infer

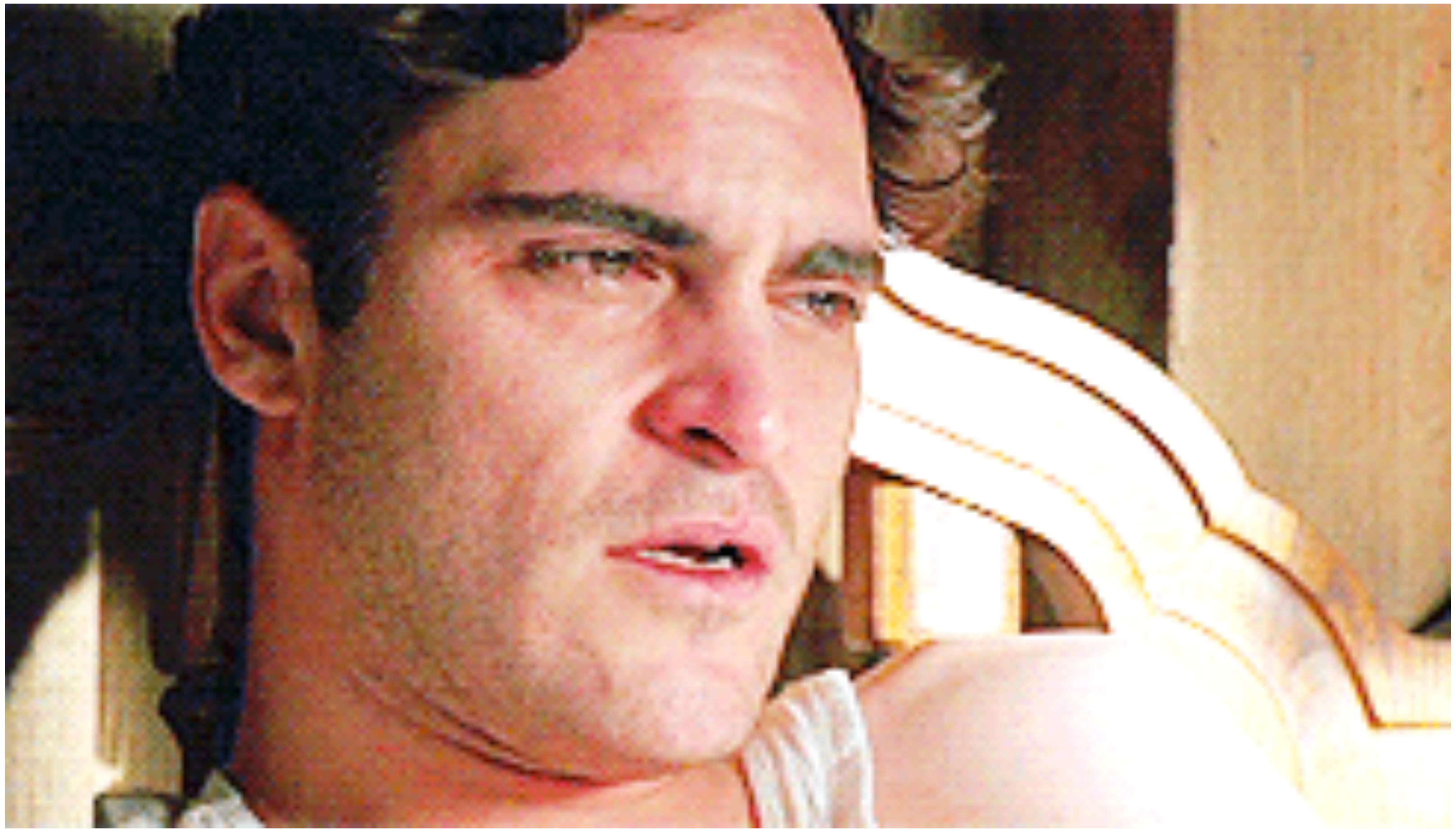
an R package for tidy statistical inference

Andrew Bray

infer.netlify.com

infer makes p-values easier to compute.

infer makes p-values easier to compute.



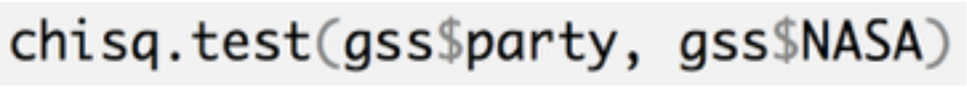
infer makes p-values easier to compute.

infer makes ~~p-values~~ statistical inference easier to compute.

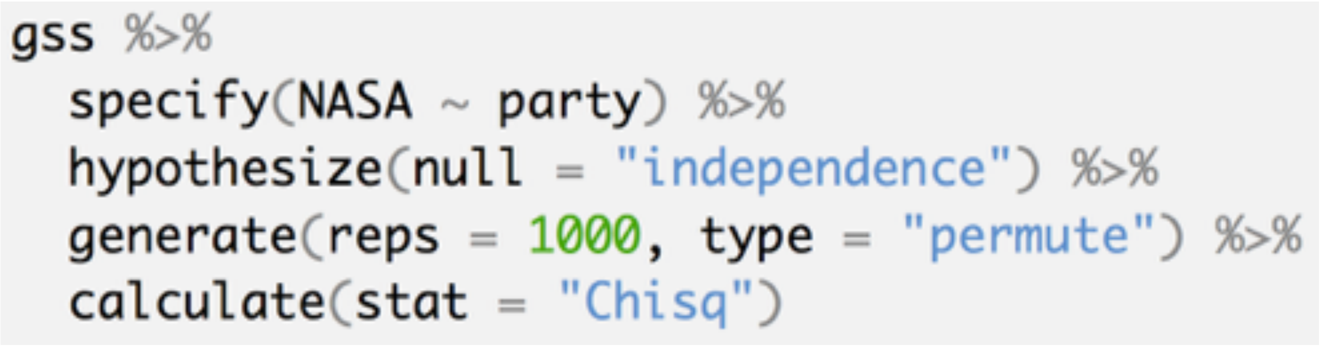
statistical inference

infer makes ~~p-values easier to compute.~~ tidy and transparent.

infer makes ~~p-values~~ statistical inference ~~easier to compute~~. tidy and transparent.



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



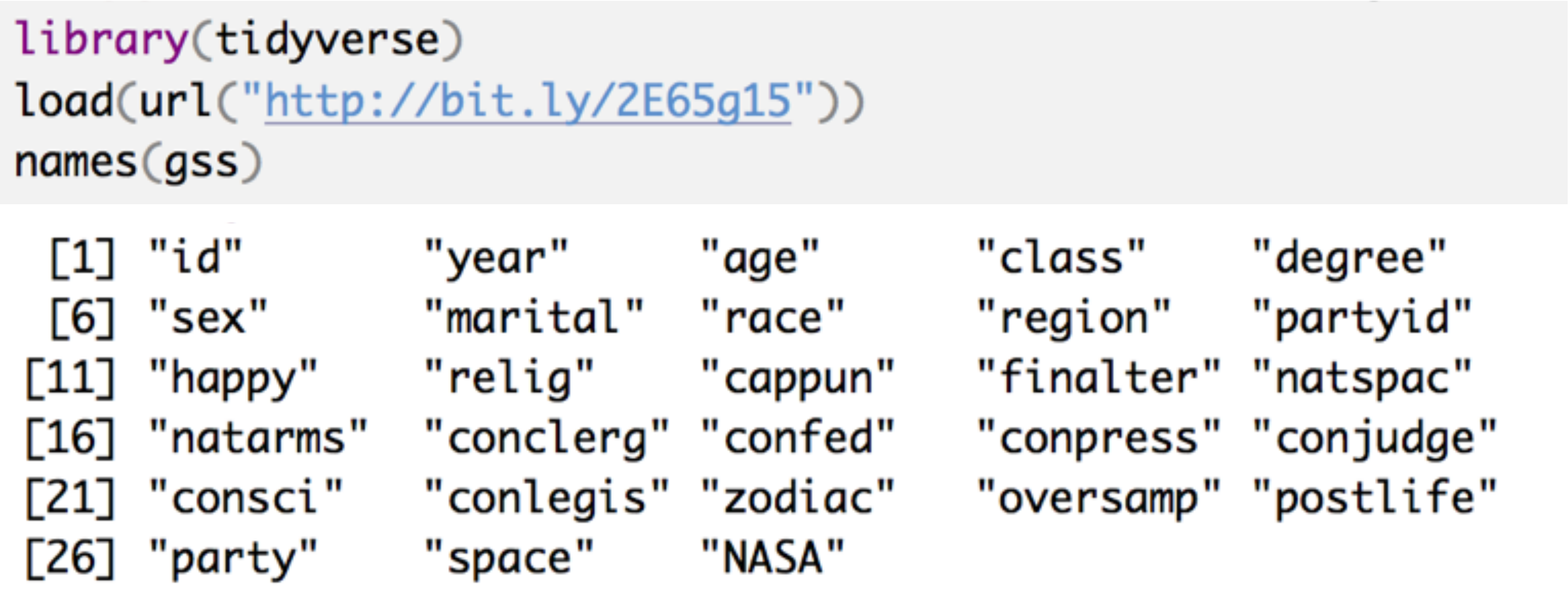
```
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq")
```

statistical
inference

infer makes ~~p-values~~
~~easier to compute.~~
tidy and
transparent.

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

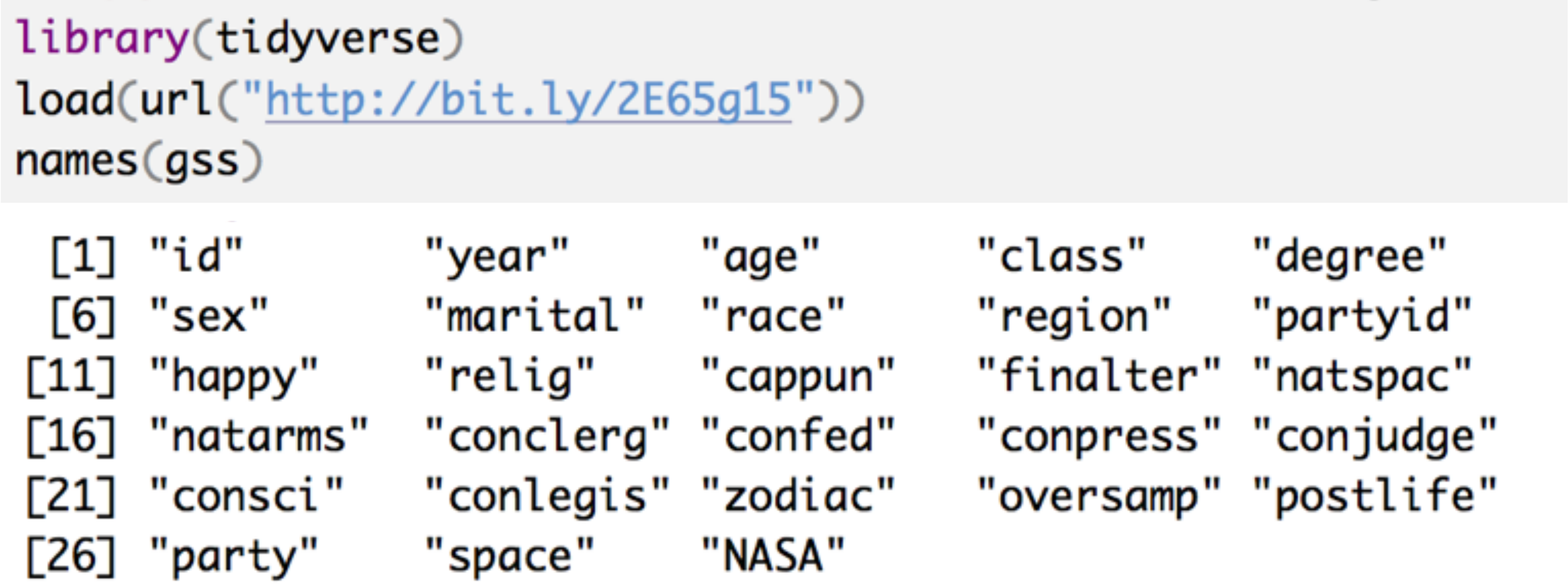
Case study: Is funding for space exploration a partisan issue?



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

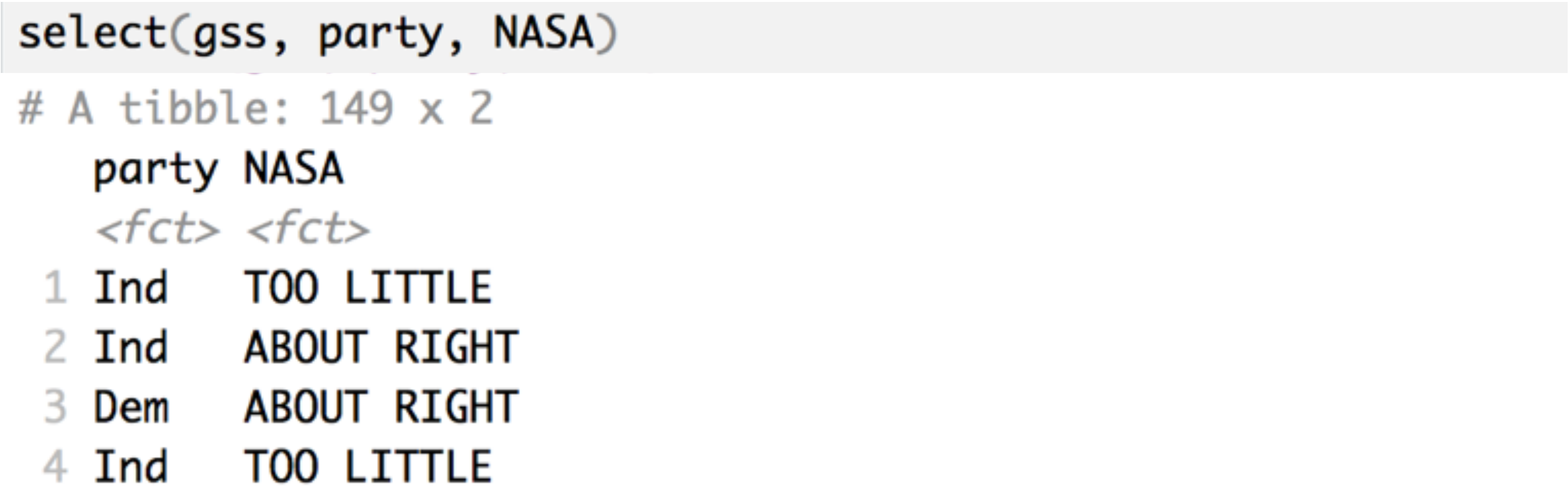
```
 [1] "id"        "year"      "age"       "class"     "degree"   
 [6] "sex"       "marital"   "race"      "region"    "partyid"  
[11] "happy"     "relig"     "cappun"    "finalter"  "natspac"  
[16] "natarms"   "conclerg"  "confed"    "conpress"  "conjudge" 
[21] "consci"    "conlegis"  "zodiac"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

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"
```

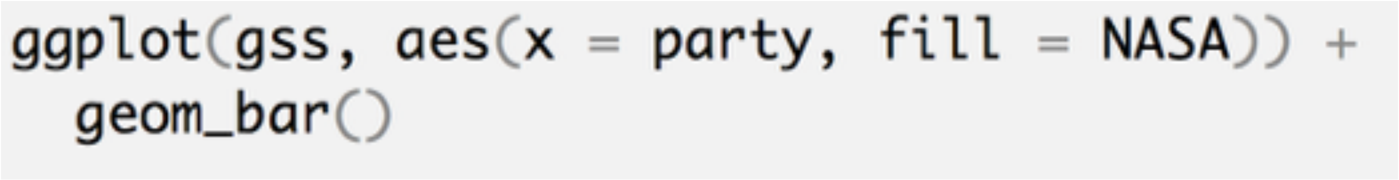


```
select(gss, party, NASA)
```

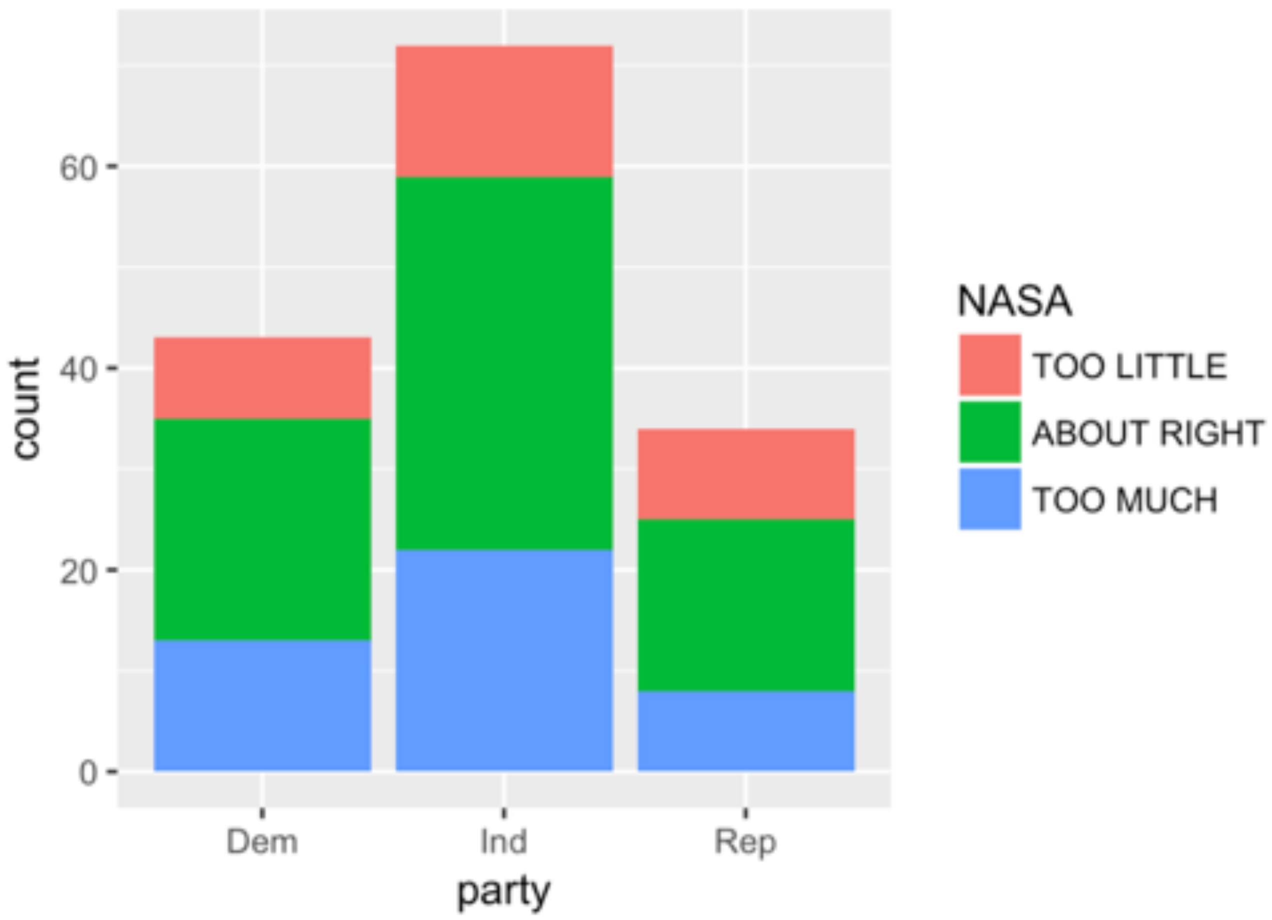
```
# A tibble: 149 x 2
   party NASA       
   <fct> <fct>      
 1 Ind   TOO LITTLE 
 2 Ind   ABOUT RIGHT
 3 Dem   ABOUT RIGHT
 4 Ind   TOO LITTLE 
```

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

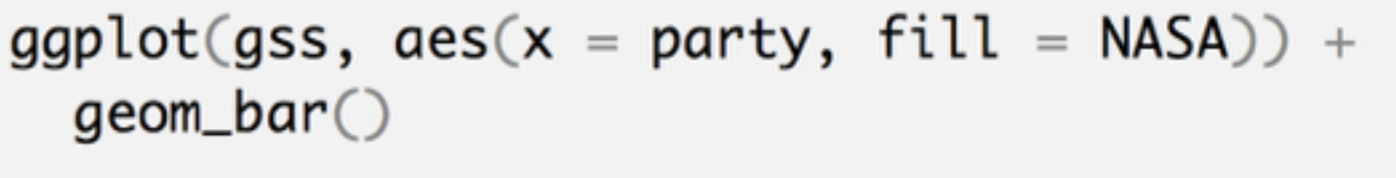
Case study: Is funding for space exploration a partisan issue?



