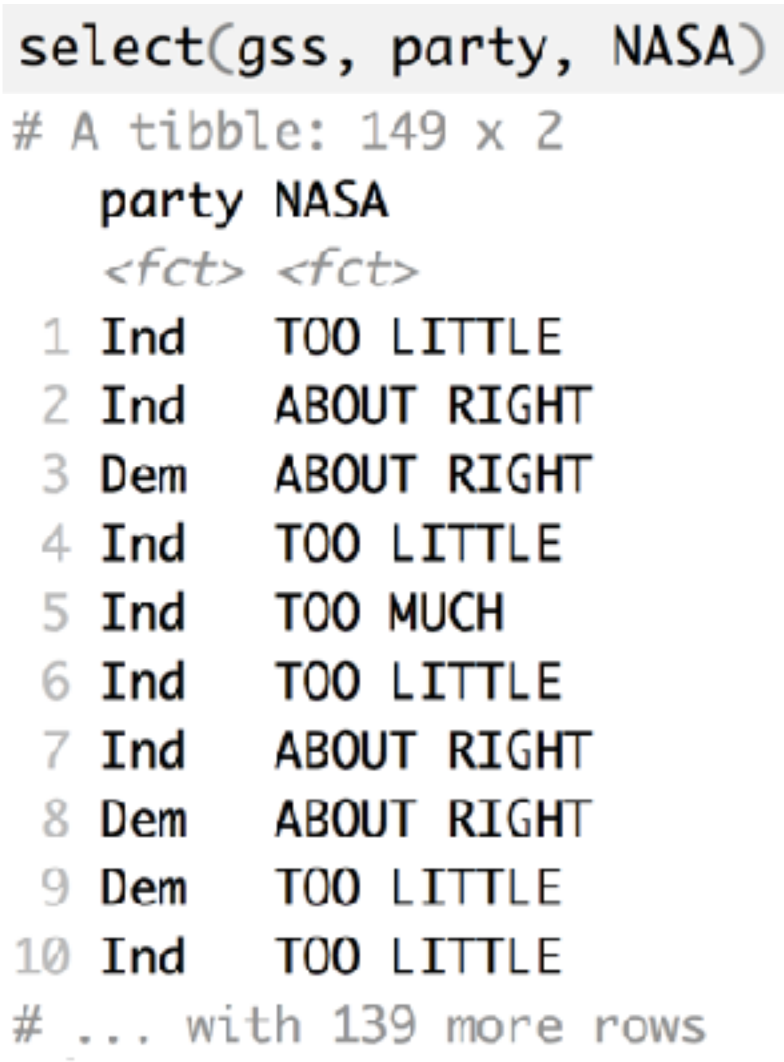
There is only one test

- Allen Downey

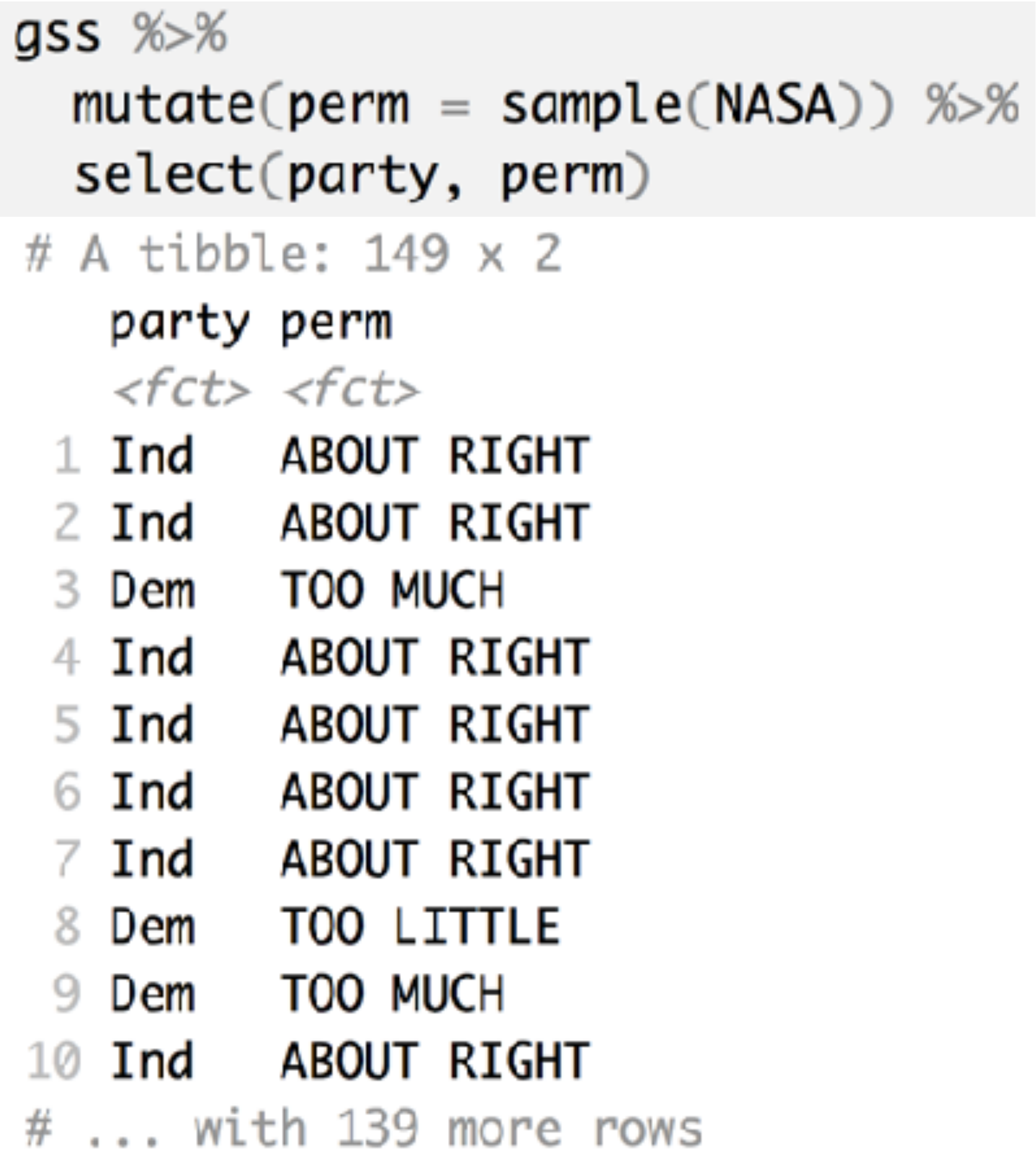


Simulation through Permutation

If we live in world where these variables are totally unrelated, the ties between variables are arbitrary, so they might just as well have been shuffled.