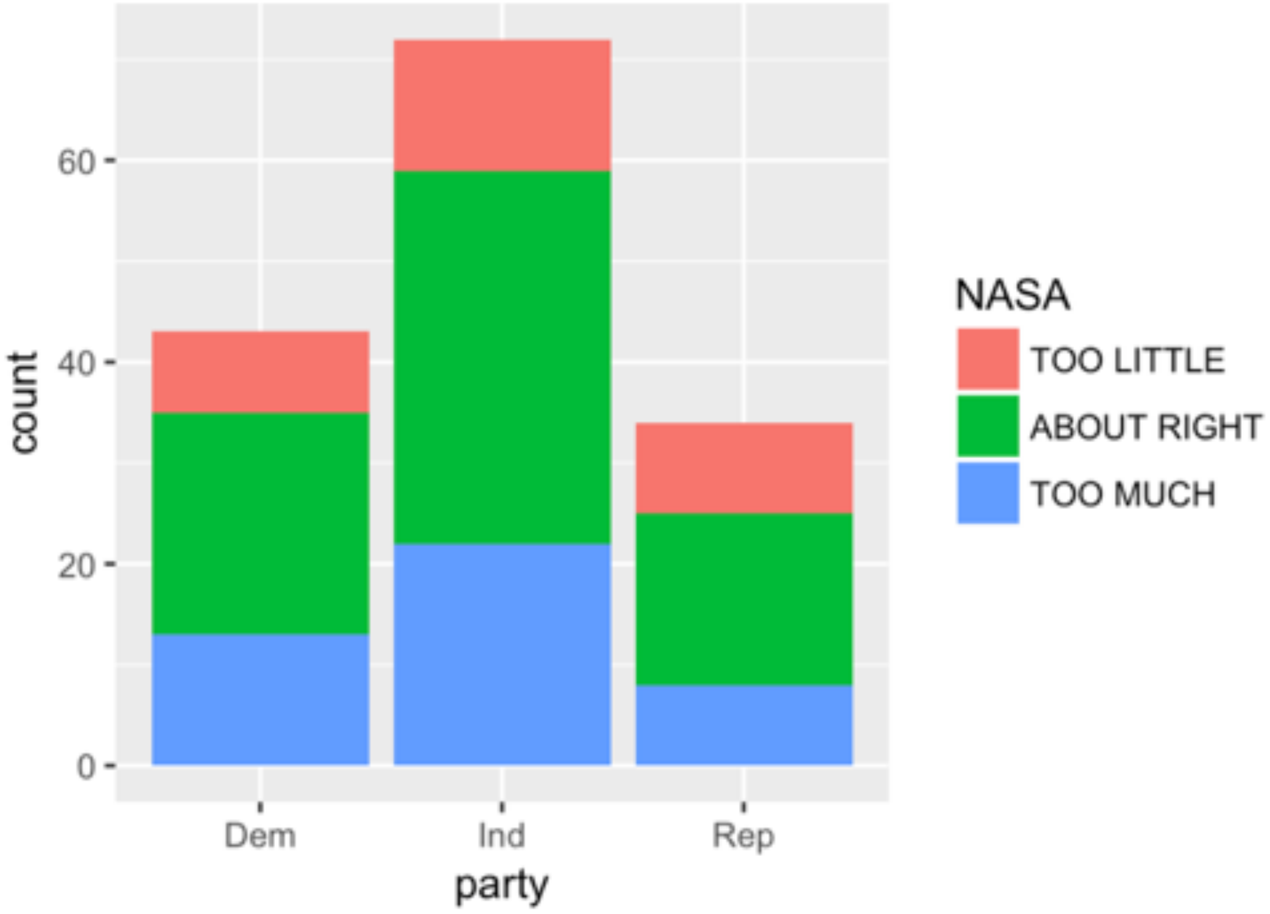
```
ggplot(gss, aes(x = party, fill = NASA)) +
  geom_bar()
```



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

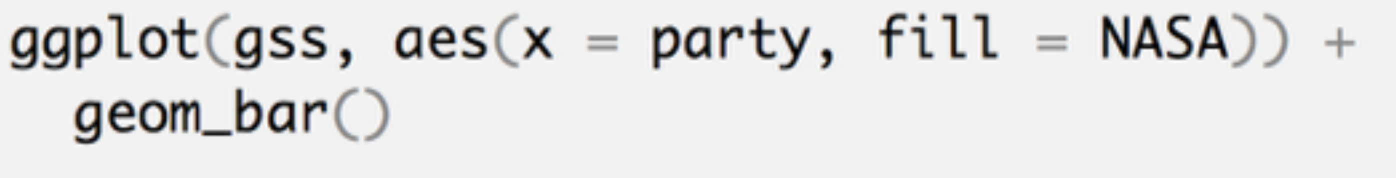


```
ggplot(gss, aes(x = party, fill = NASA)) +
  geom_bar()
```

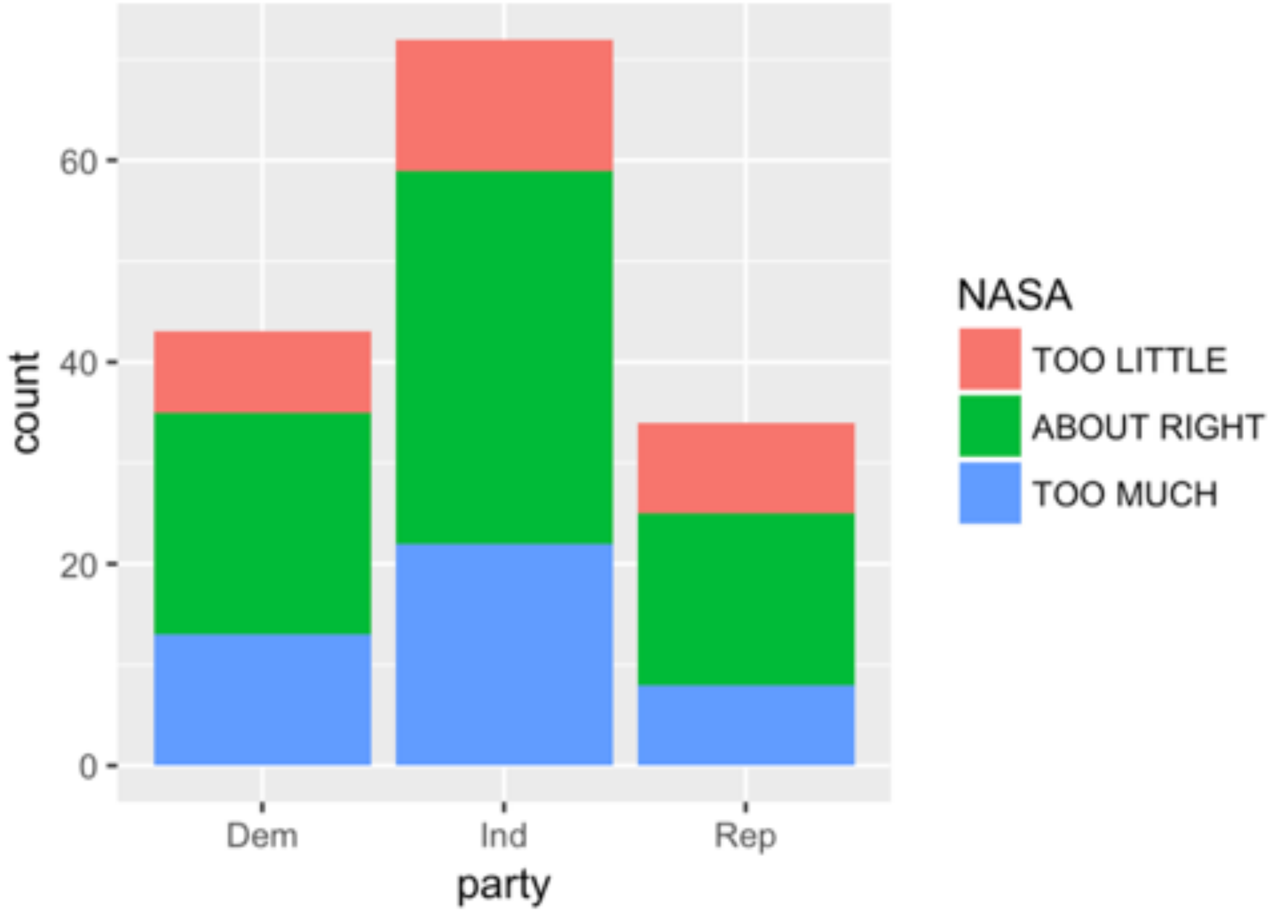


How can we test to see if the structure that we see is *significant*?

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

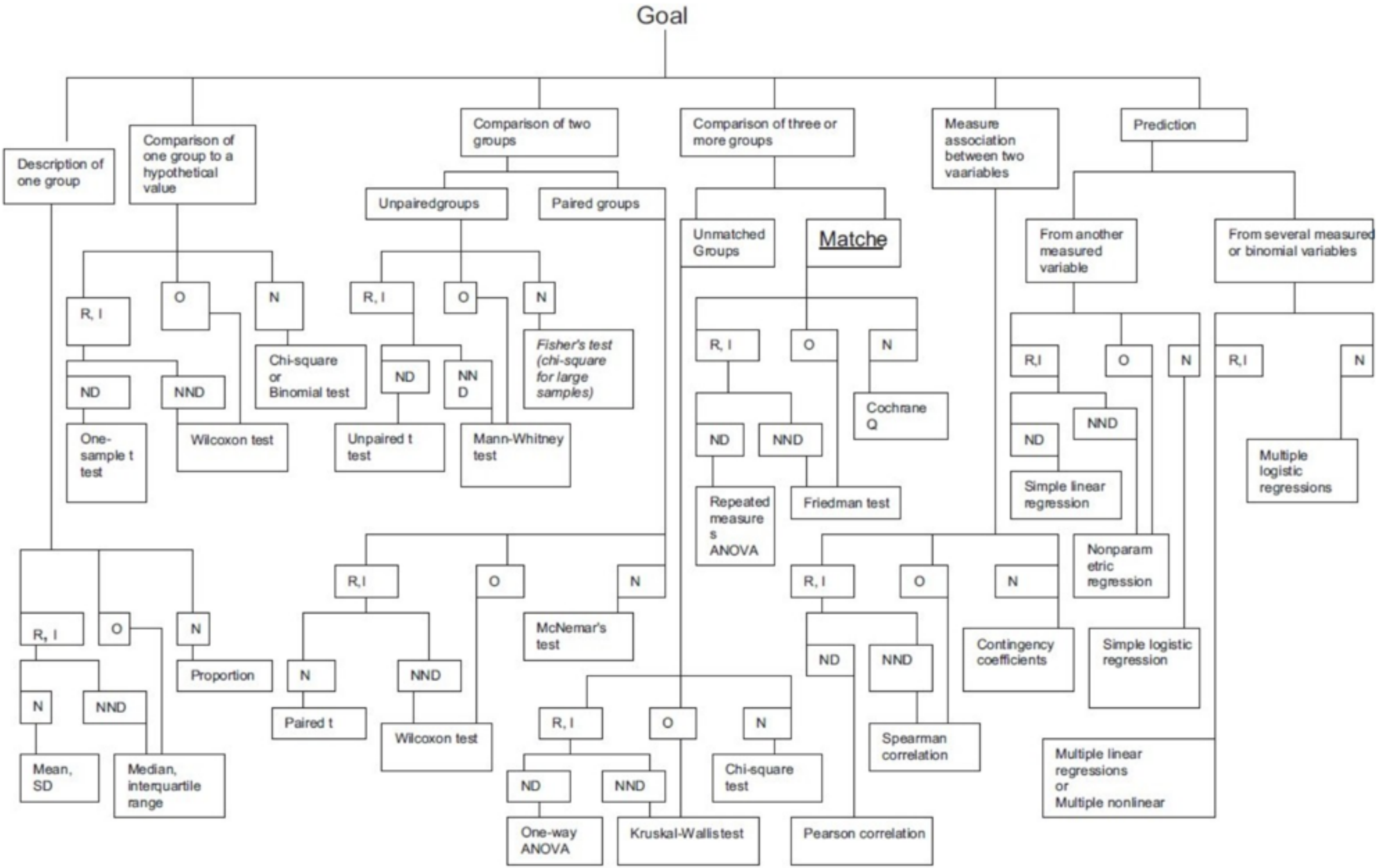


```
ggplot(gss, aes(x = party, fill = NASA)) +
  geom_bar()
```



How can we test to see if the structure that we see is *significant*?

Run a hypothesis test!



Goal

Description of one group

R, I

N

Mean, SD

NND

Median, interquartile range

O

N

Proportion

Comparison of one group to a hypothetical value

R, I

ND

One-sample t test

NND

Wilcoxon test

O

N

Chi-square or Binomial test

Comparison of two groups

Unpairedgroups

R, I

ND

Unpaired t test

NN D

Mann-Whitney test

O

N

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

Paired groups

R,I

N

Paired t

NND

Wilcoxon test

O

N

McNemar's test

Comparison of three or more groups

Unmatched Groups

R, I

ND

One-way ANOVA

NND

Kruskal-Wallistest

O

N

Chi-square test

Matche

R, I

ND

Repeated measure s ANOVA

NND

Friedman test

O

N

Cochrane Q

Measure association between two vaariables

R, I

ND

Pearson correlation

NND

Spearman correlation

O

N

Contingency coefficients

Prediction

From another measured variable

R,I

ND

Simple linear regression

NND

Nonparam etric regression

O

N

Simple logistic regression

From several measured or binomial variables

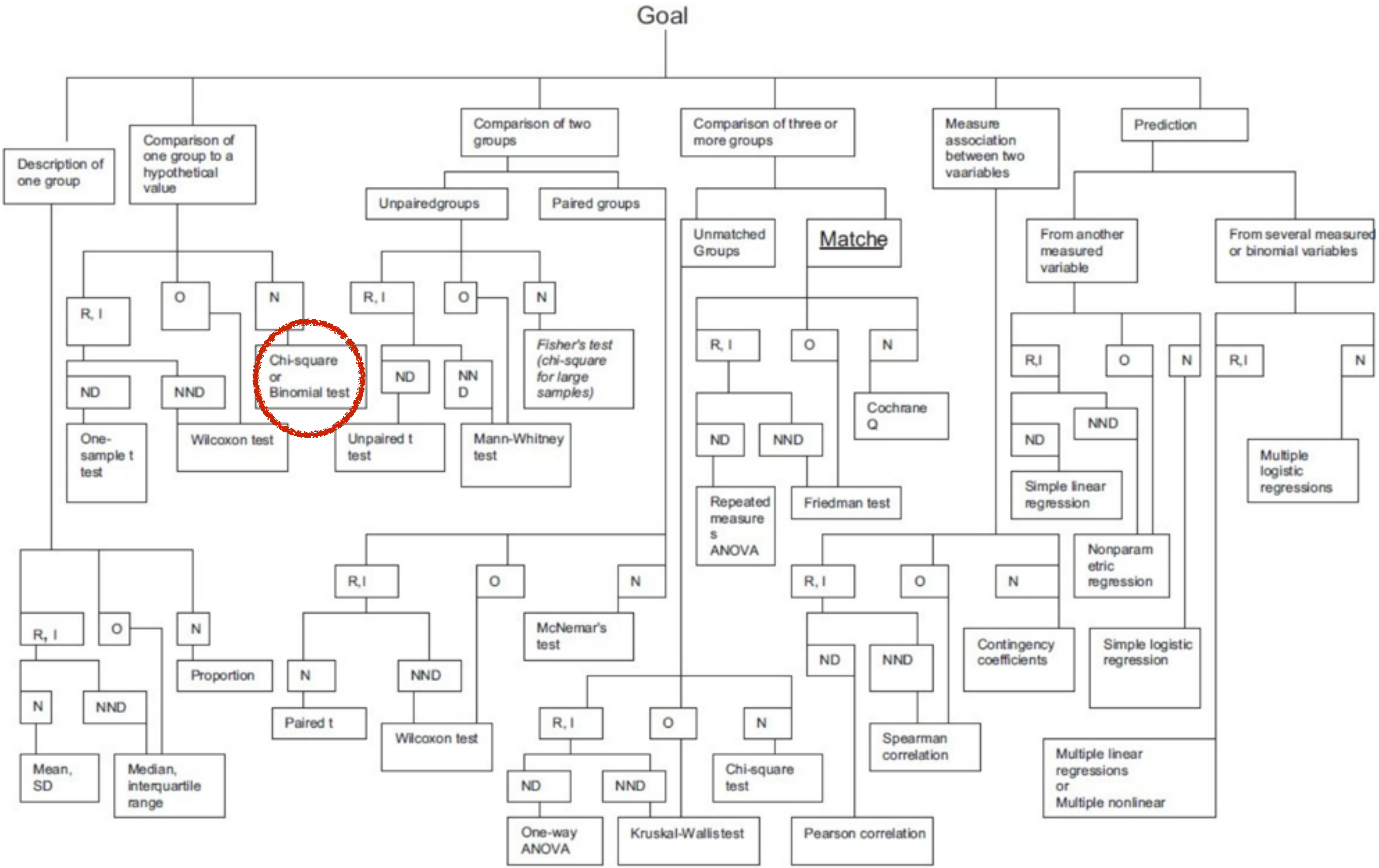
R,I

Multiple linear regressions or Multiple nonlinear

N

Multiple logistic regressions

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



Goal

Description of one group

R, I

N

NND

Mean, SD

O

Median, interquartile range

N

Proportion

Comparison of one group to a hypothetical value

R, I

ND

NND

One-sample t test

O

Wilcoxon test

N

Chi-square or Binomial test

Comparison of two groups

Unpairedgroups

R, I

ND

NN D

Unpaired t test

O

Mann-Whitney test

N

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

Paired groups

R,I

N

NND

Paired t

Wilcoxon test

O

N

McNemar's test

Comparison of three or more groups

Unmatched Groups

R, I

ND

NND

One-way ANOVA

O

Kruskal-Wallistest

N

Chi-square test

Matche

R, I

ND

NND

Repeated measure s ANOVA

O

Friedman test

N

Cochrane Q

Measure association between two vaariables

R, I

ND

NND

Pearson correlation

O

Spearman correlation

N

Contingency coefficients

Prediction

From another measured variable

R,I

ND

NND

Simple linear regression

O

Nonparametric regression

N

Simple logistic regression

From several measured or binomial variables

R,I

Multiple linear regressions or Multiple nonlinear

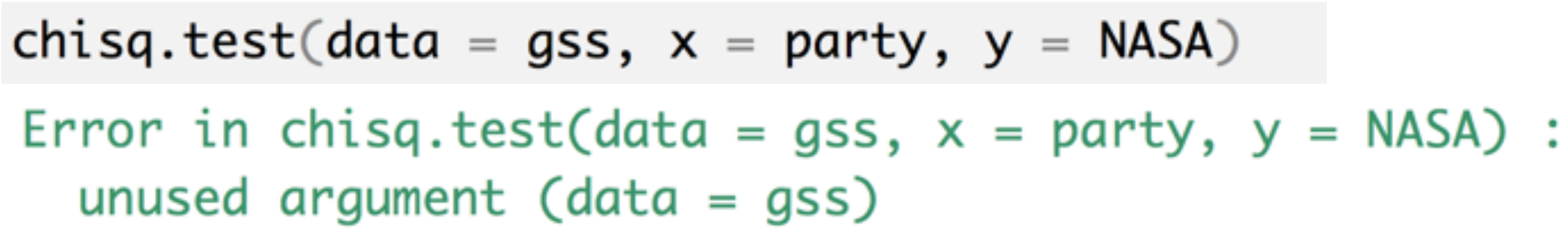
N

Multiple logistic regressions

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

Optimistic effort I

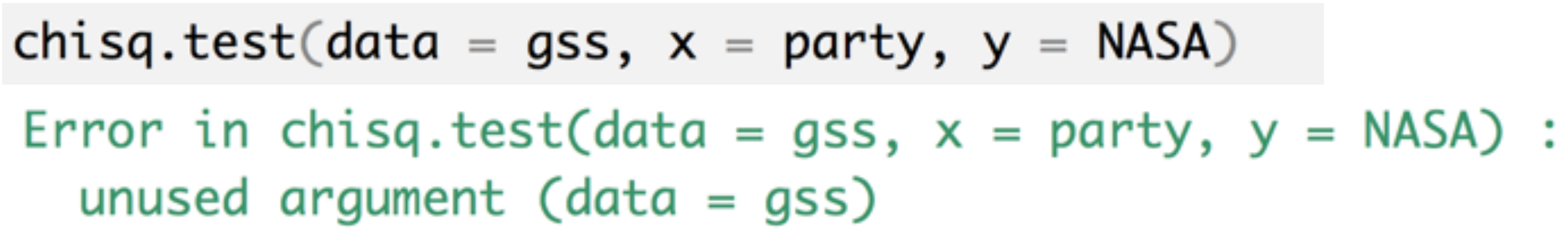
Optimistic effort I



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

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

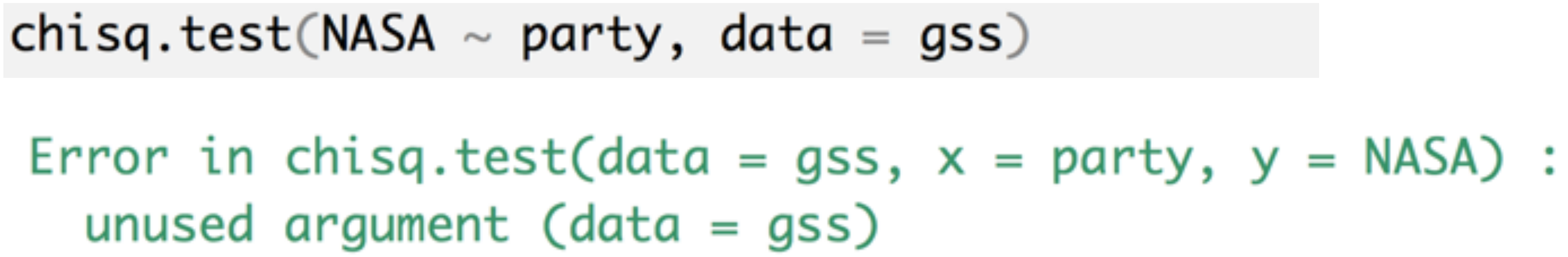
Optimistic effort I



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

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

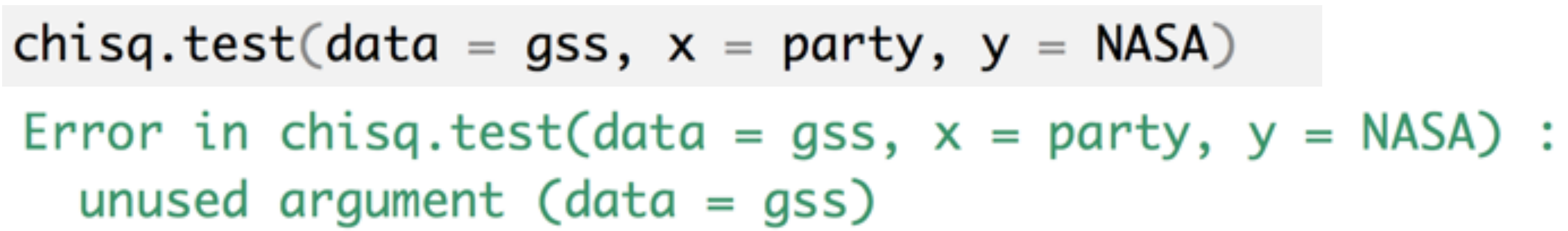
. . . optimistic effort II



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

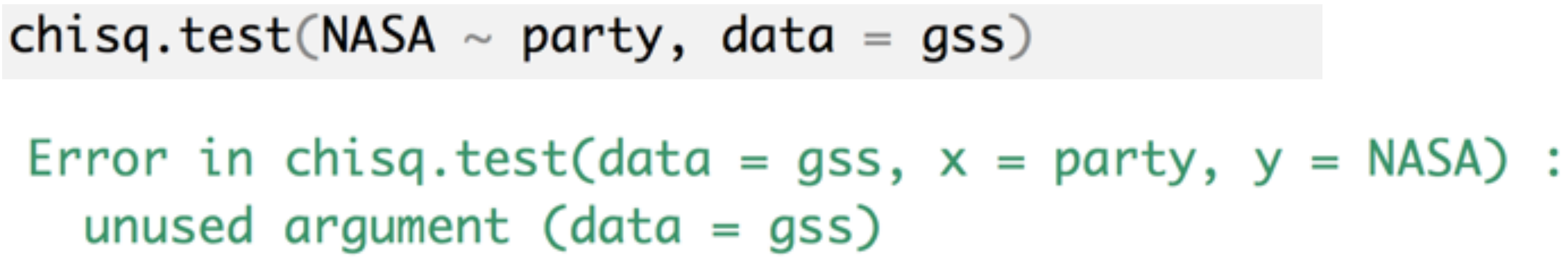
```
Error in chisq.test(data = gss, x = party, y = NASA) :
  unused argument (data = gss)
```

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)
```

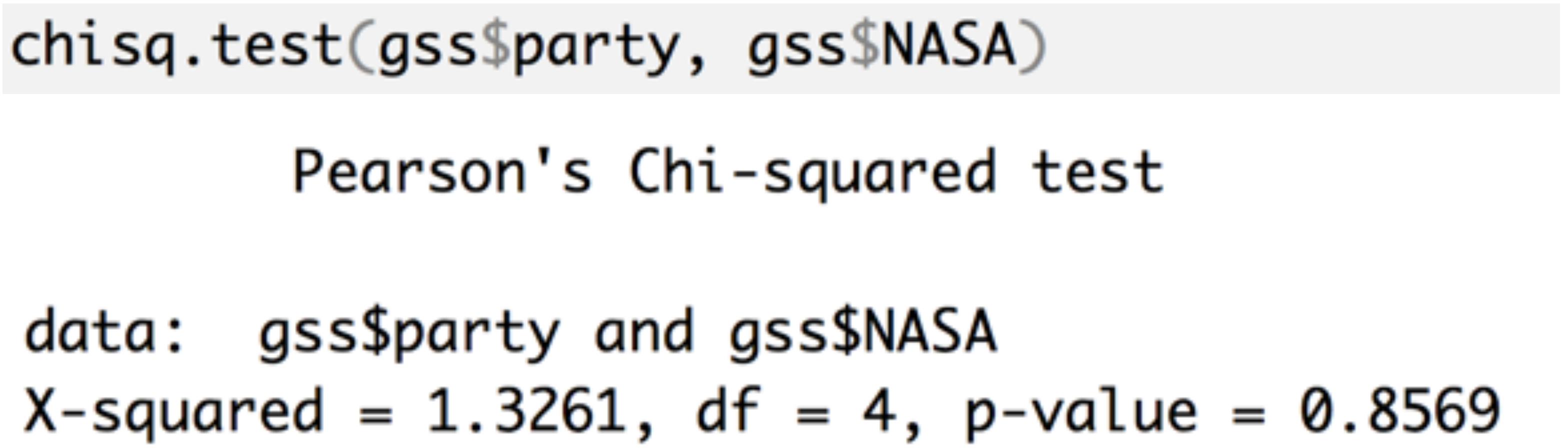
. . . optimistic effort II



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

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

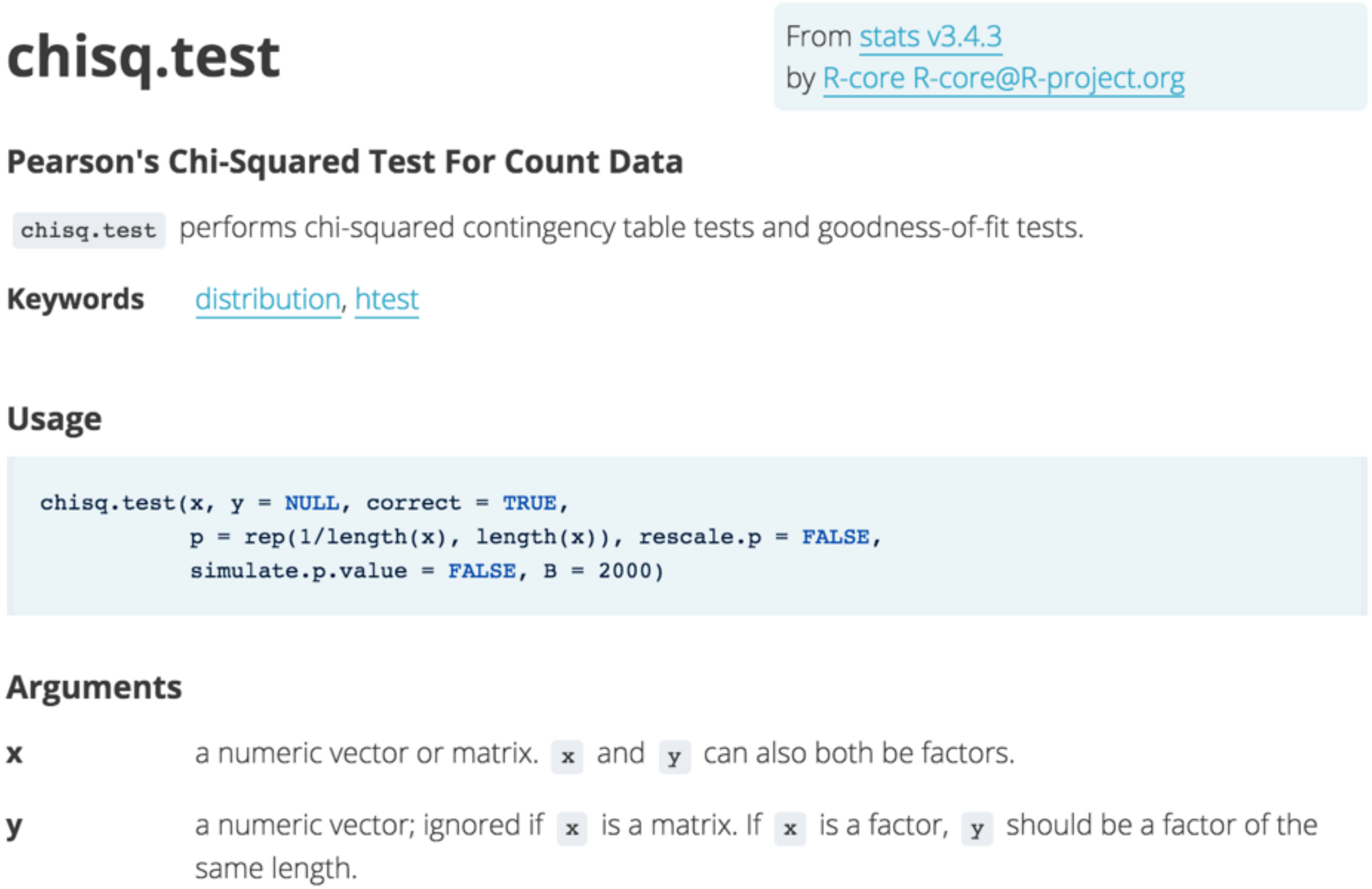
...after looking at the help file



```
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

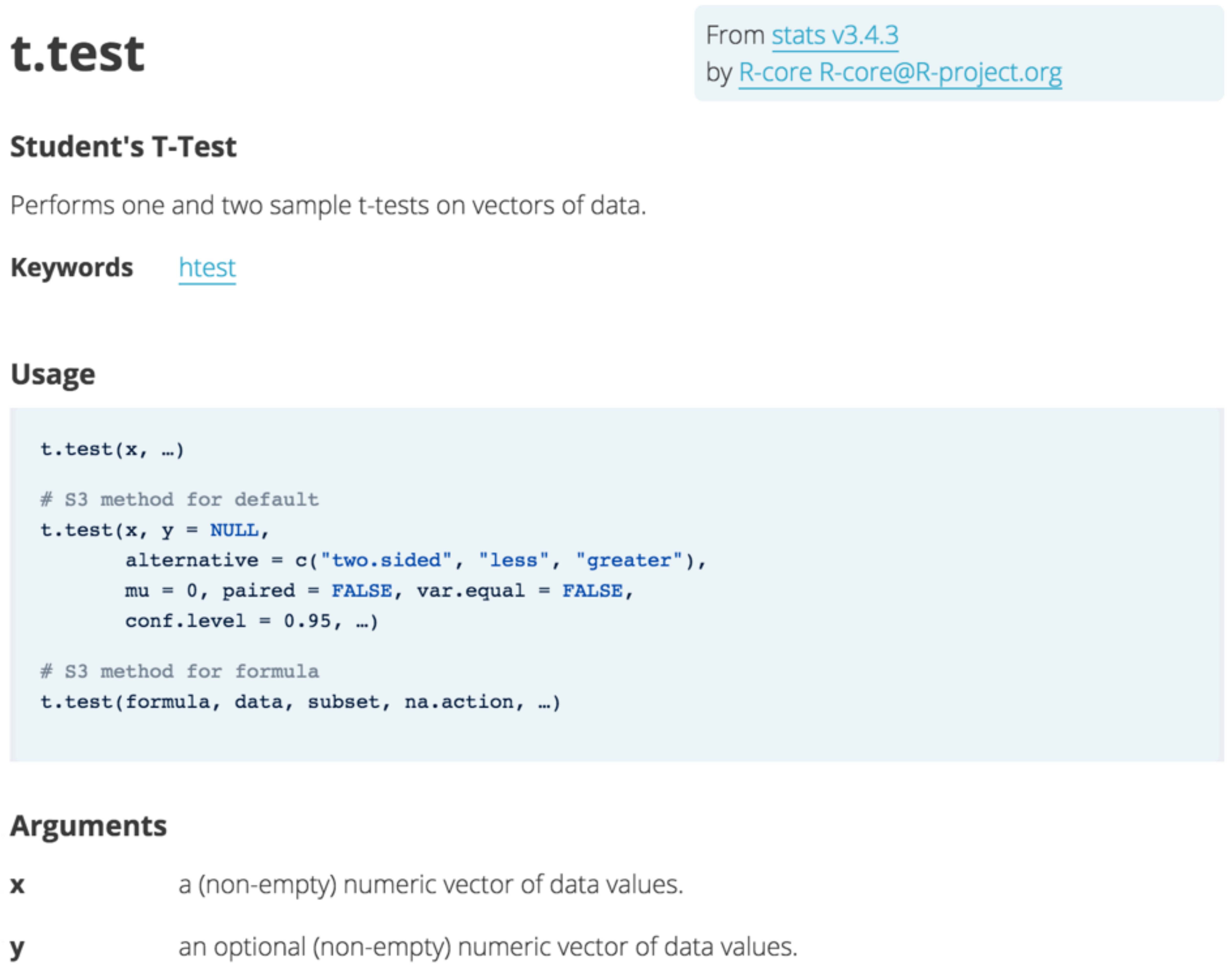
Usage

```
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.



t.test

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

Student's T-Test

Performs one and two sample t-tests on vectors of data.

Keywords htest

Usage

```
t.test(x, …)