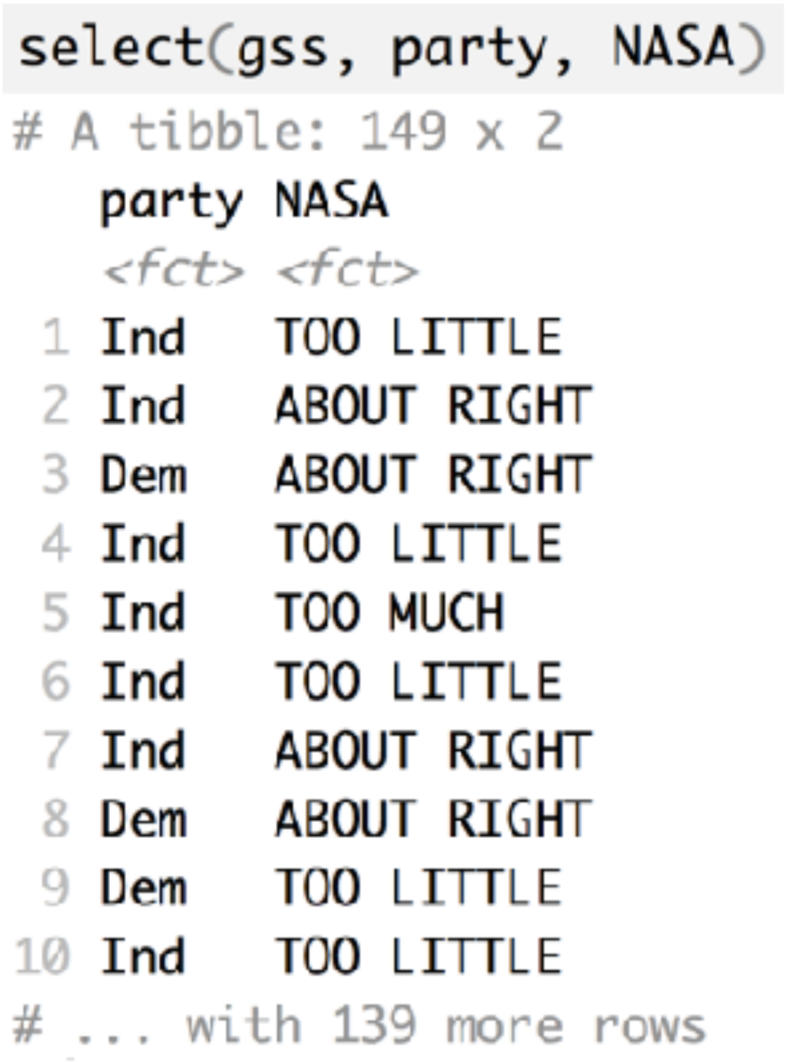
```
select(gss, party, NASA)
# A tibble: 149 x 2
   party NASA
   <fct> <fct>
 1 Ind   TOO LITTLE
 2 Ind   ABOUT RIGHT
 3 Dem   ABOUT RIGHT
 4 Ind   TOO LITTLE
 5 Ind   TOO MUCH
 6 Ind   TOO LITTLE
 7 Ind   ABOUT RIGHT
 8 Dem   ABOUT RIGHT
 9 Dem   TOO LITTLE
10 Ind   TOO LITTLE
# ... with 139 more rows
```



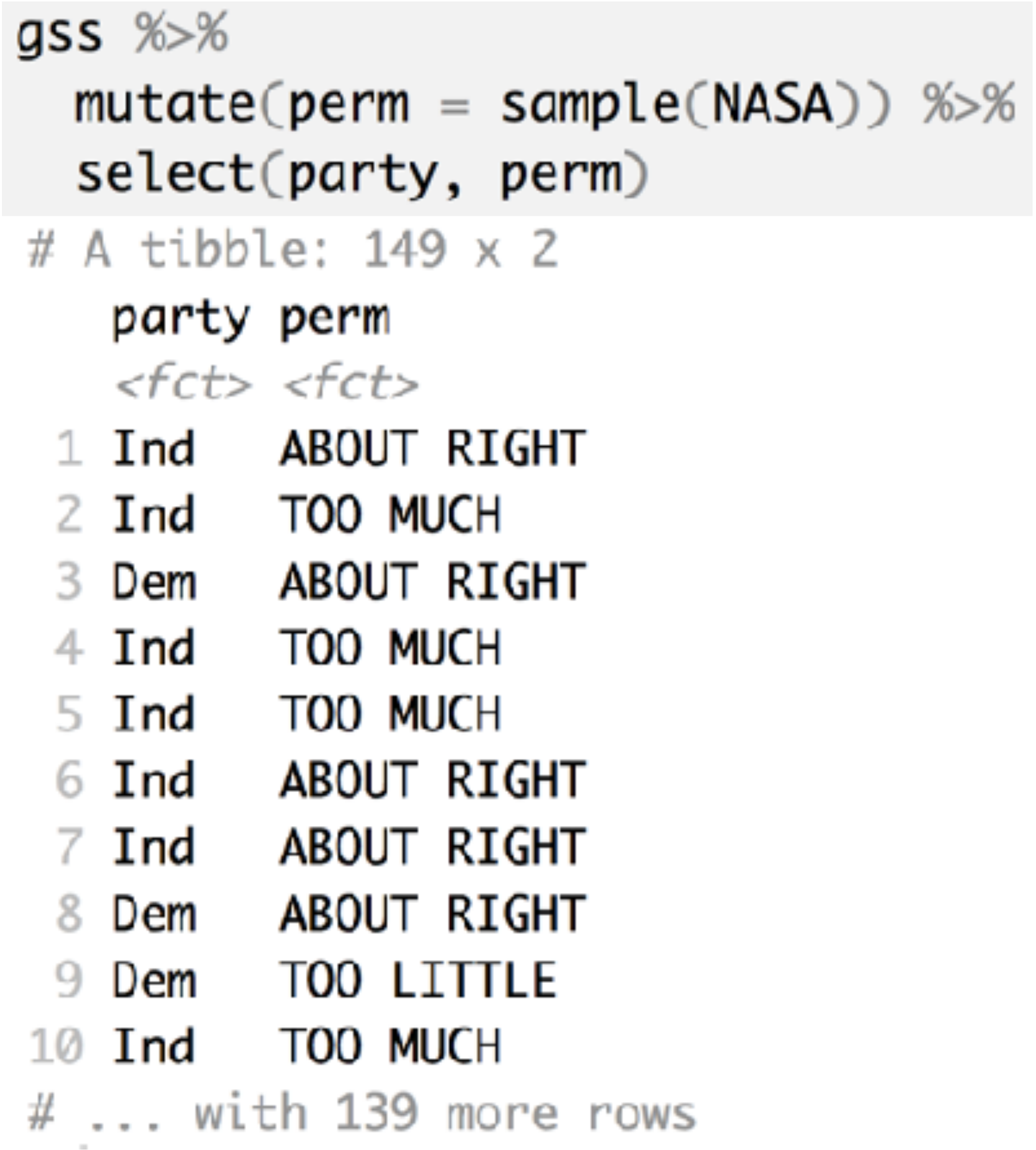
```
gss %>%
  mutate(perm = sample(NASA)) %>%
  select(party, perm)
# A tibble: 149 x 2
   party perm
   <fct> <fct>
 1 Ind   ABOUT RIGHT
 2 Ind   ABOUT RIGHT
 3 Dem   TOO MUCH
 4 Ind   ABOUT RIGHT
 5 Ind   ABOUT RIGHT
 6 Ind   ABOUT RIGHT
 7 Ind   ABOUT RIGHT
 8 Dem   TOO LITTLE
 9 Dem   TOO MUCH
10 Ind   ABOUT RIGHT
# ... with 139 more rows
```