# S3 method for default
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95, …)

# S3 method for formula
t.test(formula, data, subset, na.action, …)
```

Arguments

x a (non-empty) numeric vector of data values.

y an optional (non-empty) numeric vector of data values.

statistical inference

infer makes ~~p-values easier to compute.~~ tidy and transparent.

Two paradigms

Two paradigms

Mathematical Approximation

- Chi-squared
- Student t
- Normal

Two paradigms

Mathematical Approximation

- Chi-squared
- Student t
- Normal

Computational

- Permutation
- Bootstrap

Two paradigms

Mathematical Approximation

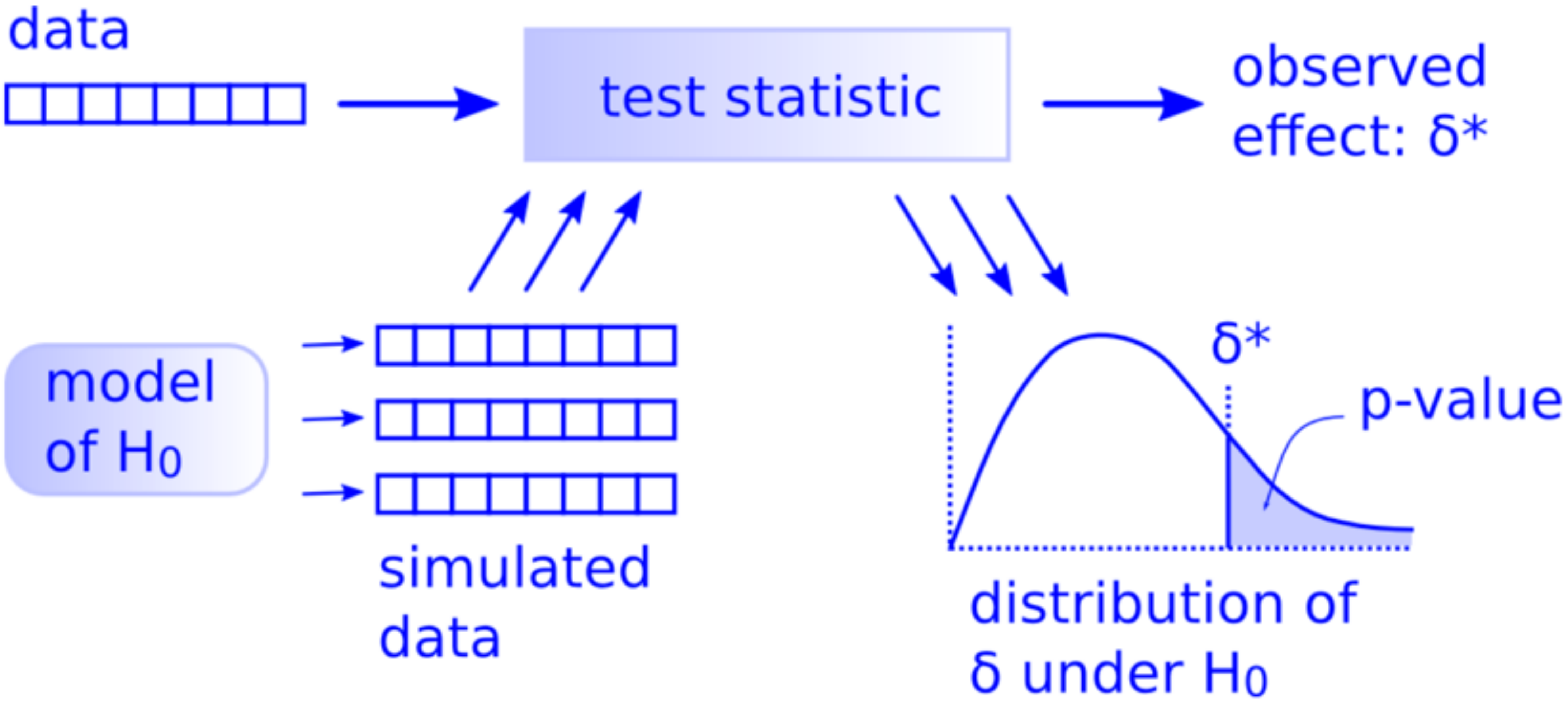
- Chi-squared
- Student t
- Normal

Computational

- Permutation
- Bootstrap

There is only one test

- Allen Downey

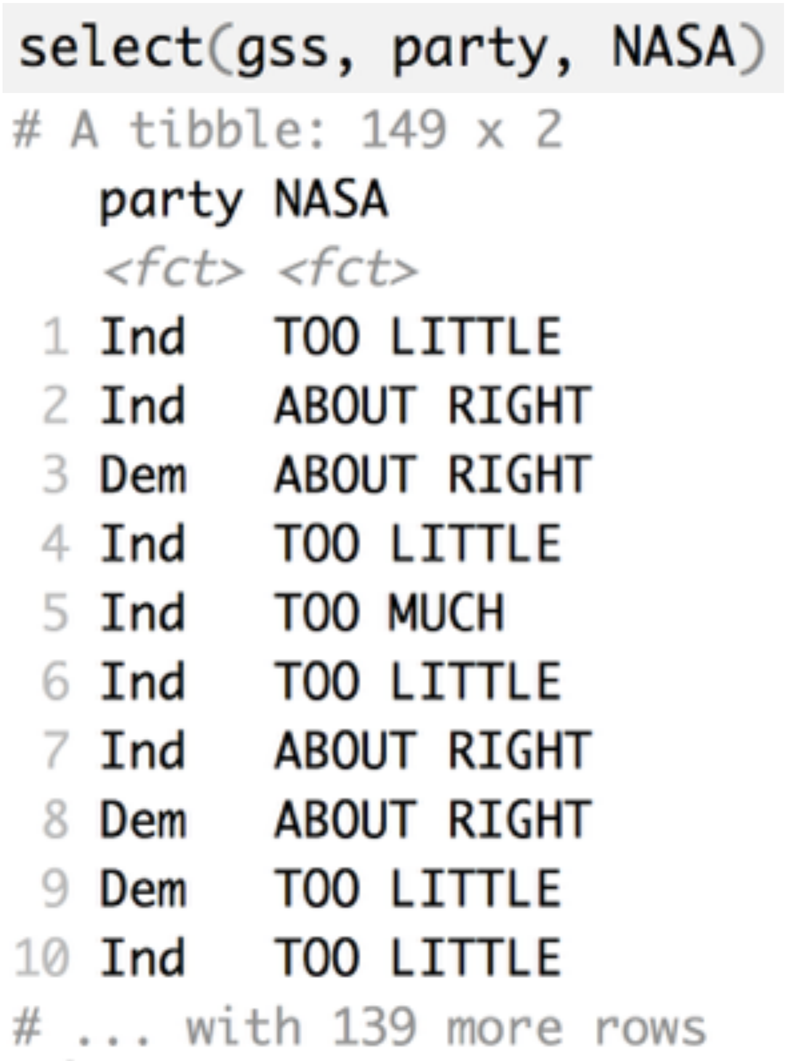


Simulation through Permutation

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.

Simulation through Permutation

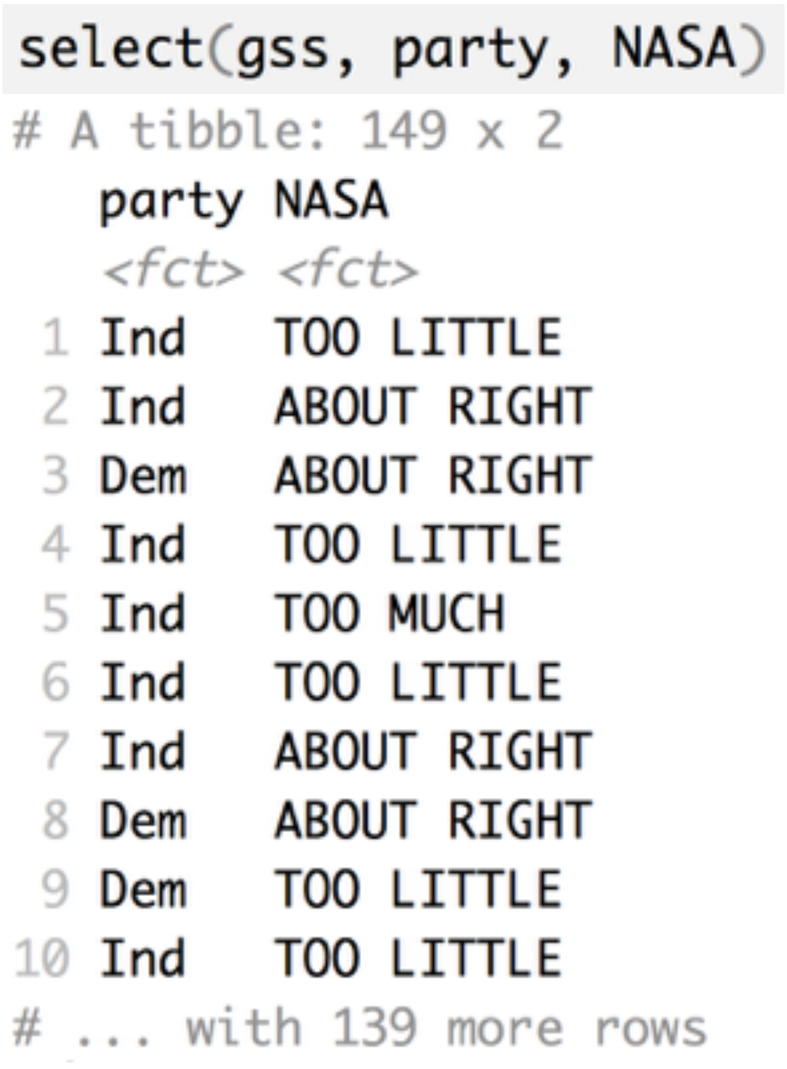
If we live in world where these variables are totally unrelated, the ties between variables are arbitrary, so they might just as well have been shuffled.



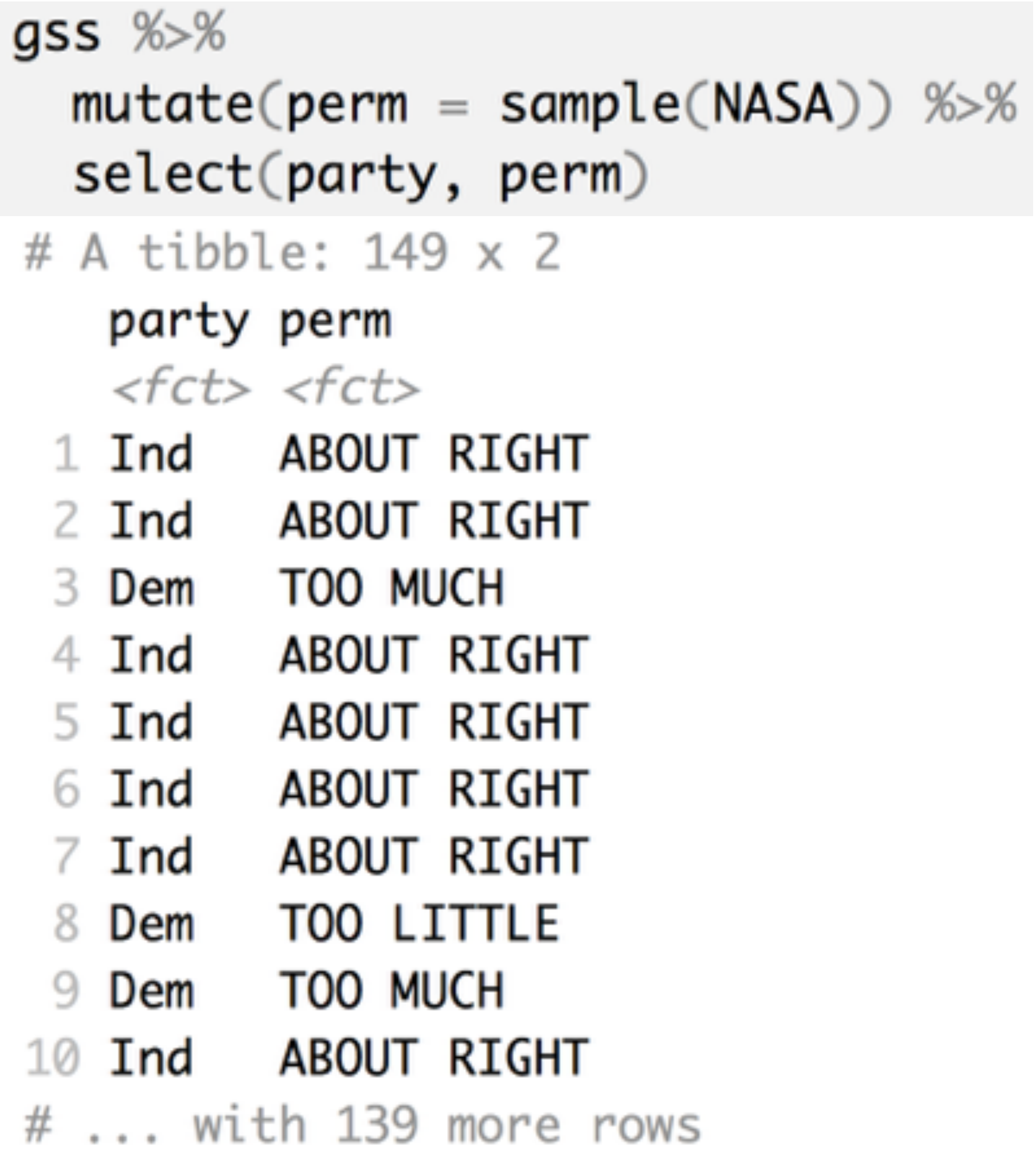
```
select(gss, party, NASA)
# A tibble: 149 x 2
   party NASA
   <fct> <fct>
 1 Ind   TOO LITTLE
 2 Ind   ABOUT RIGHT
 3 Dem   ABOUT RIGHT
 4 Ind   TOO LITTLE
 5 Ind   TOO MUCH
 6 Ind   TOO LITTLE
 7 Ind   ABOUT RIGHT
 8 Dem   ABOUT RIGHT
 9 Dem   TOO LITTLE
10 Ind   TOO LITTLE
# ... with 139 more rows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

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.



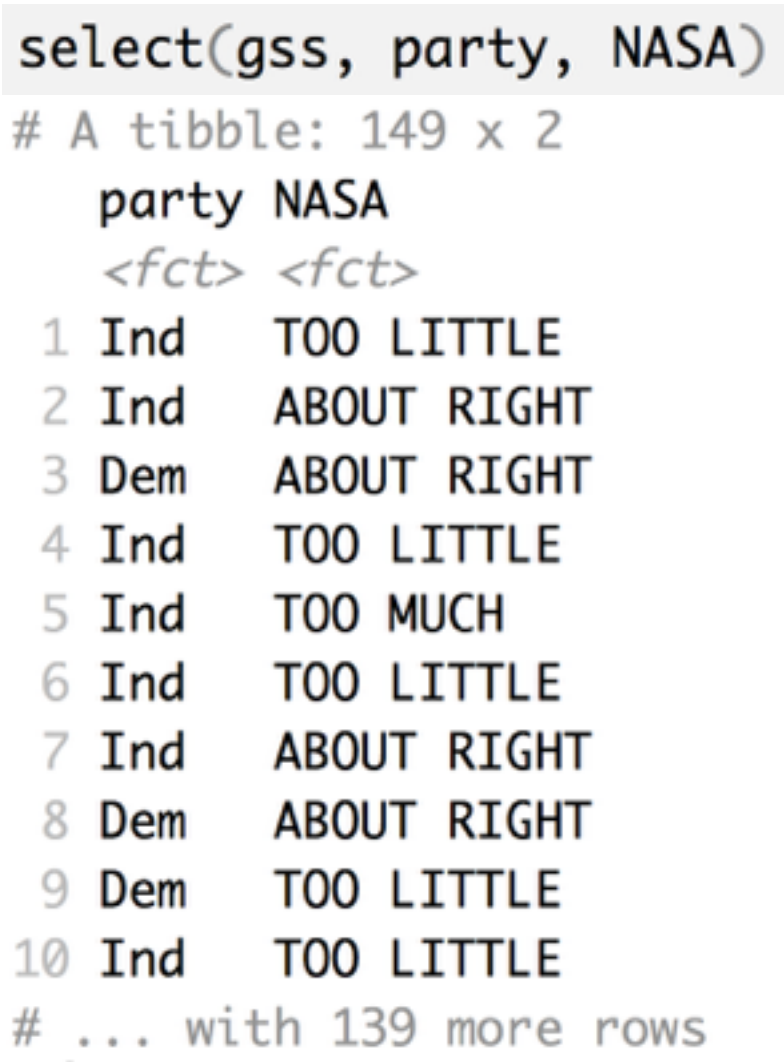
```
select(gss, party, NASA)
# A tibble: 149 x 2
   party NASA
   <fct> <fct>
 1 Ind   TOO LITTLE
 2 Ind   ABOUT RIGHT
 3 Dem   ABOUT RIGHT
 4 Ind   TOO LITTLE
 5 Ind   TOO MUCH
 6 Ind   TOO LITTLE
 7 Ind   ABOUT RIGHT
 8 Dem   ABOUT RIGHT
 9 Dem   TOO LITTLE
10 Ind   TOO LITTLE
# ... with 139 more rows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



```
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

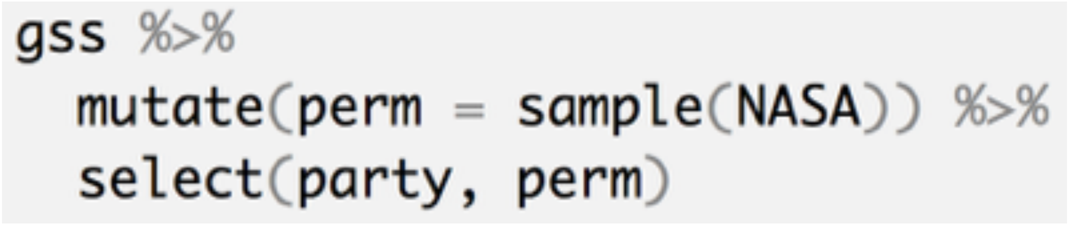
If we live in world where these variables are totally unrelated, the ties between variables are arbitrary, so they might just as well have been shuffled.



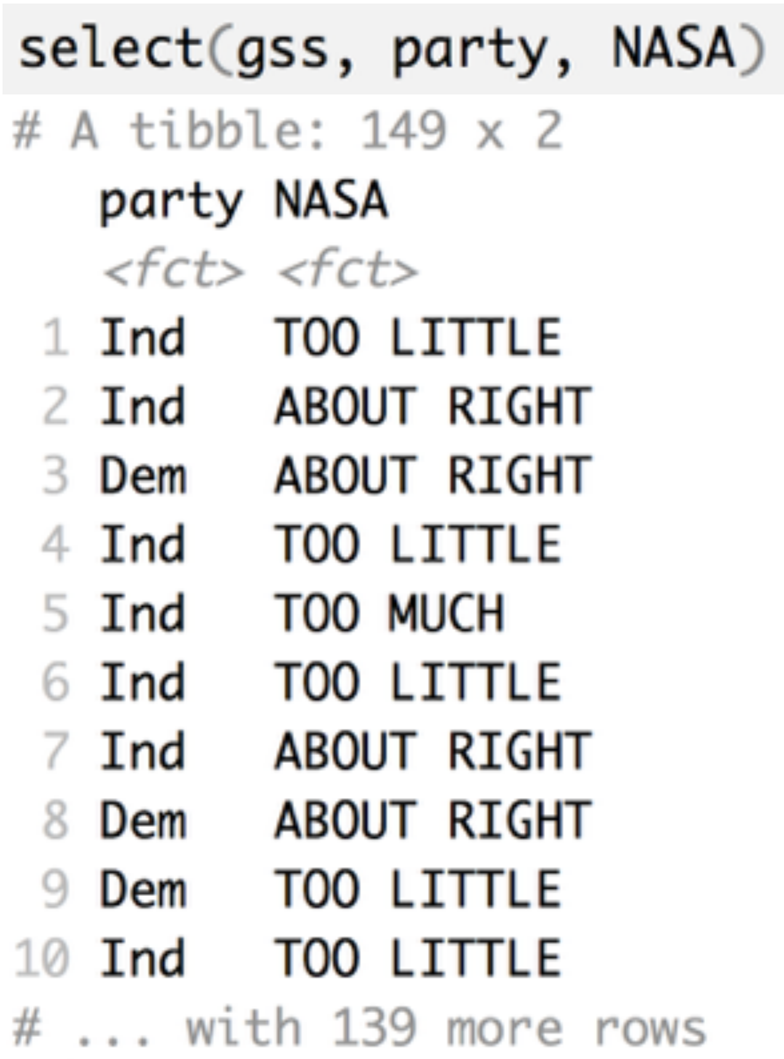
```
select(gss, party, NASA)
# A tibble: 149 x 2
   party NASA
   <fct> <fct>
 1 Ind   TOO LITTLE
 2 Ind   ABOUT RIGHT
 3 Dem   ABOUT RIGHT
 4 Ind   TOO LITTLE
 5 Ind   TOO MUCH
 6 Ind   TOO LITTLE
 7 Ind   ABOUT RIGHT
 8 Dem   ABOUT RIGHT
 9 Dem   TOO LITTLE
10 Ind   TOO LITTLE
# ... with 139 more rows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

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.



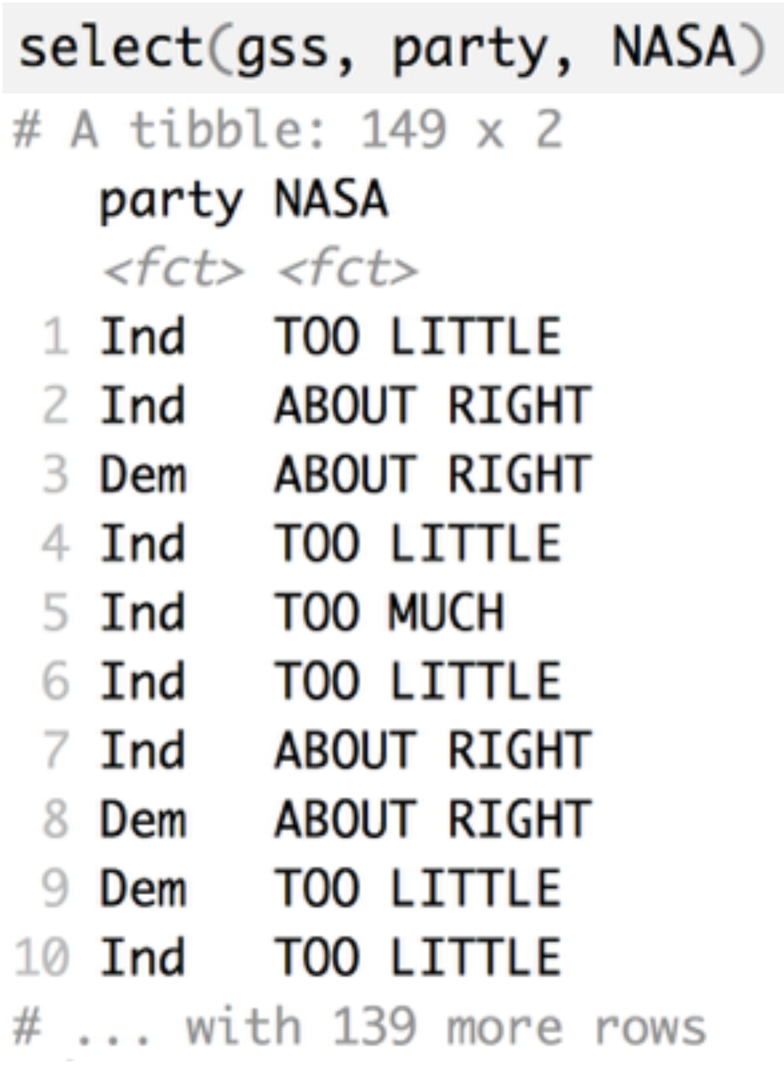
```
gss %>%
  mutate(perm = sample(NASA)) %>%
  select(party, perm)
```



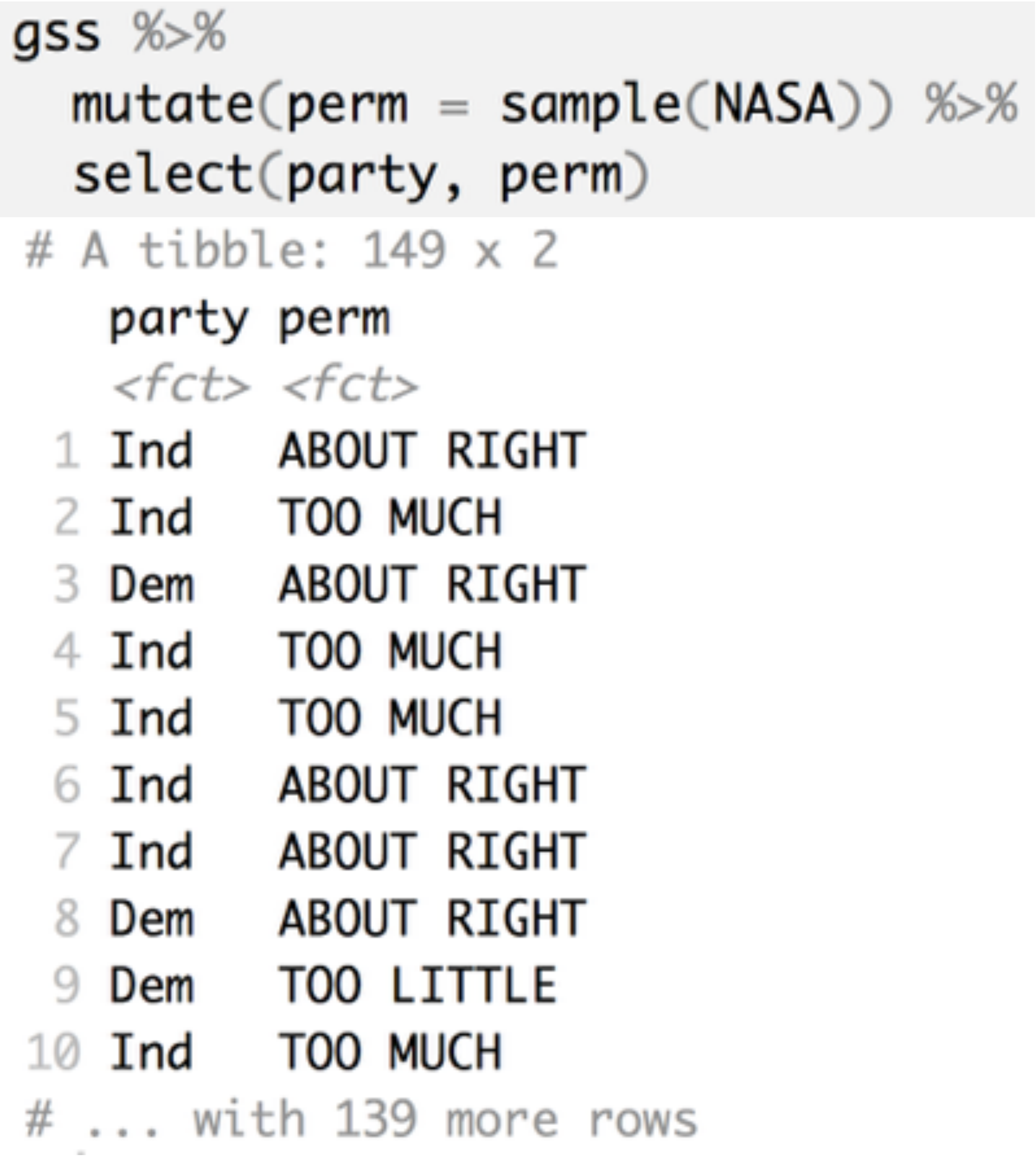
```
select(gss, party, NASA)
# A tibble: 149 x 2
   party NASA
   <fct> <fct>
 1 Ind   TOO LITTLE
 2 Ind   ABOUT RIGHT
 3 Dem   ABOUT RIGHT
 4 Ind   TOO LITTLE
 5 Ind   TOO MUCH
 6 Ind   TOO LITTLE
 7 Ind   ABOUT RIGHT
 8 Dem   ABOUT RIGHT
 9 Dem   TOO LITTLE
10 Ind   TOO LITTLE
# ... with 139 more rows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

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
```



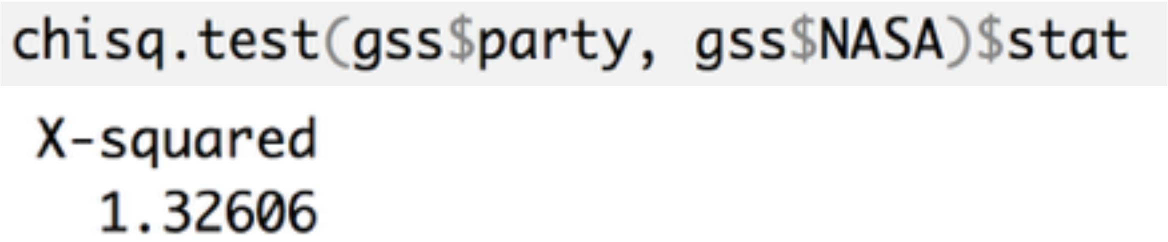
```
gss %>%
  mutate(perm = sample(NASA)) %>%
  select(party, perm)
# A tibble: 149 x 2
   party perm
   <fct> <fct>
 1 Ind   ABOUT RIGHT
 2 Ind   TOO MUCH
 3 Dem   ABOUT RIGHT
 4 Ind   TOO MUCH
 5 Ind   TOO MUCH
 6 Ind   ABOUT RIGHT
 7 Ind   ABOUT RIGHT
 8 Dem   ABOUT RIGHT
 9 Dem   TOO LITTLE
10 Ind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

Test statistic

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

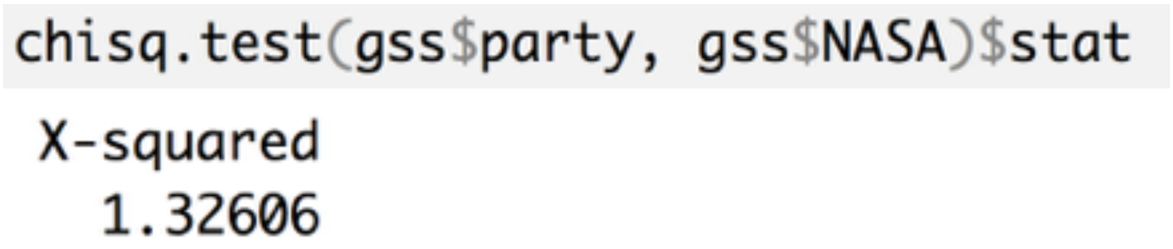


```
chisq.test(gss$party, gss$NASA)$stat
```

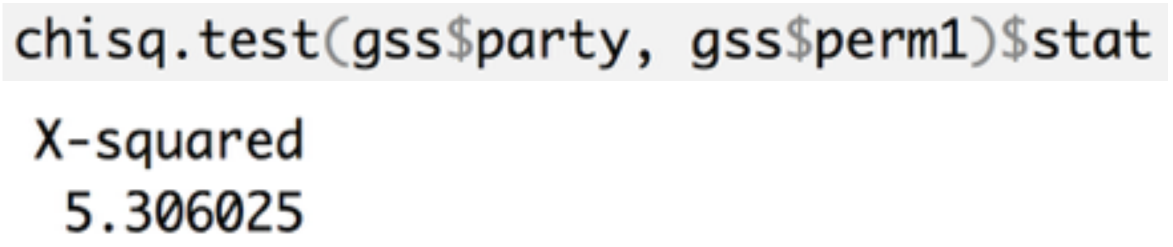
```
 X-squared 
   1.32606 
```

Test statistic

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



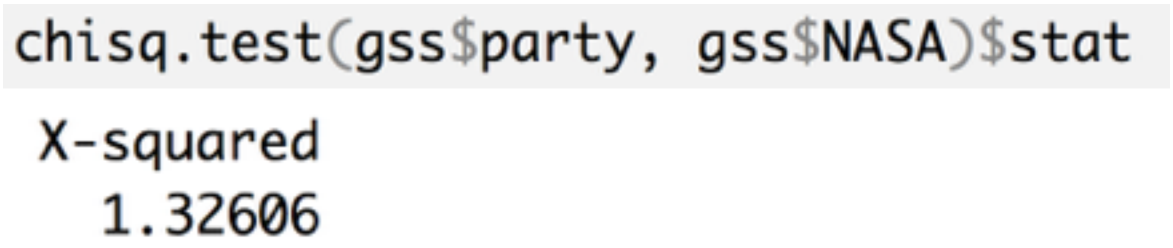
```
chisq.test(gss$party, gss$NASA)$stat
 X-squared
   1.32606
```



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

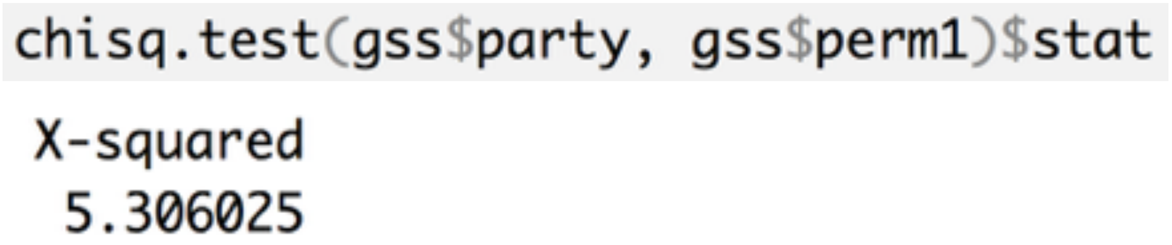
Test statistic

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



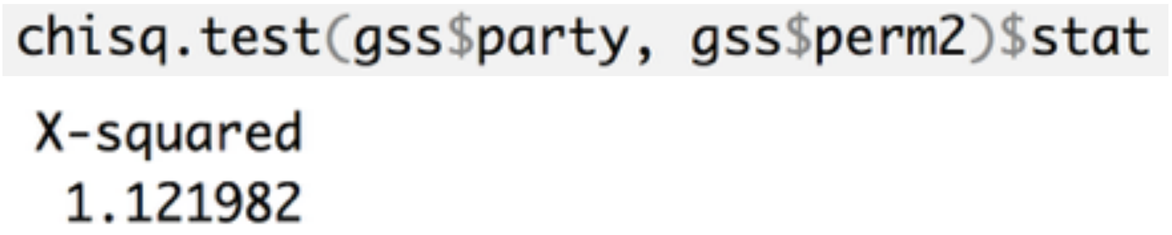
```
chisq.test(gss$party, gss$NASA)$stat
```

```
X-squared
  1.32606
```



```
chisq.test(gss$party, gss$perm1)$stat
```

```
X-squared
 5.306025
```

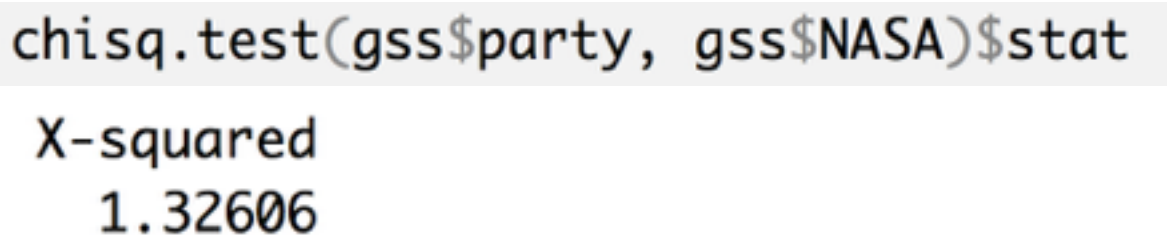


```
chisq.test(gss$party, gss$perm2)$stat
```

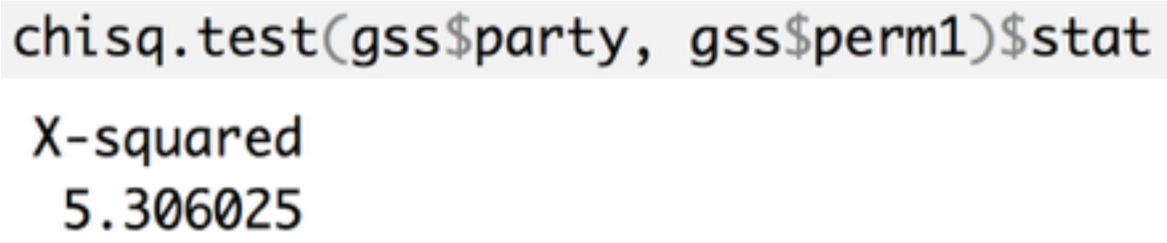
```
X-squared
 1.121982
```

Test statistic

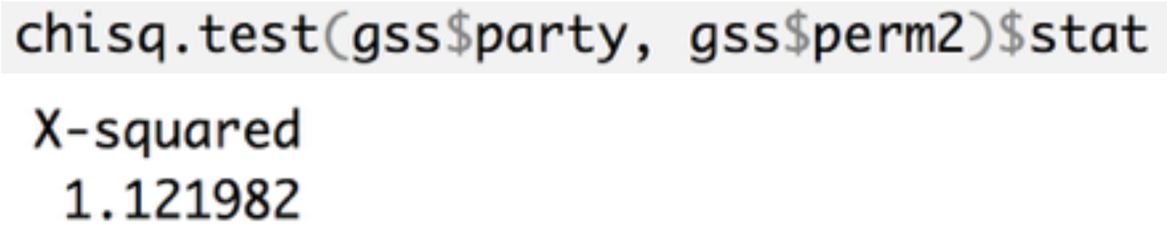
Chi-squared statistic: a measure of the difference between your data and what you would expect if the null hypothesis were true.



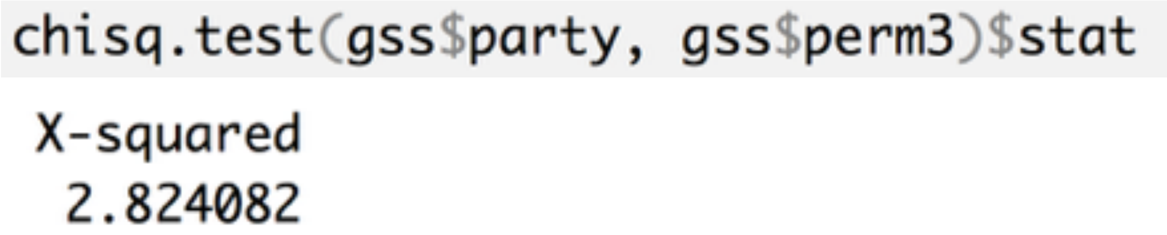
```
chisq.test(gss$party, gss$NASA)$stat
 X-squared
   1.32606
```



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



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

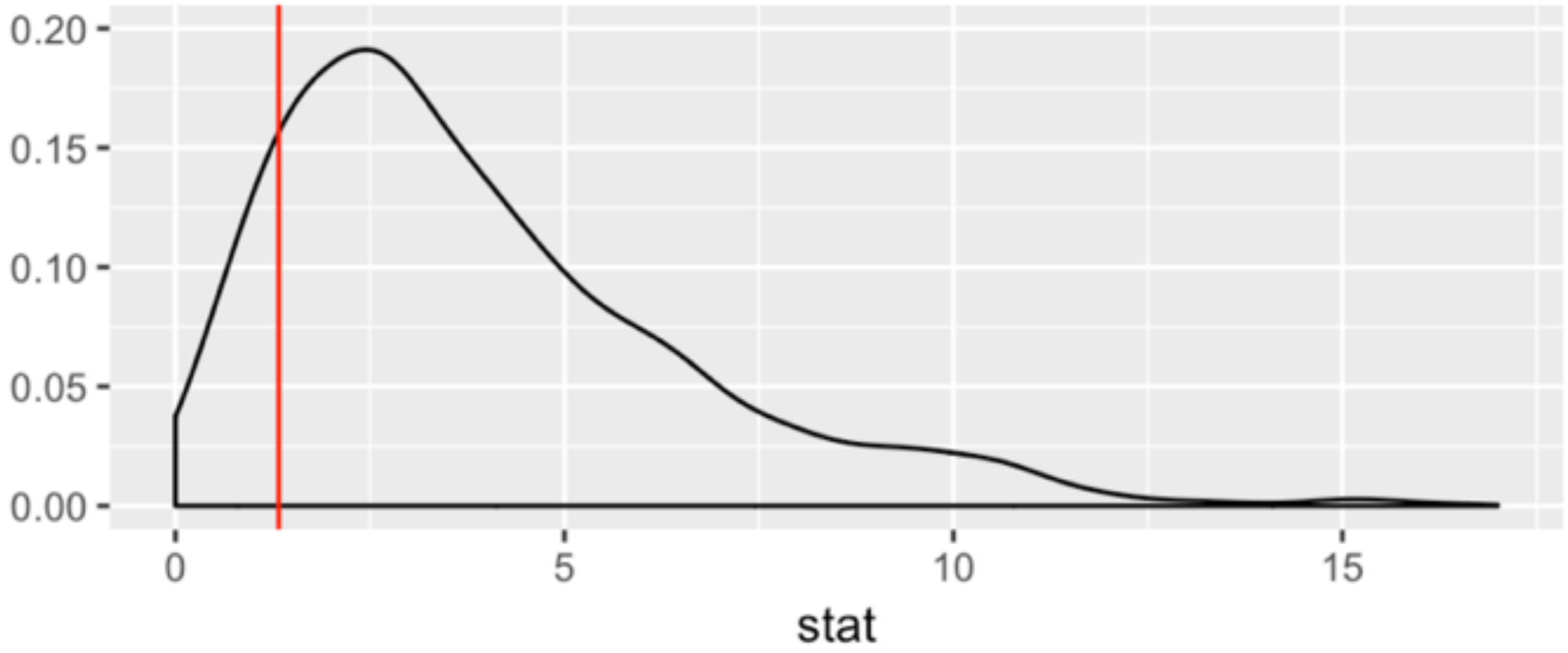


```
chisq.test(gss$party, gss$perm3)$stat
 X-squared
  2.824082
```