Simulation through Permutation

If we live in world where these variables are totally unrelated, the ties between variables are arbitrary, so they might just as well have been shuffled.



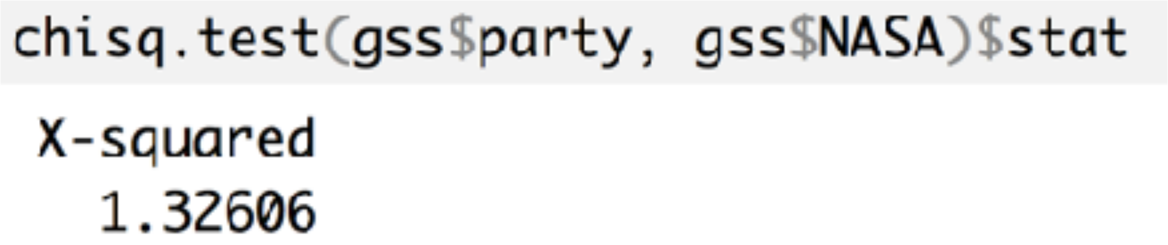
```
select(gss, party, NASA)
# A tibble: 149 x 2
   party NASA
   <fct> <fct>
 1 Ind   TOO LITTLE
 2 Ind   ABOUT RIGHT
 3 Dem   ABOUT RIGHT
 4 Ind   TOO LITTLE
 5 Ind   TOO MUCH
 6 Ind   TOO LITTLE
 7 Ind   ABOUT RIGHT
 8 Dem   ABOUT RIGHT
 9 Dem   TOO LITTLE
10 Ind   TOO LITTLE
# ... with 139 more rows
```



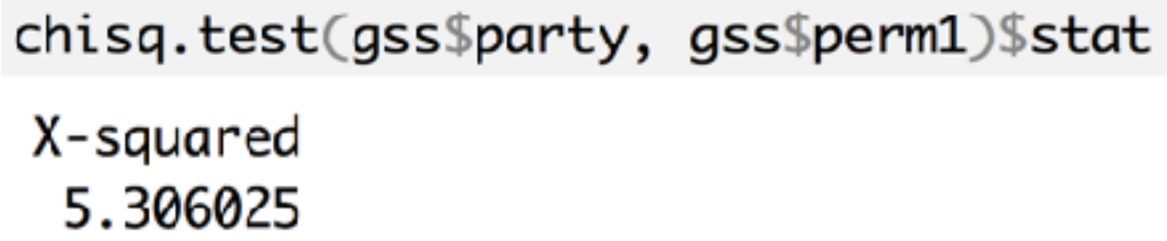
```
gss %>%
  mutate(perm = sample(NASA)) %>%
  select(party, perm)
# A tibble: 149 x 2
   party perm
   <fct> <fct>
 1 Ind   ABOUT RIGHT
 2 Ind   TOO MUCH
 3 Dem   ABOUT RIGHT
 4 Ind   TOO MUCH
 5 Ind   TOO MUCH
 6 Ind   ABOUT RIGHT
 7 Ind   ABOUT RIGHT
 8 Dem   ABOUT RIGHT
 9 Dem   TOO LITTLE
10 Ind   TOO MUCH
# ... with 139 more rows
```

Test statistic

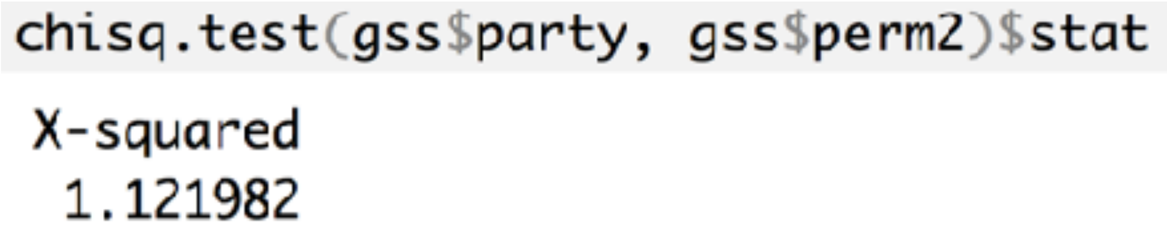
Chi-squared statistic: a measure of the difference between your data and what you would expect if the null hypothesis were true.



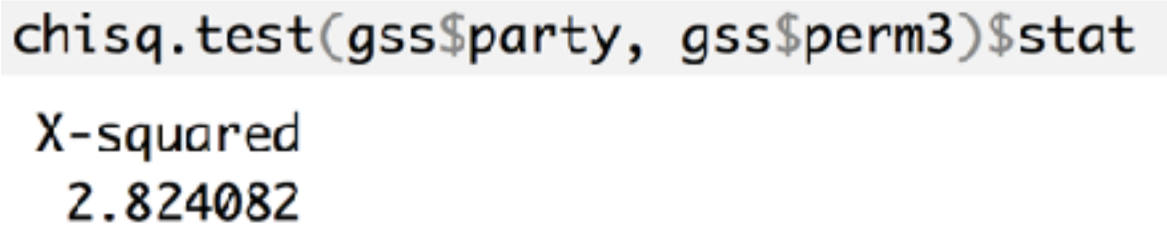
```
chisq.test(gss$party, gss$NASA)$stat
 X-squared
   1.32606
```



```
chisq.test(gss$party, gss$perm1)$stat
 X-squared
  5.306025
```



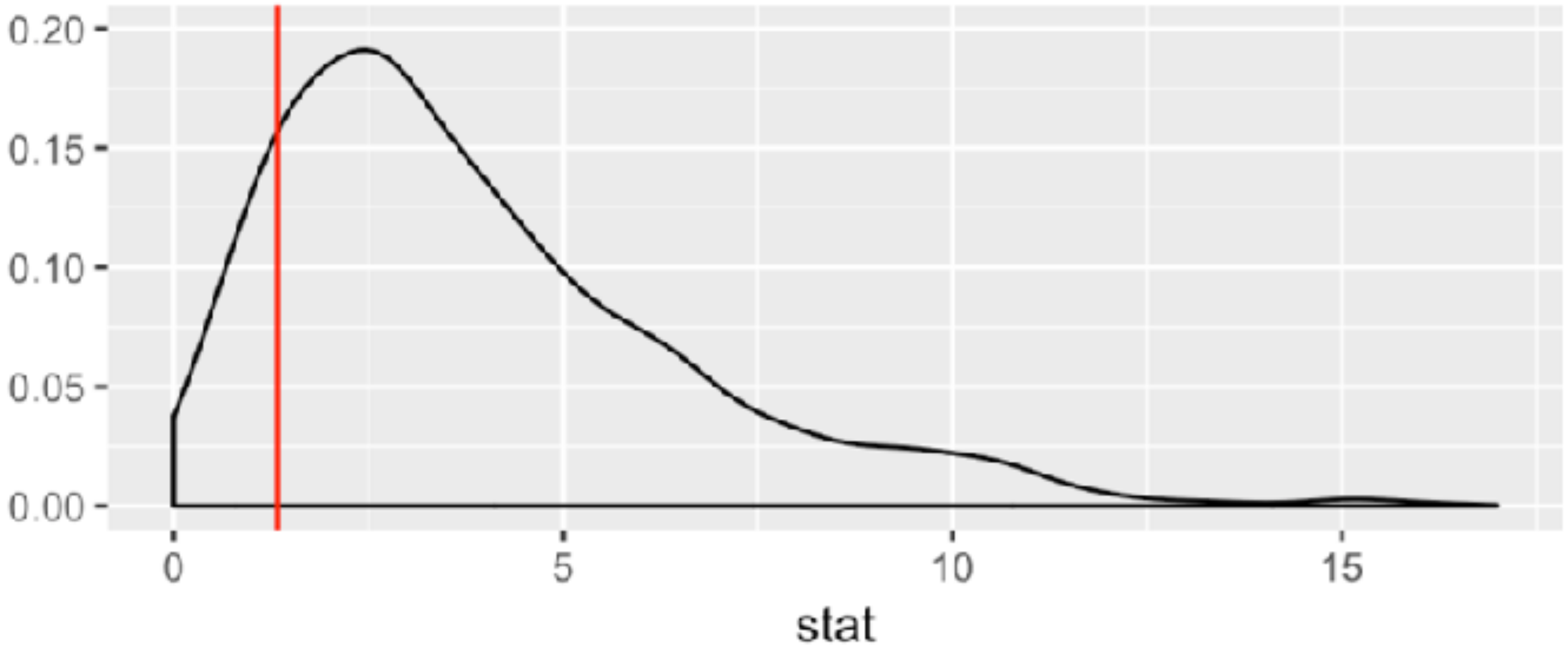
```
chisq.test(gss$party, gss$perm2)$stat
 X-squared
  1.121982
```



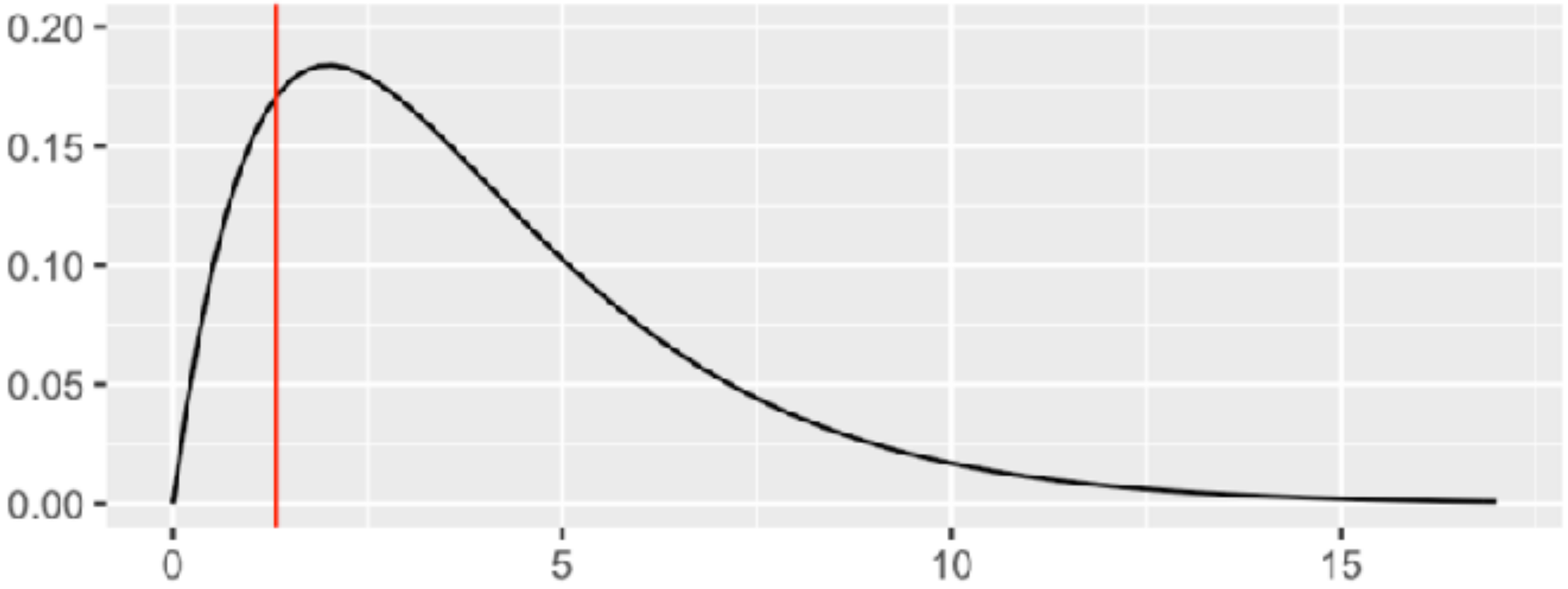
```
chisq.test(gss$party, gss$perm3)$stat
 X-squared
  2.824082
```