Distribution of statistic

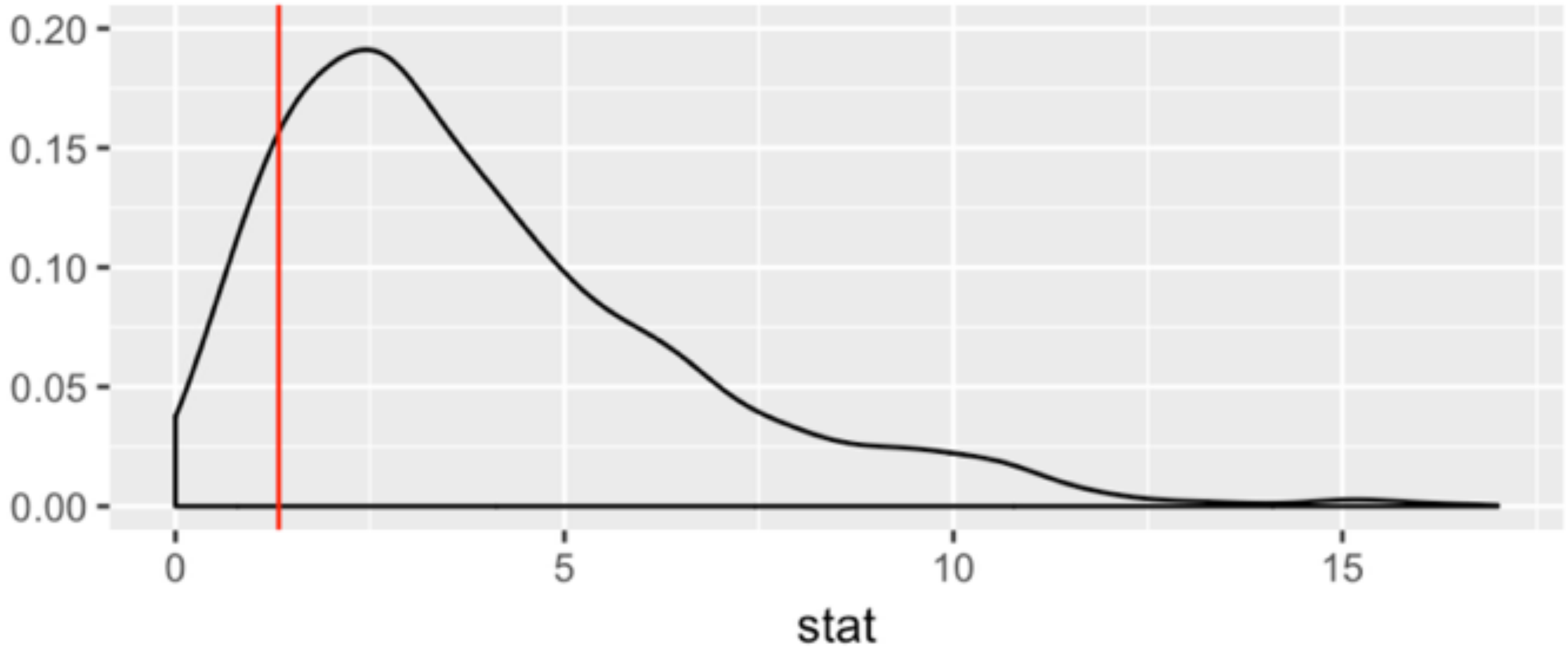
Distribution of statistic

via permutation



Distribution of statistic

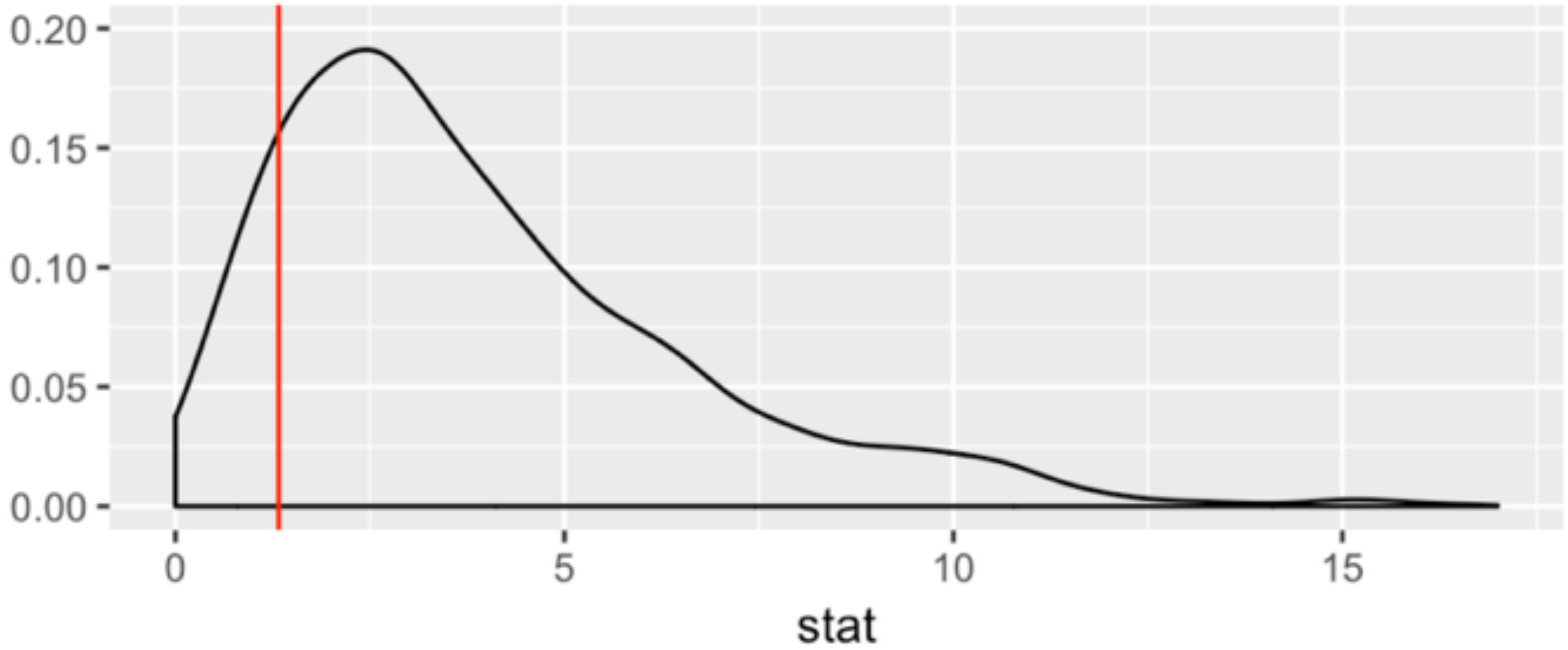
via permutation



Question: Is funding for space exploration a partisan issue?

Distribution of statistic

via permutation

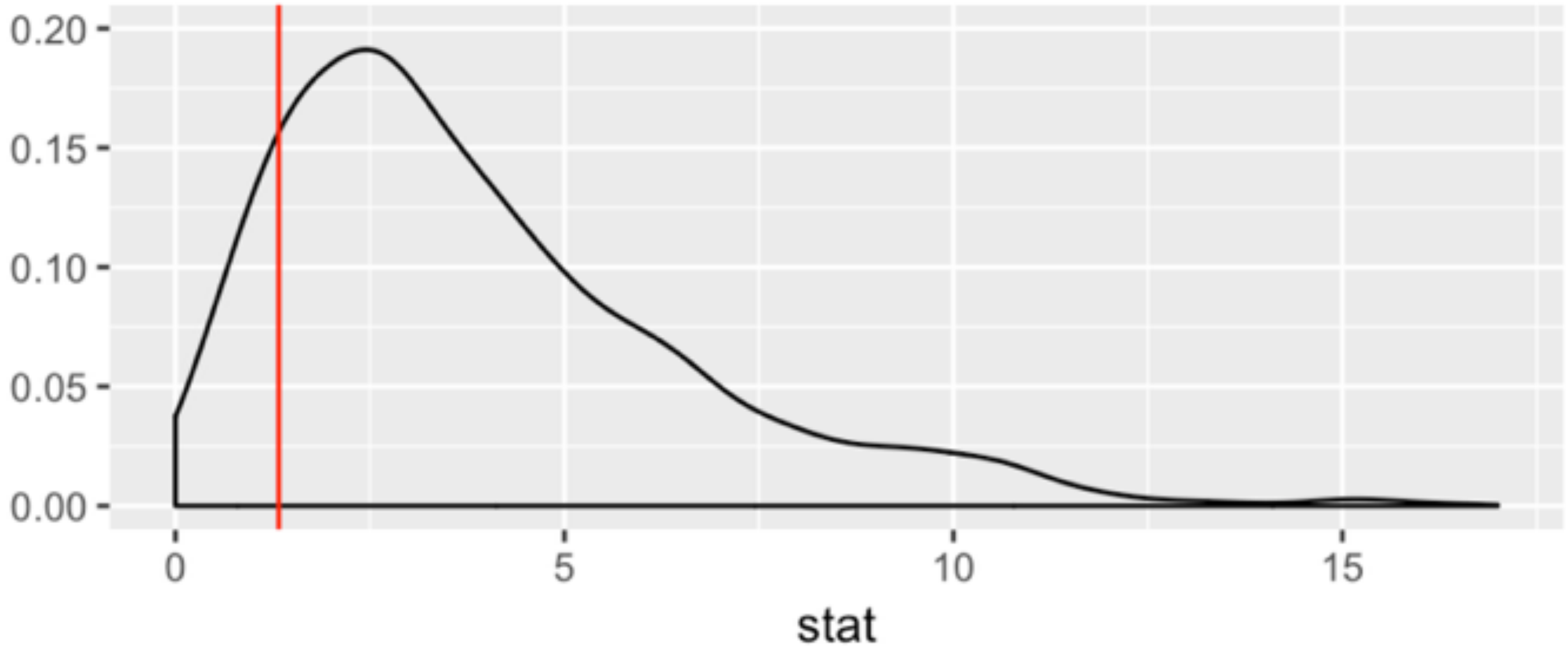


Question: Is funding for space exploration a partisan issue?

Answer: The GSS data is consistent with the model that there is no association between party and space funding.

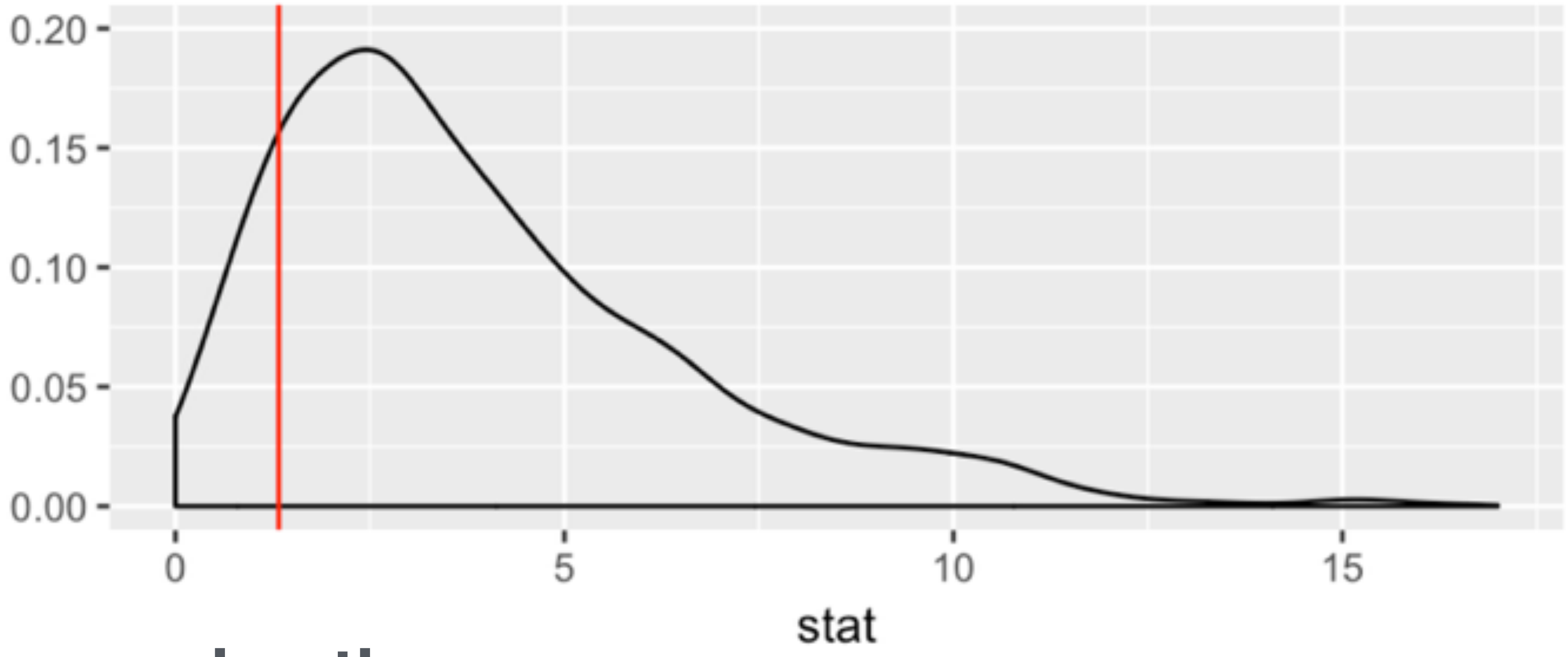
Distribution of statistic

via permutation

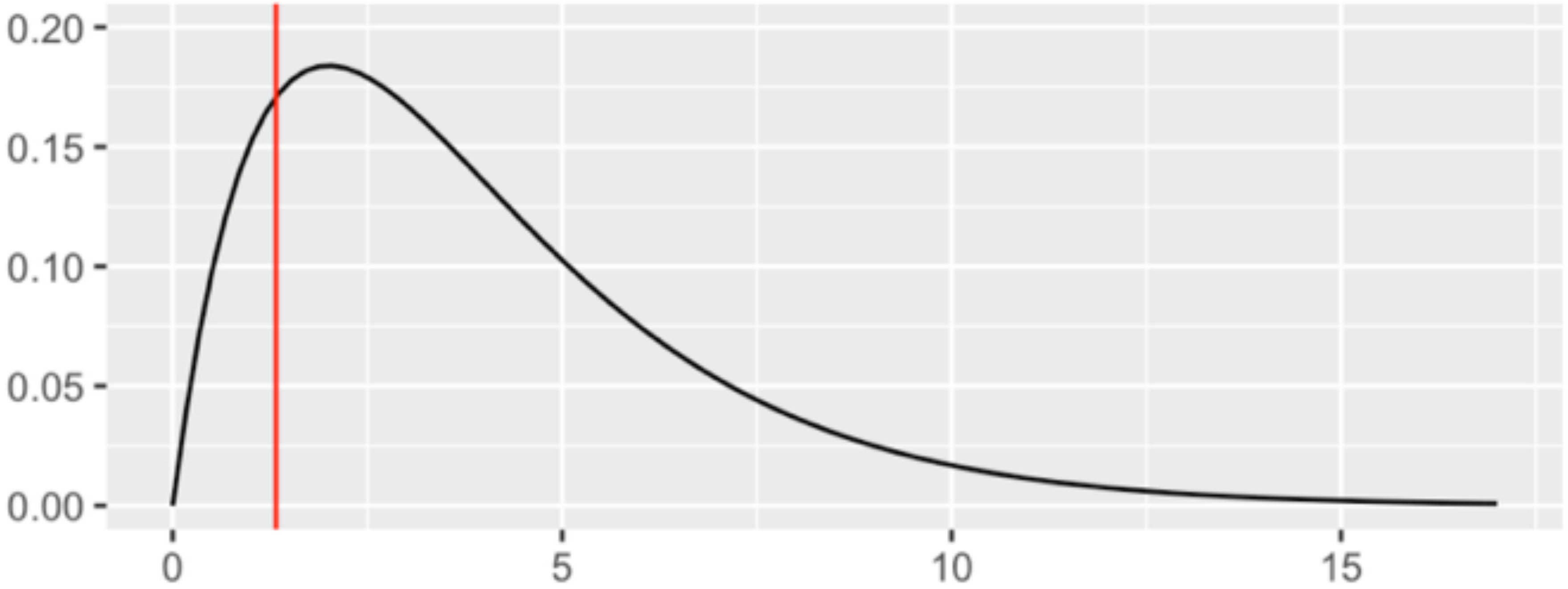


Distribution of statistic

via permutation



via approximation



infer makes statistical inference tidy and transparent.

infer makes statistical inference tidy and transparent.

- dataframe in, dataframe out

infer makes statistical inference tidy and transparent.

- dataframe in, dataframe out
- compose tests and intervals with pipes

infer makes statistical inference tidy and transparent.

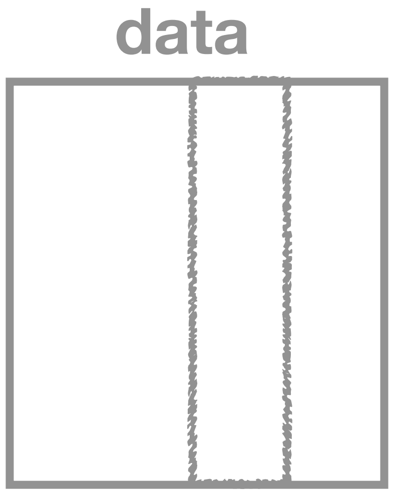
- dataframe in, dataframe out
- compose tests and intervals with pipes
- unite computational and approximation methods

infer makes statistical inference tidy and transparent.

- dataframe in, dataframe out
- compose tests and intervals with pipes
- unite computational and approximation methods
- reading an infer chain describes an inferential procedure