Distribution of statistic

via permutation

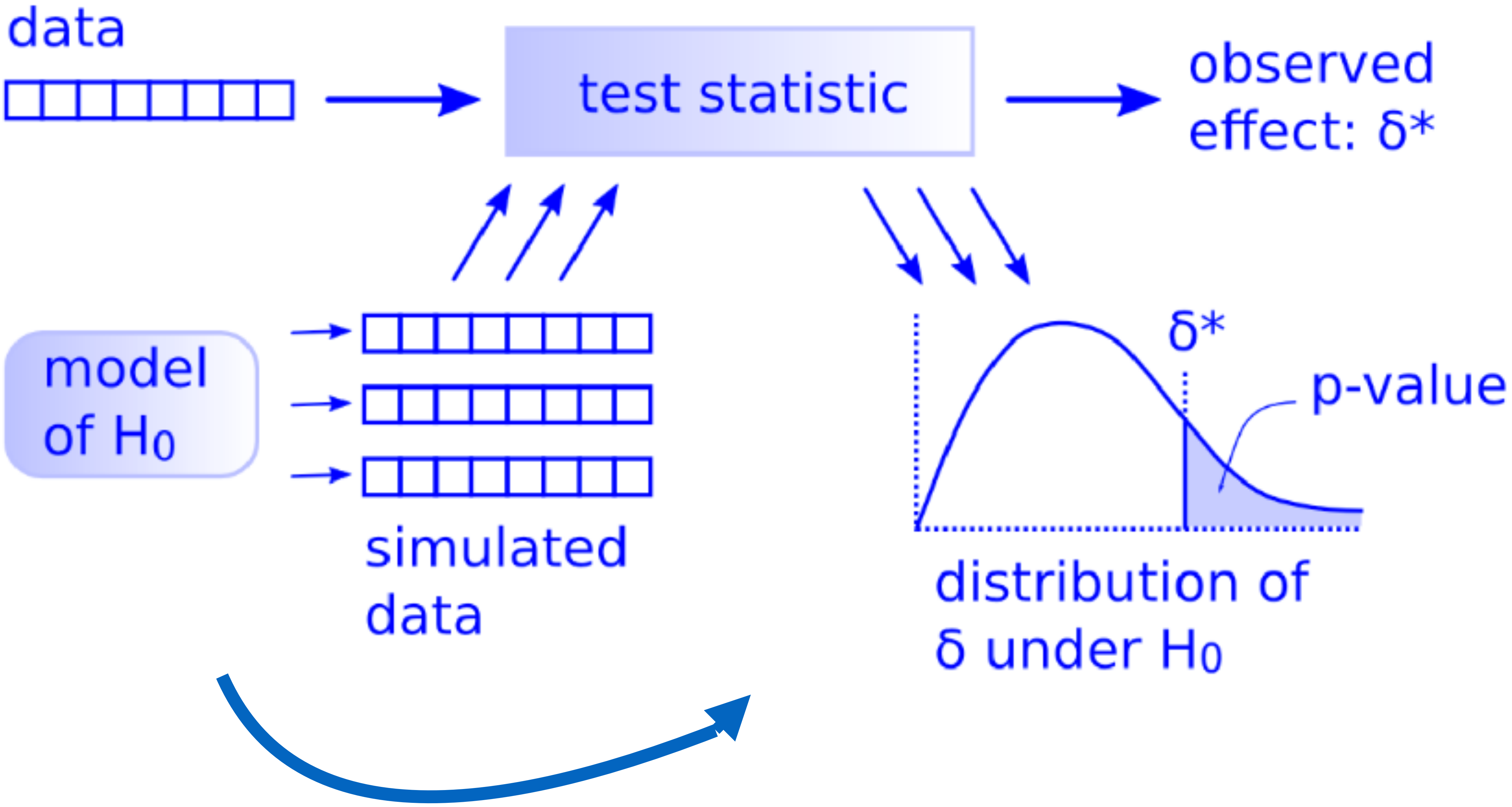


via approximation

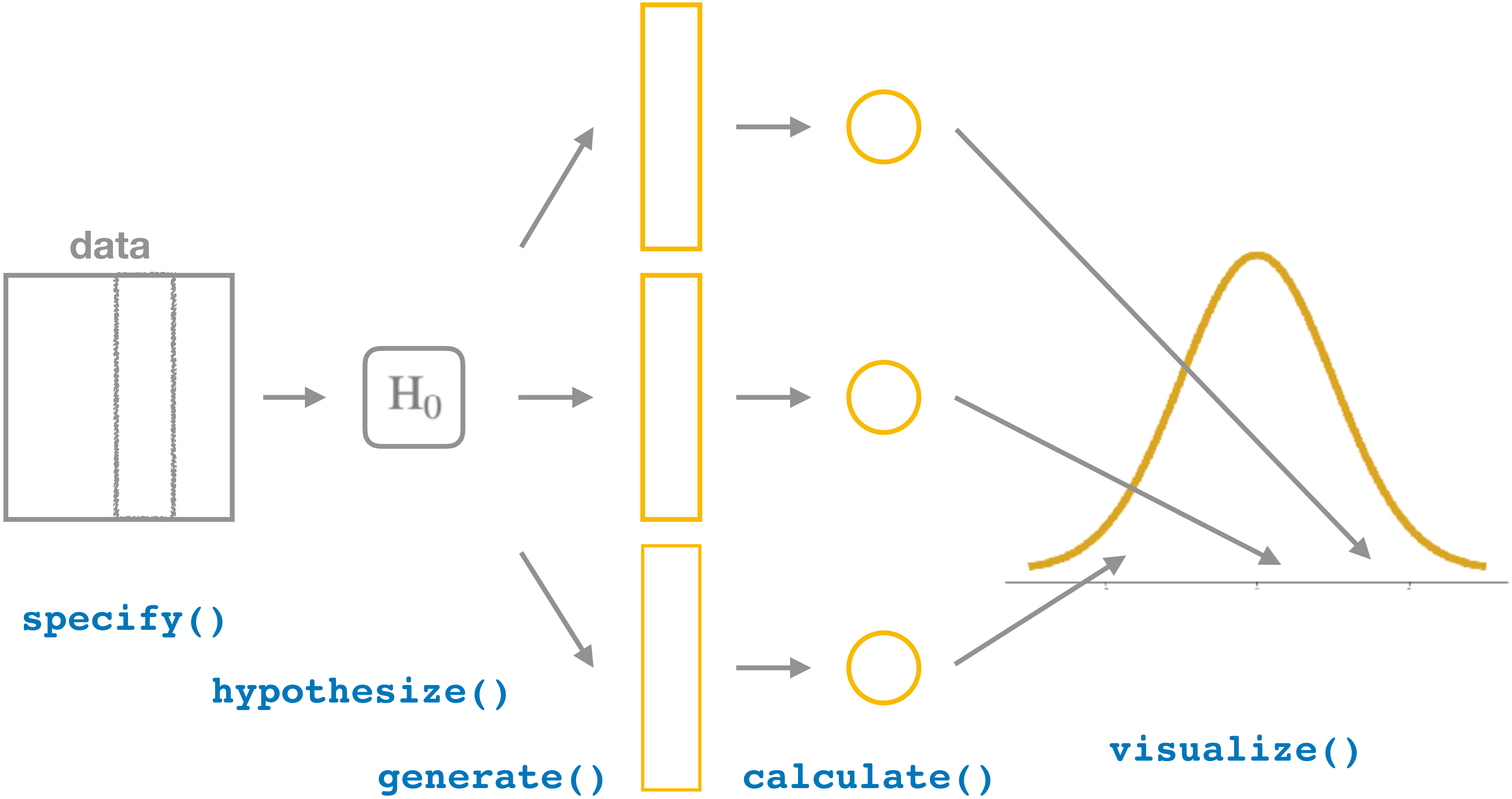


There is only one test

- Allen Downey



The `infer` verbs



data

H_0

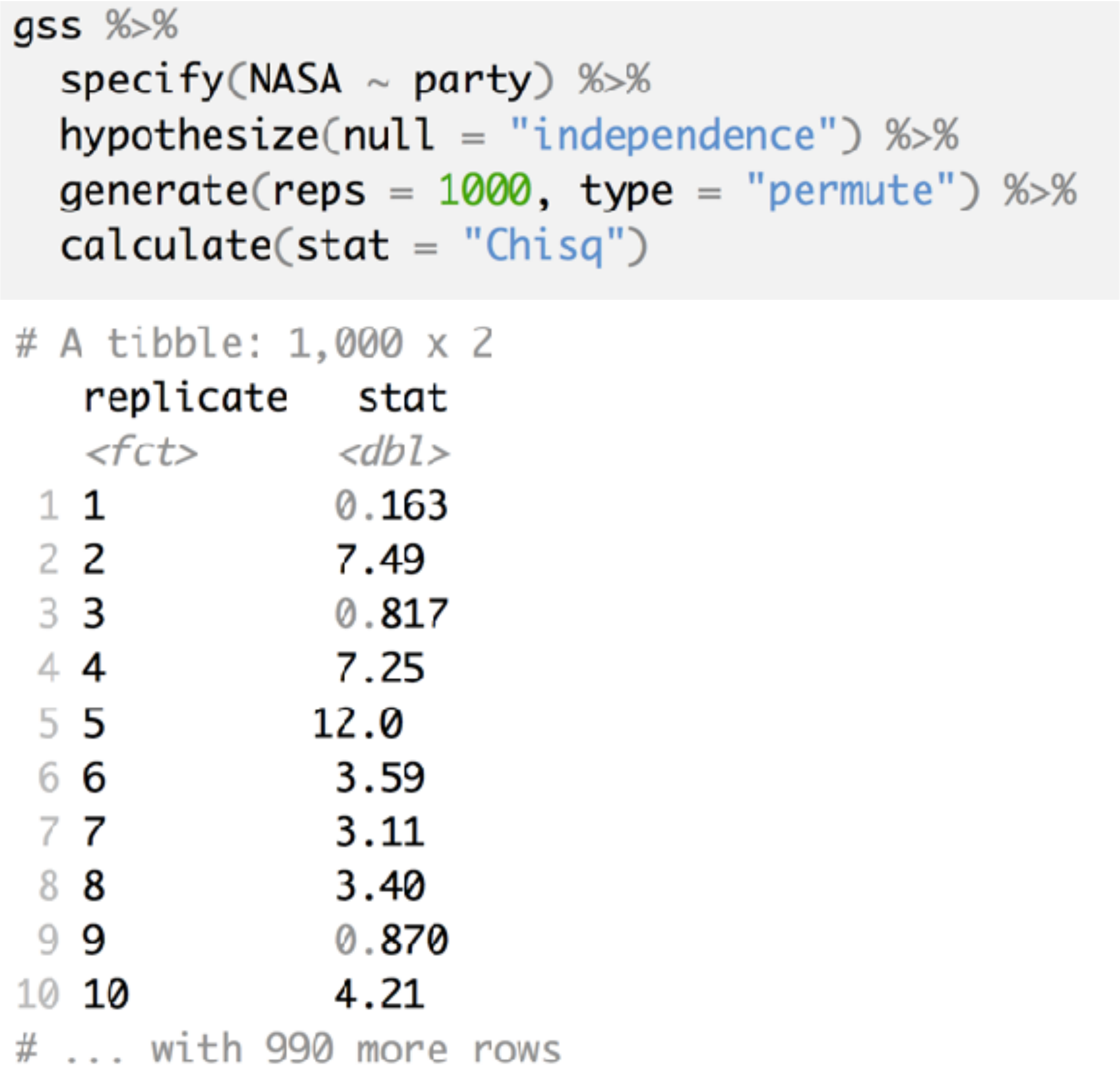
`specify()`

`hypothesize()`

`generate()`

`calculate()`

`visualize()`

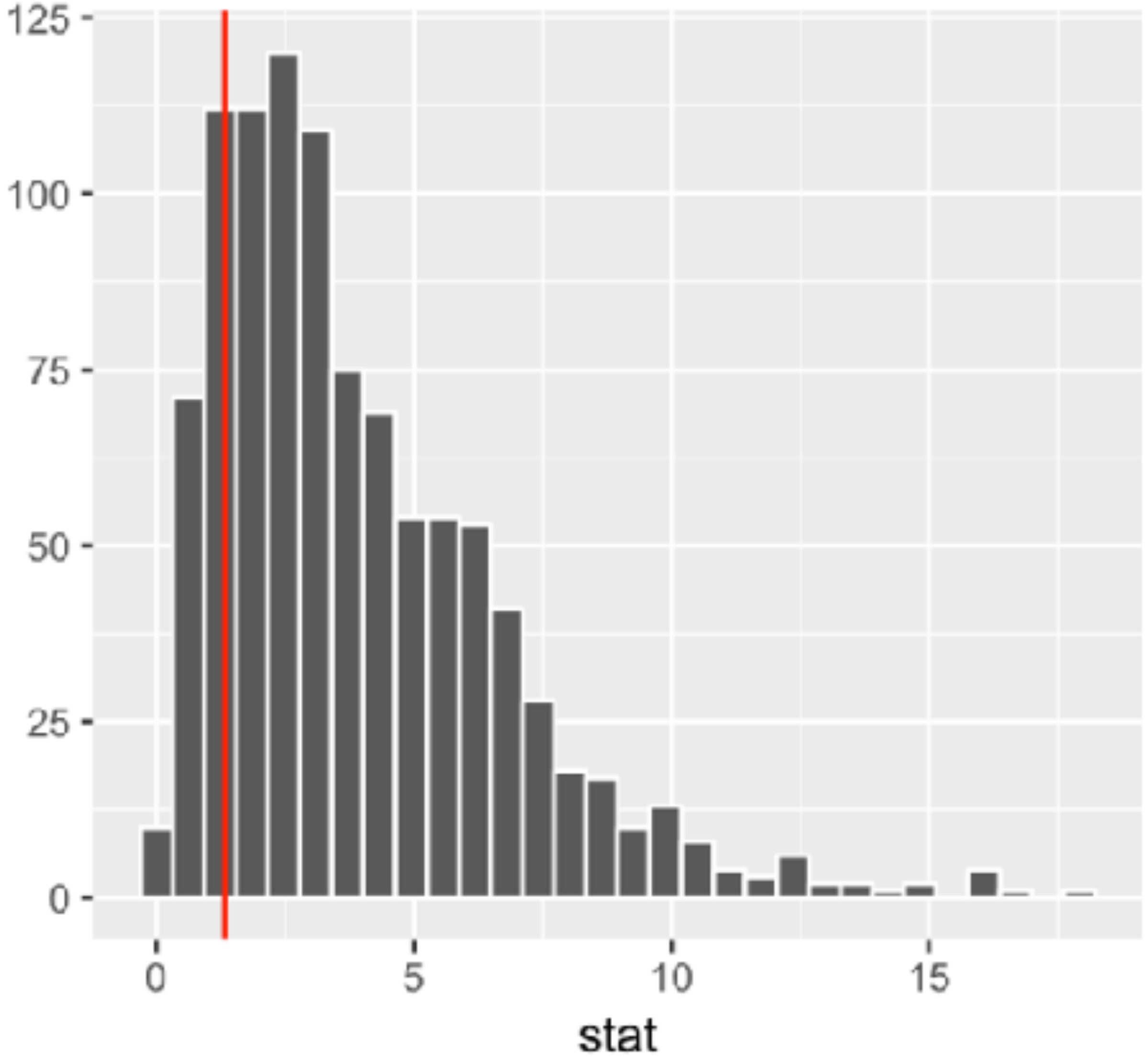


```r
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq")
```

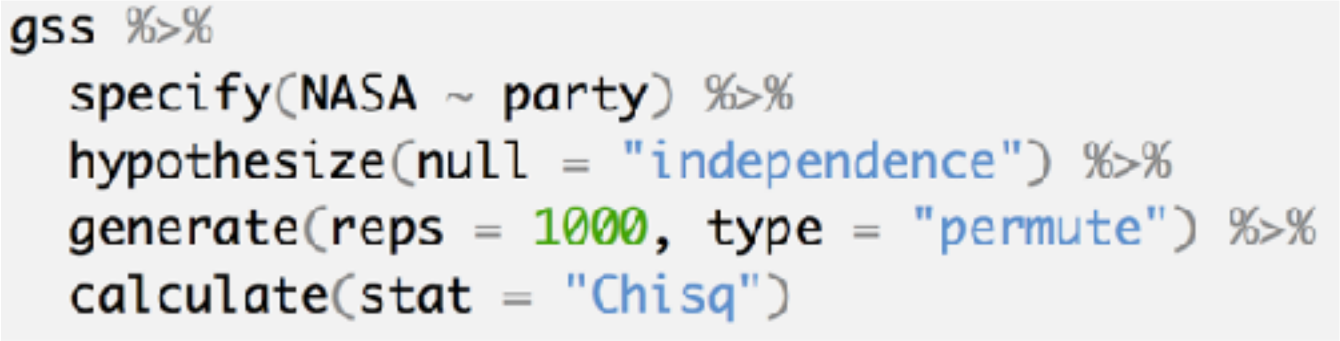
```
# A tibble: 1,000 x 2
   replicate   stat
   <fct>      <dbl>
 1 1          0.163
 2 2          7.49
 3 3          0.817
 4 4          7.25
 5 5         12.0
 6 6          3.59
 7 7          3.11
 8 8          3.40
 9 9          0.870
10 10         4.21
# ... with 990 more rows
```

```r
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq") %>%
  summarize(p_val = mean(stat > obs_stat))
```

```
# A tibble: 1 x 1
  p_val
  <dbl>
1 0.864
```