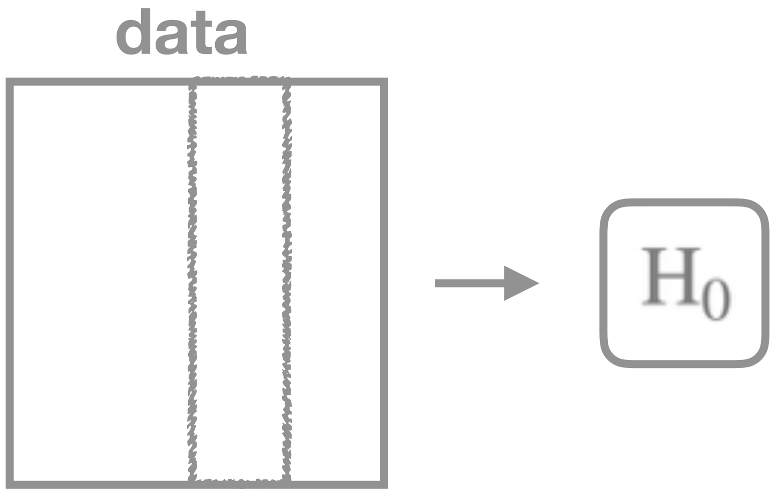
The **`infer`** verbs

The `infer` verbs



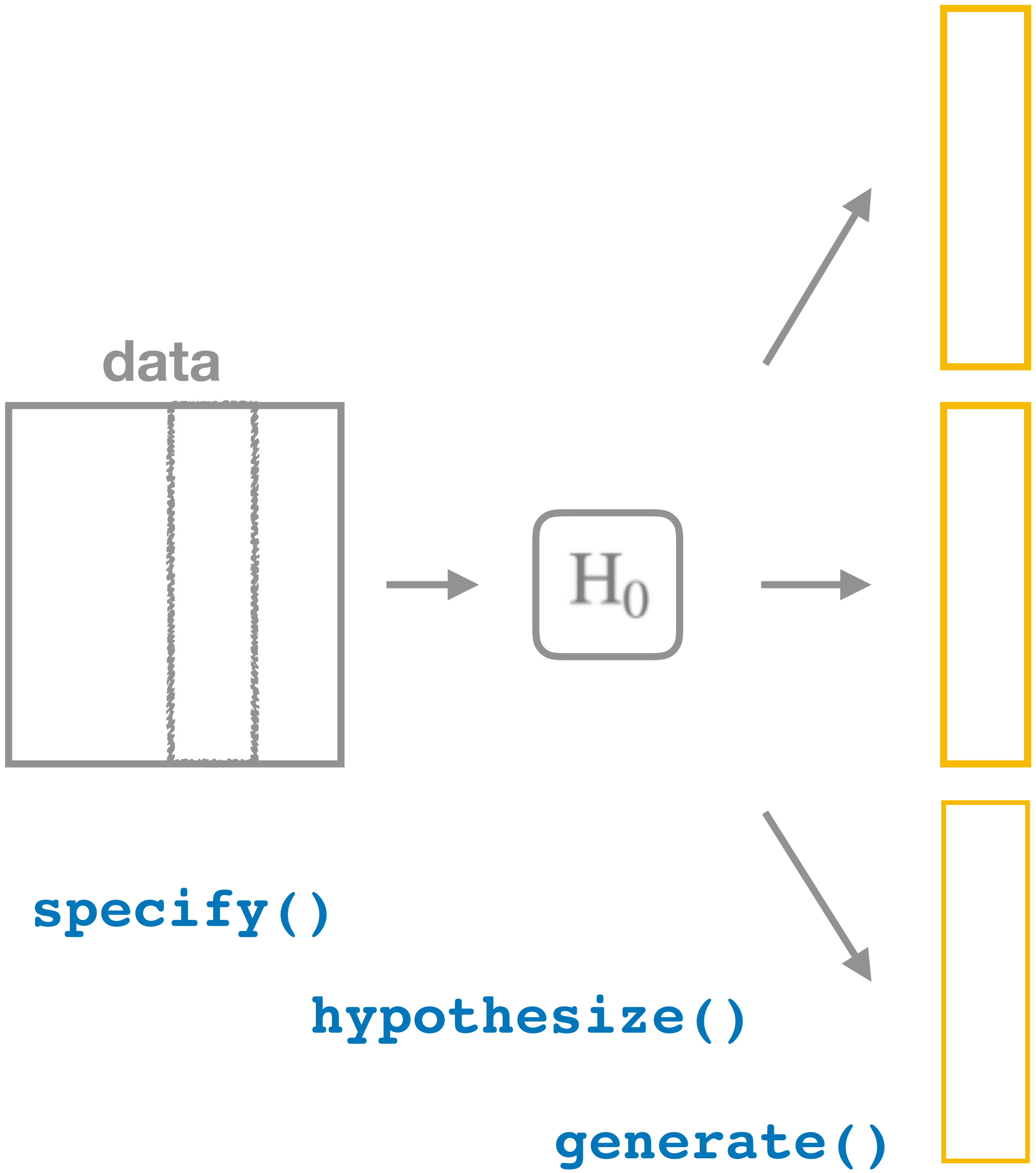
`specify()`

The `infer` verbs



```
specify()
          hypothesize()
```

The infer verbs

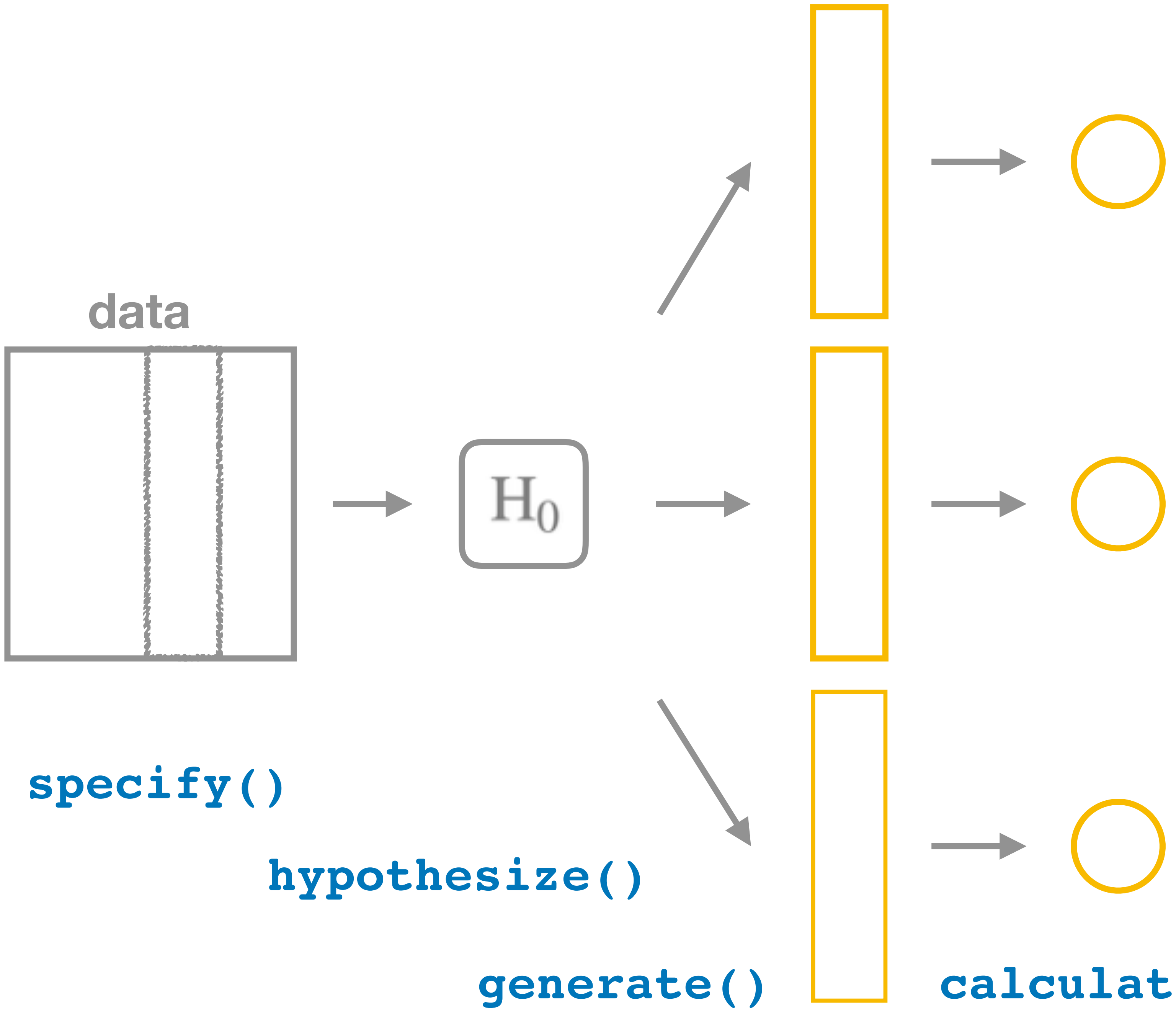


specify()

hypothesize()

generate()

The infer verbs



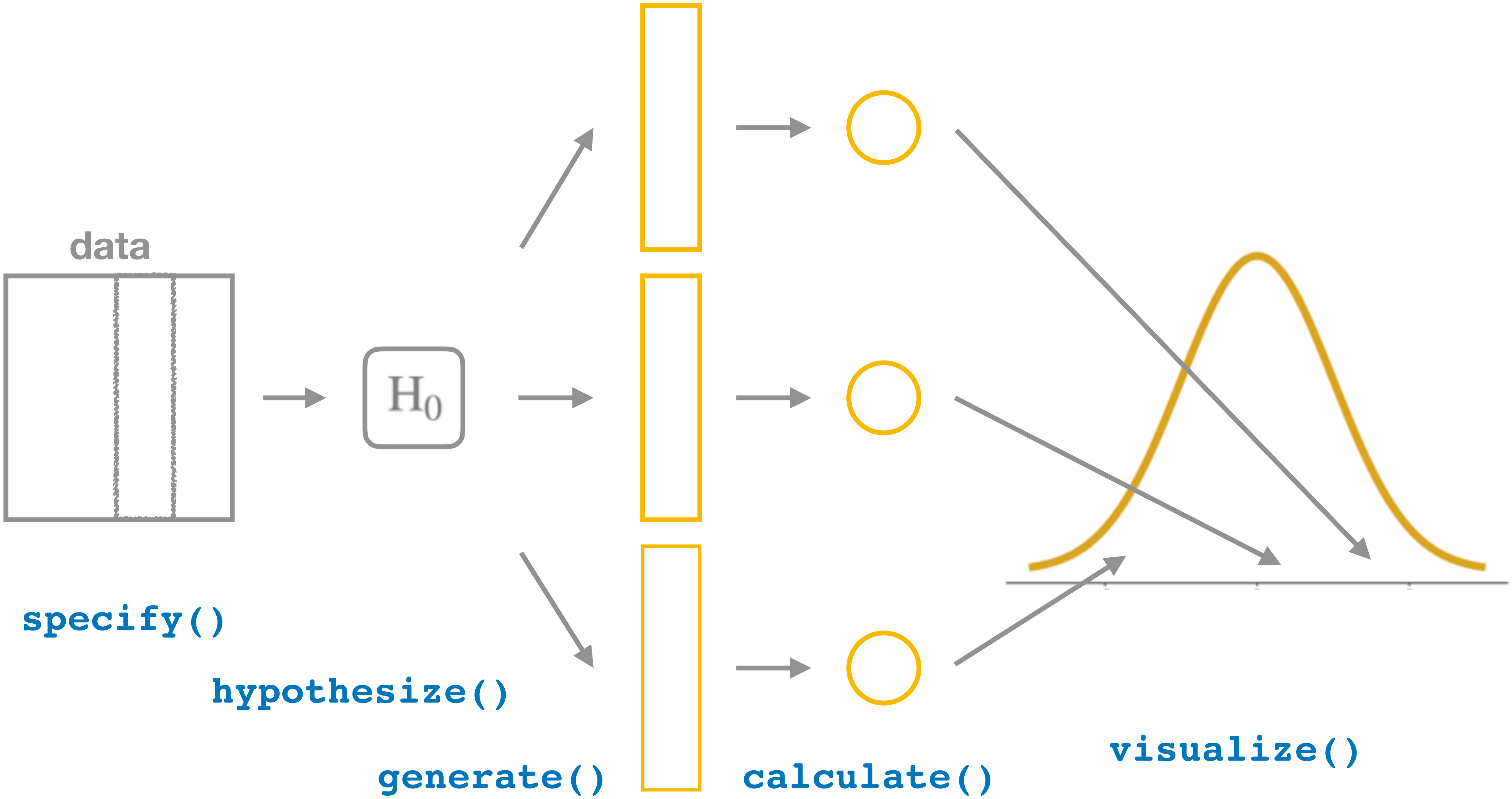
specify()

hypothesize()

generate()

calculate()

The `infer` verbs



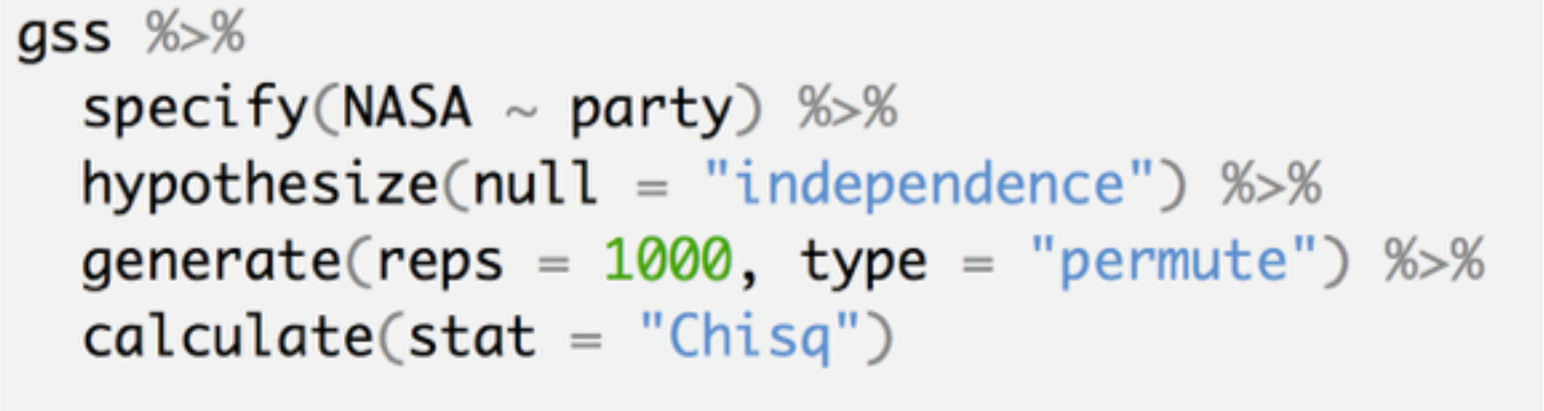
`specify()`

`hypothesize()`

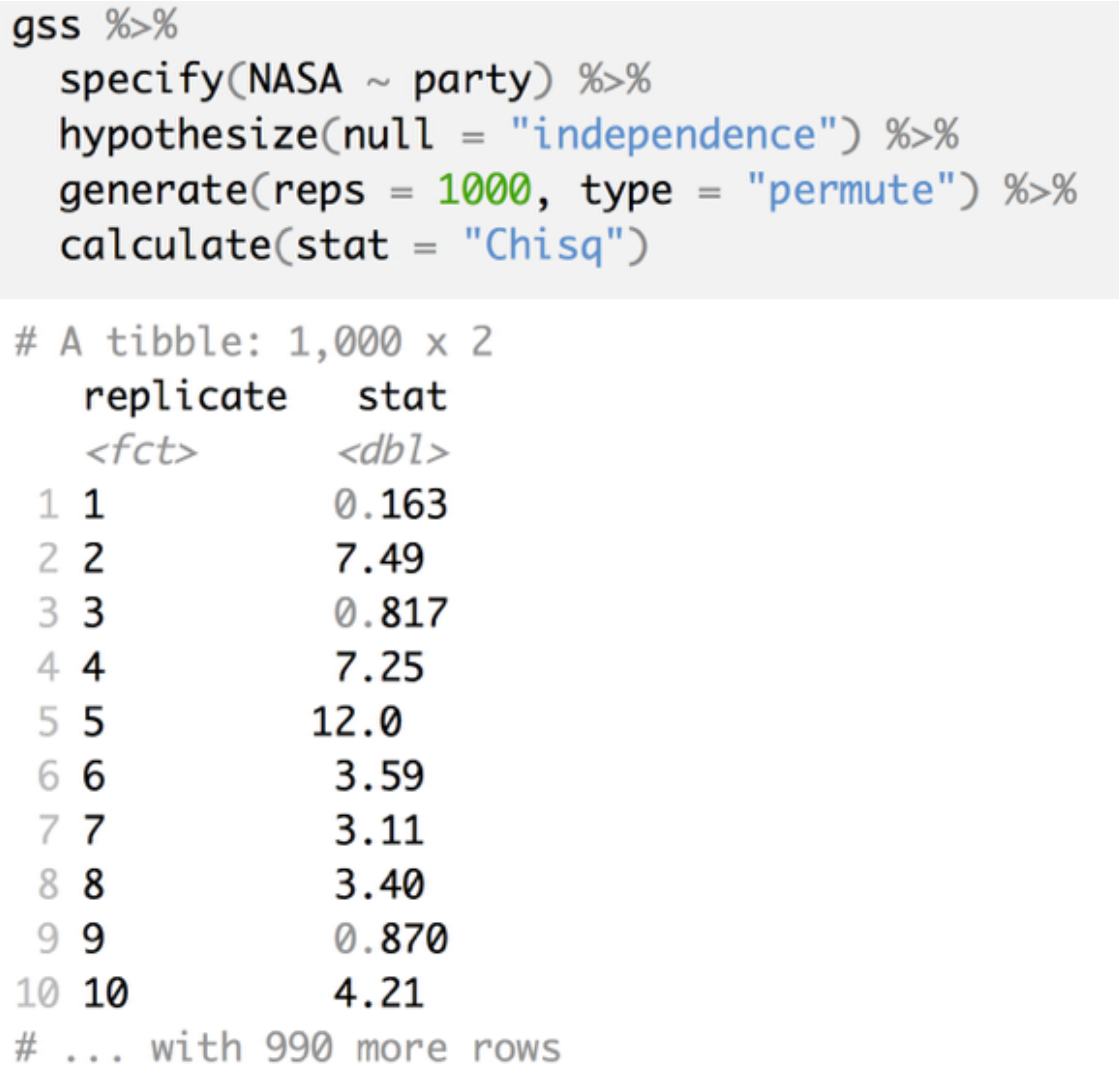
`generate()`

`calculate()`

`visualize()`

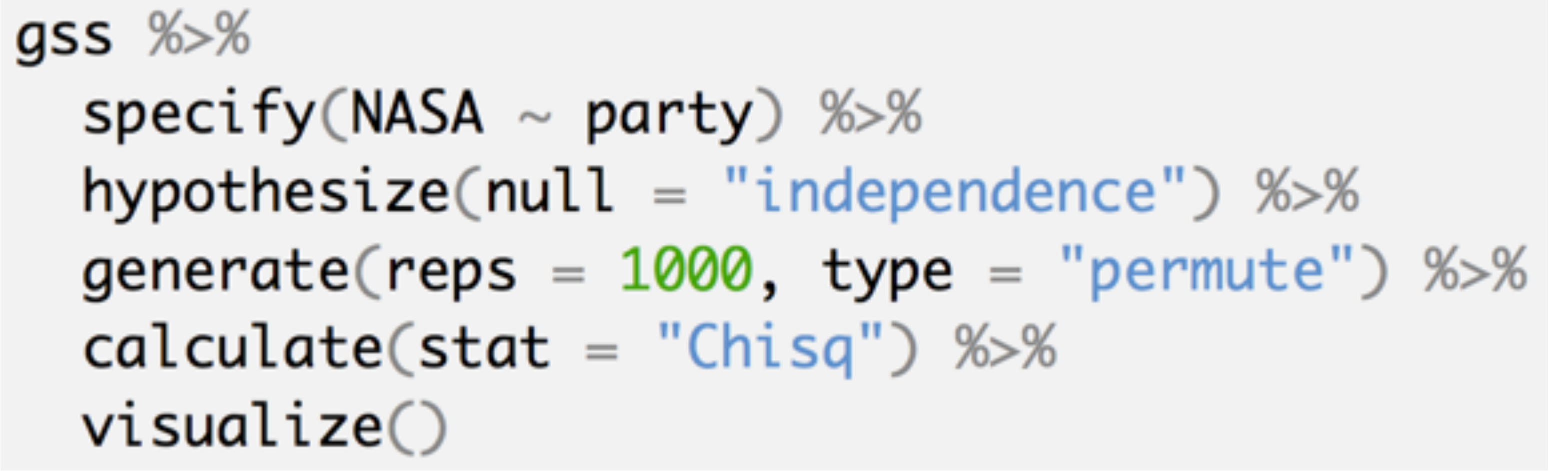


```r
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq")
```



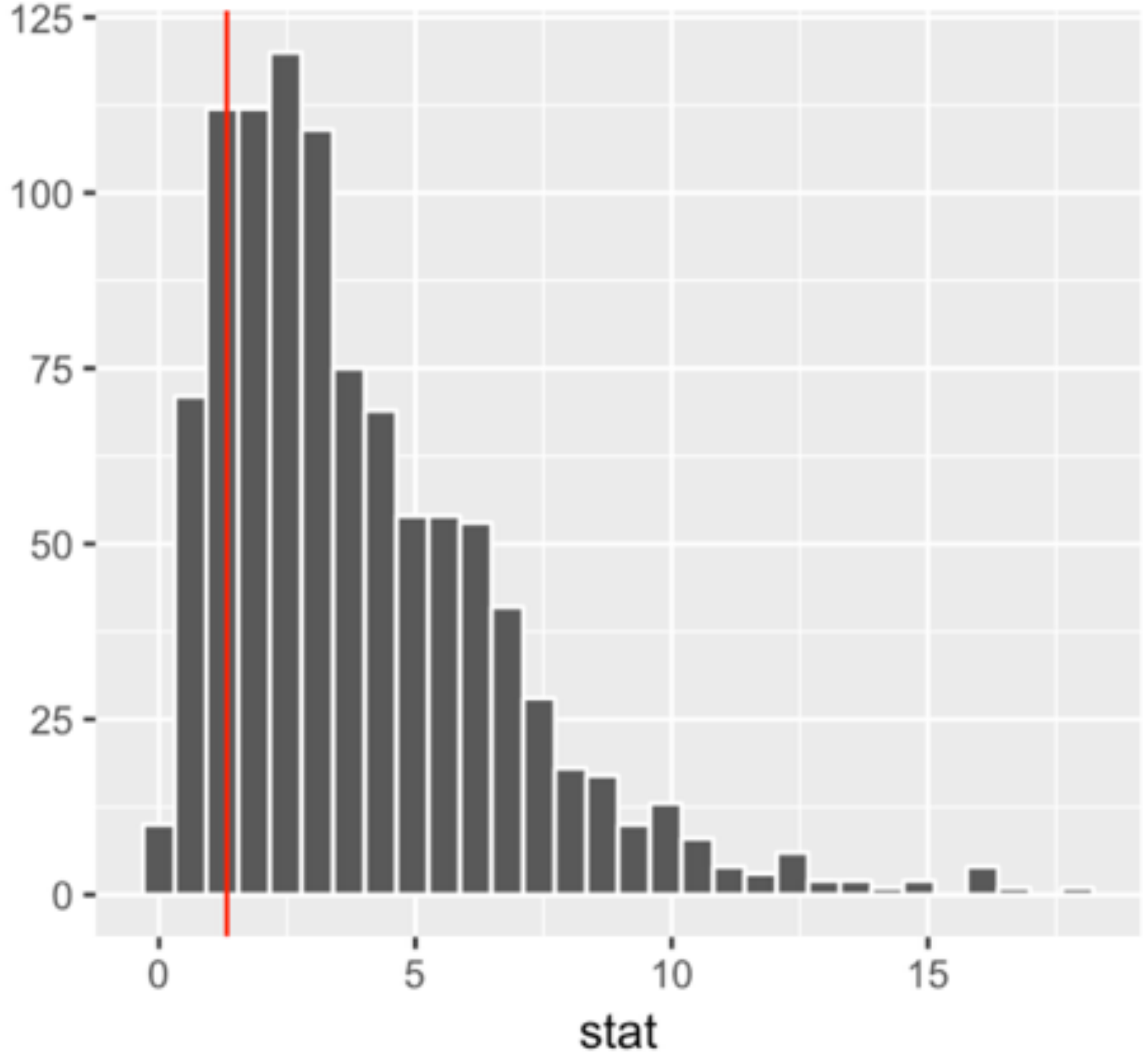
```
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq")
```

```
# 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
```