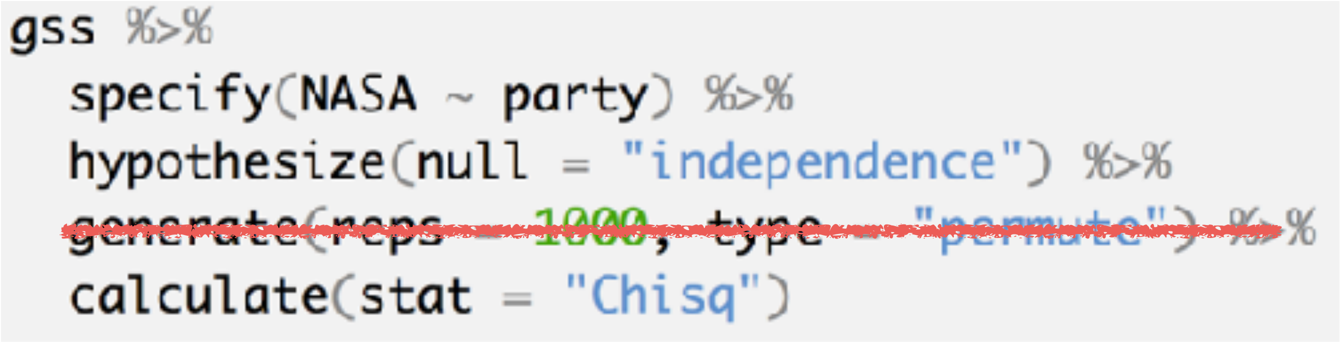


Reusable parts



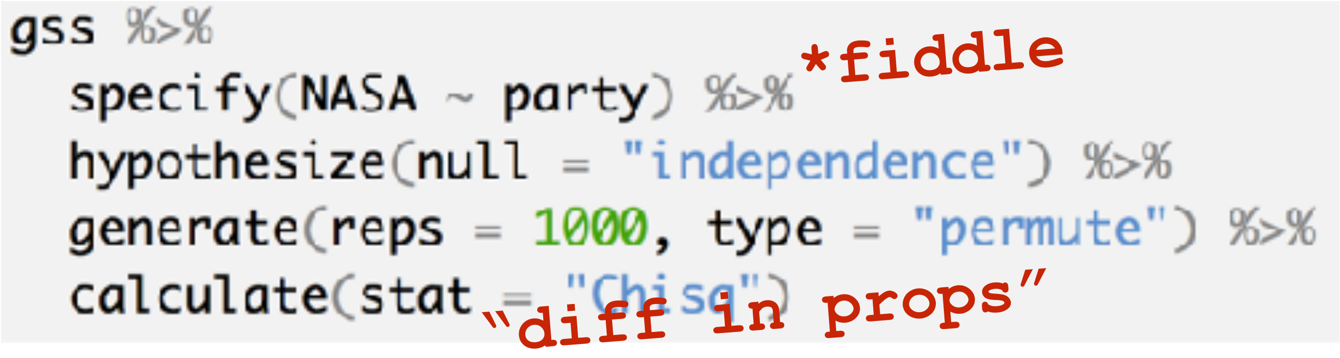
```
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq")
```

Permutation Chi-squared



```
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  ~~generate(reps = 1000, type = "permute") %>%~~
  calculate(stat = "Chisq")
```

Approximation Chi-squared*

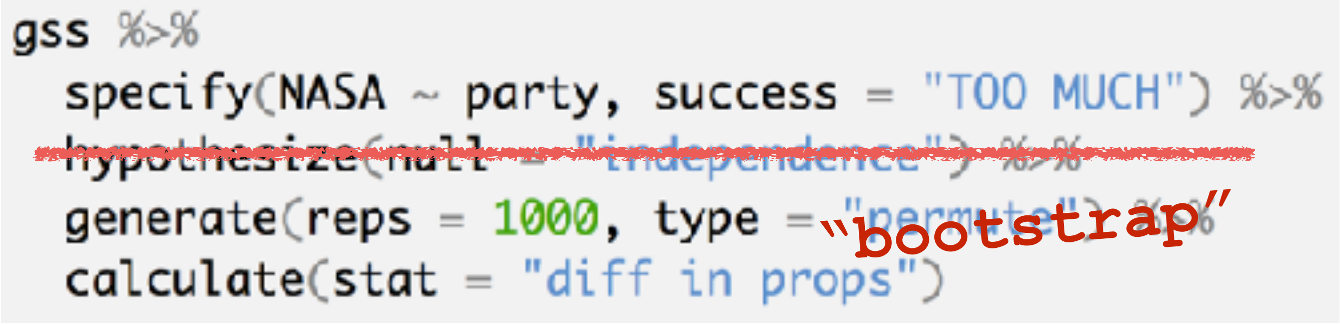


```
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq")
```

*fiddle

"diff in props"

Permutation p1 - p2

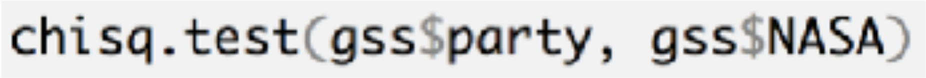


```
gss %>%
  specify(NASA ~ party, success = "TOO MUCH") %>%
  ~~hypothesize(null = "independence") %>%~~
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in props")
```

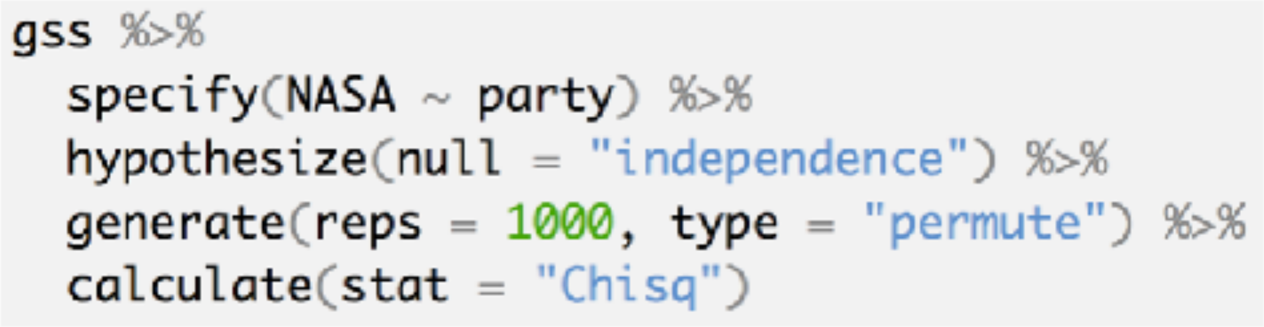
"bootstrap"

Confidence interval for p1 - p2

The goal of this presentation



```r
chisq.test(gss$party, gss$NASA)
```



```r
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq")
```

- Thanks to Chester Ismay, Ben Baumer, Mine Cetinkaya-Rundel, Jo Hardin, and the other contributors.
- website: infer.netlify.com