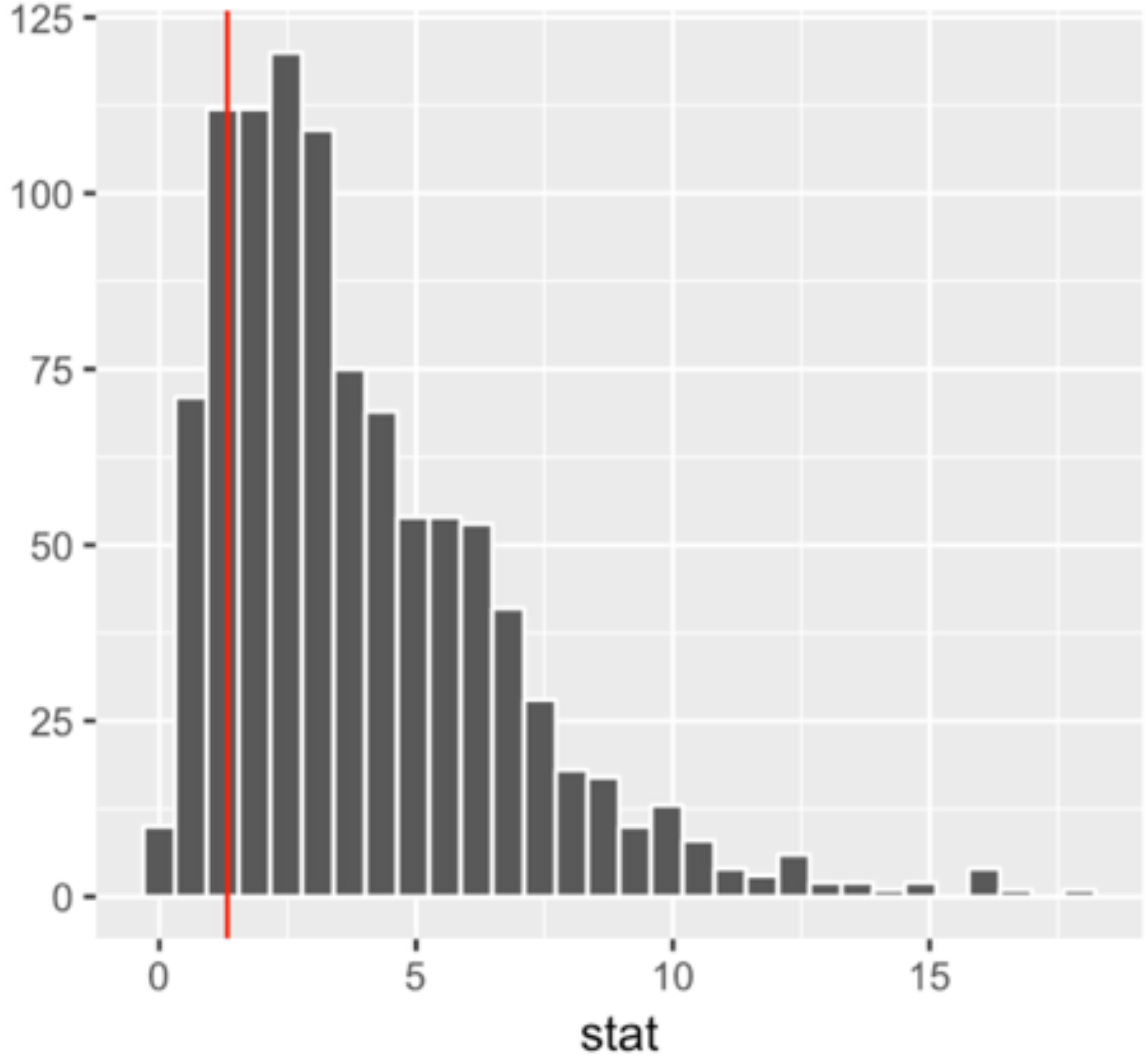


```
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq") %>%
  visualize()
```

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

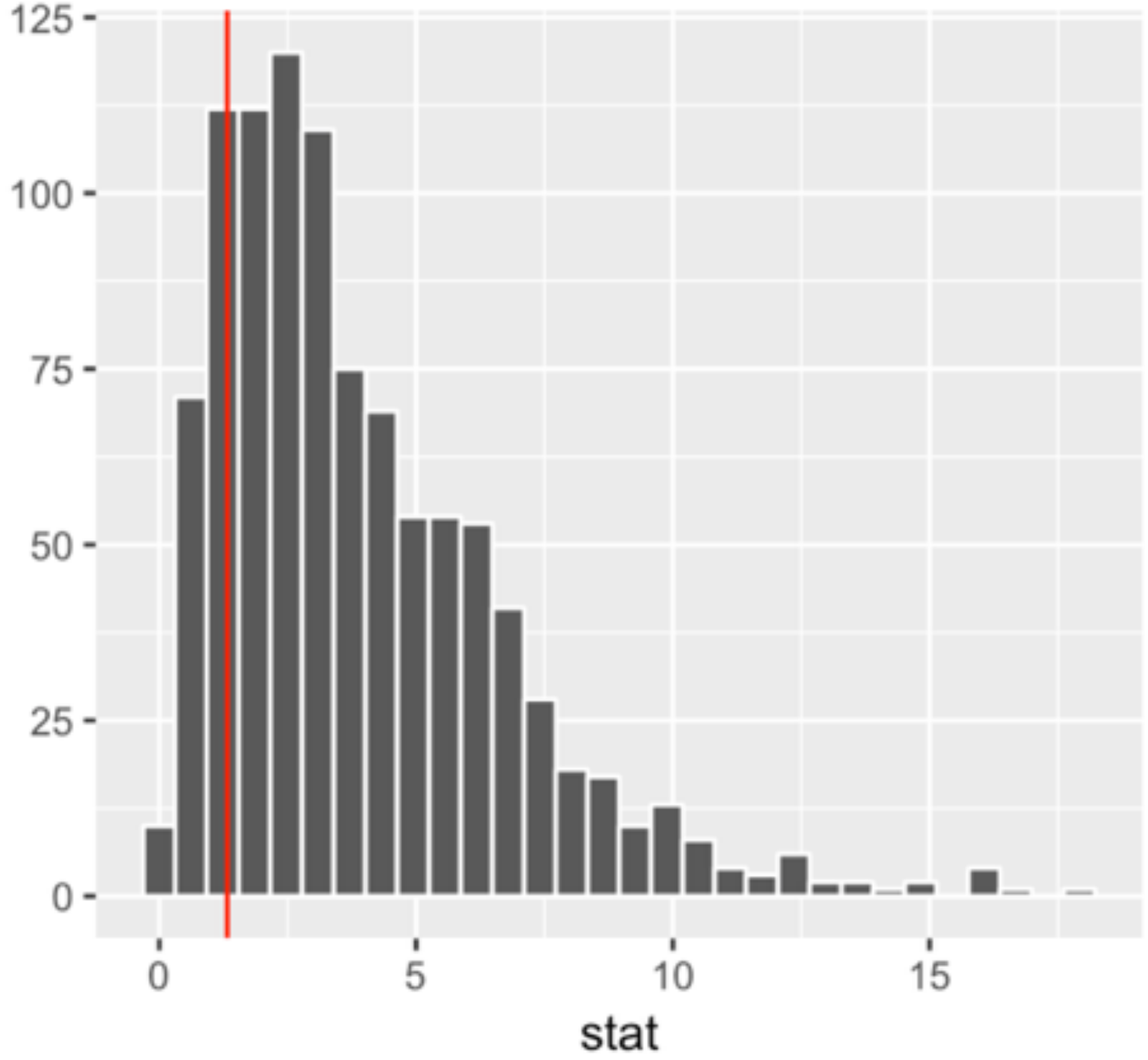


```
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq") %>%
  summarize(p_val = mean(stat > obs_stat))
```



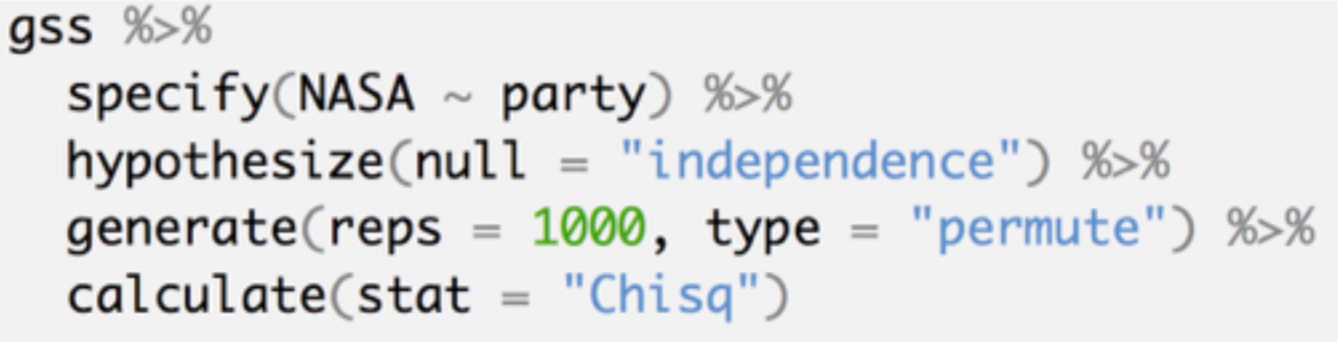
```
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq") %>%
  summarize(p_val = mean(stat > obs_stat))
```

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



Reusable parts

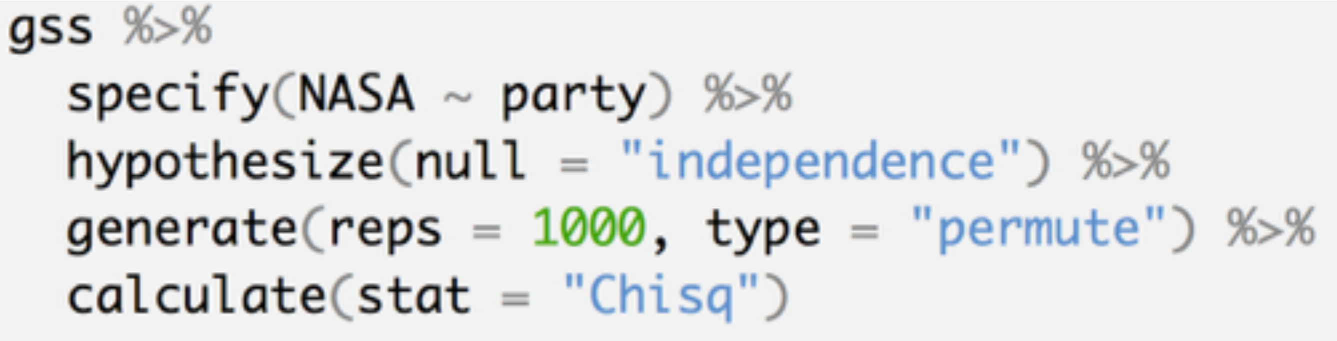
Reusable parts



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

Permutation Chi-squared

Reusable parts

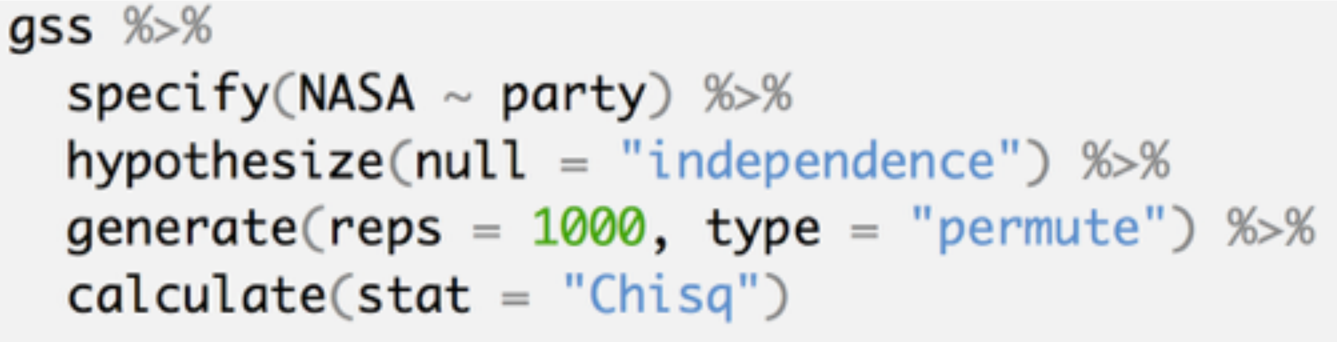


```
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq")
```

Permutation Chi-squared

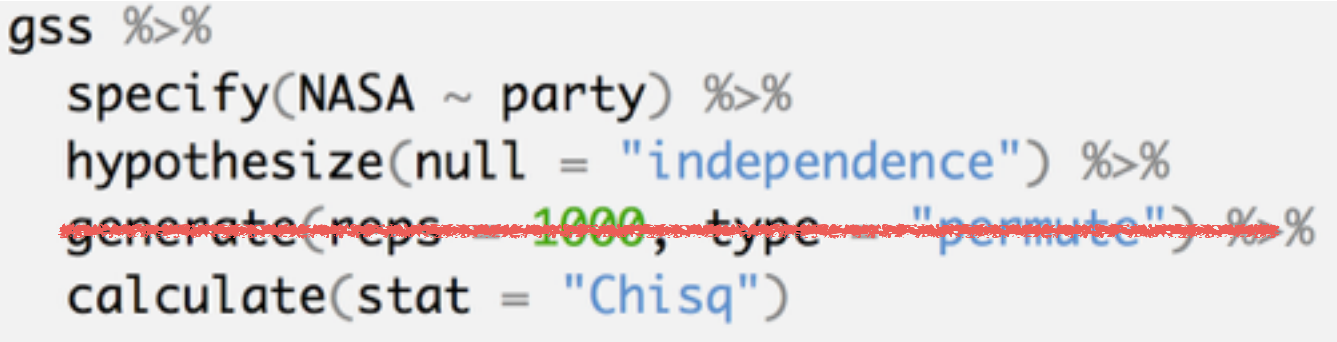
Approximation Chi-squared*

Reusable parts



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

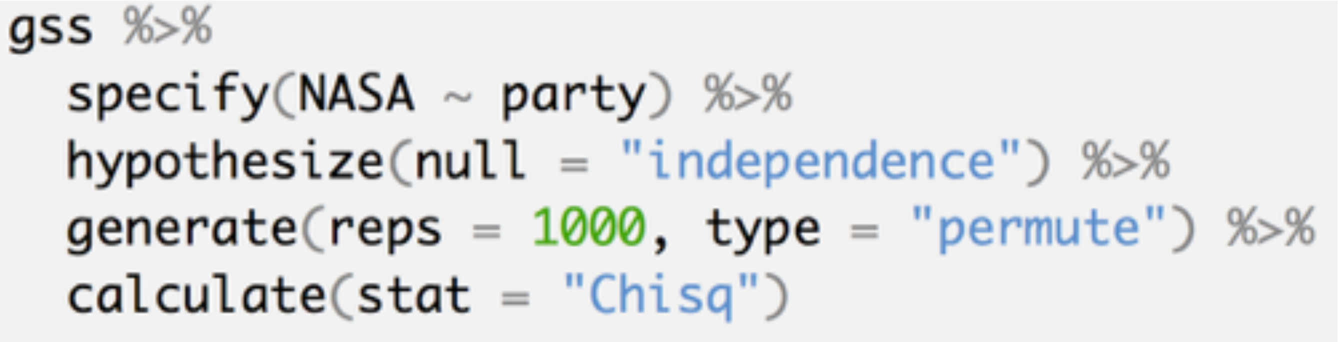
Permutation Chi-squared



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

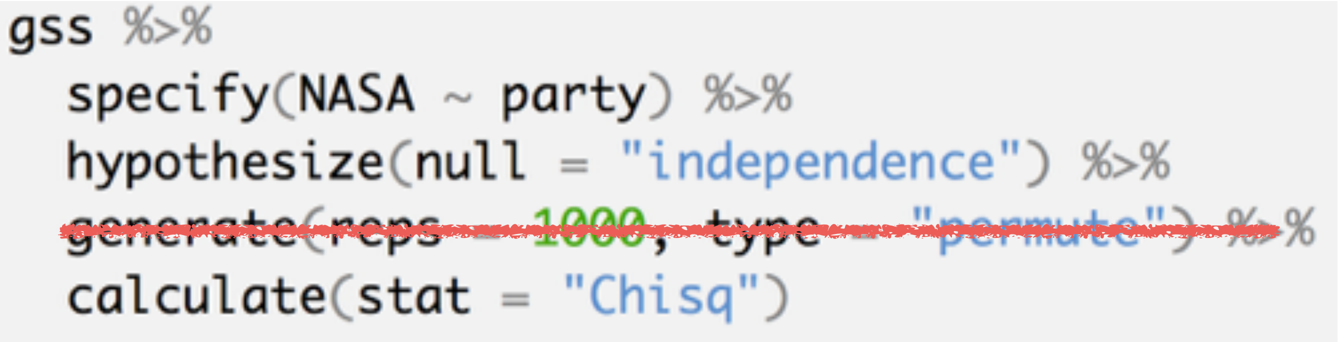
Approximation Chi-squared*

Reusable parts



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

Permutation Chi-squared

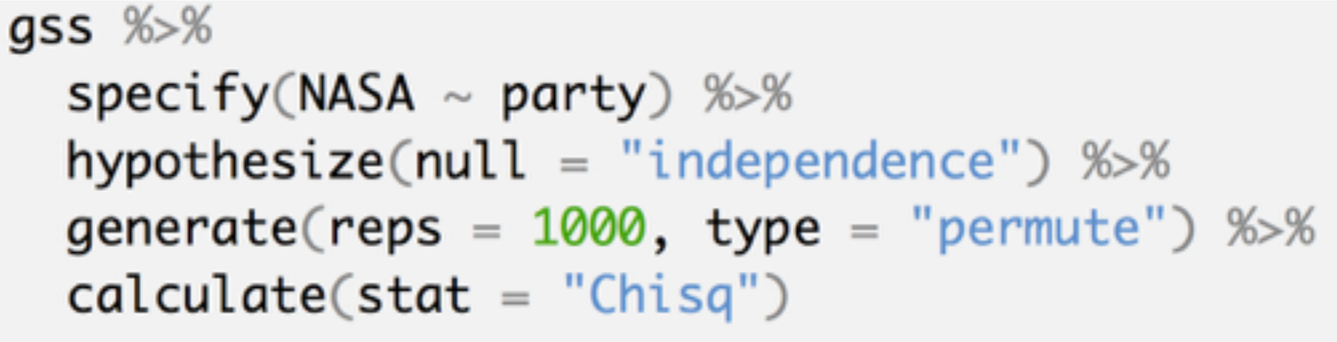


```r
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  ~~generate(reps = 1000, type = "permute") %>%~~
  calculate(stat = "Chisq")
```

Approximation Chi-squared*

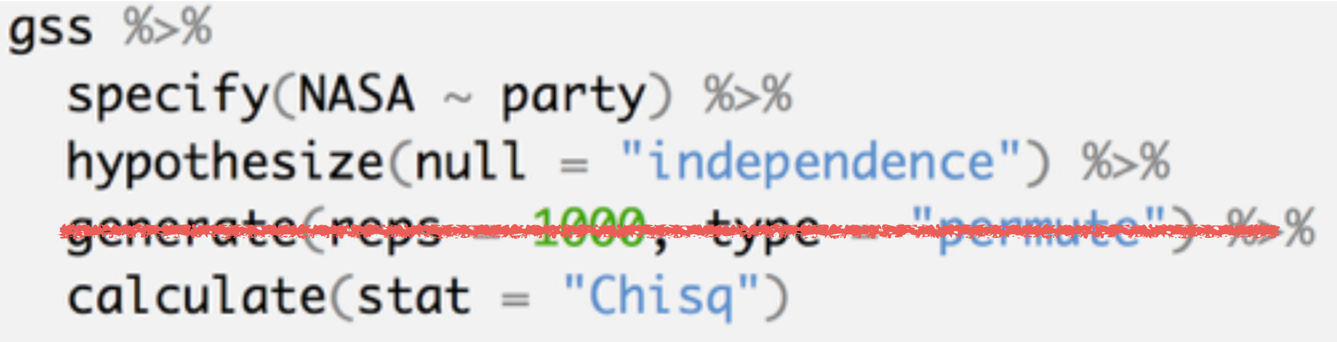
Permutation p1 - p2

Reusable parts



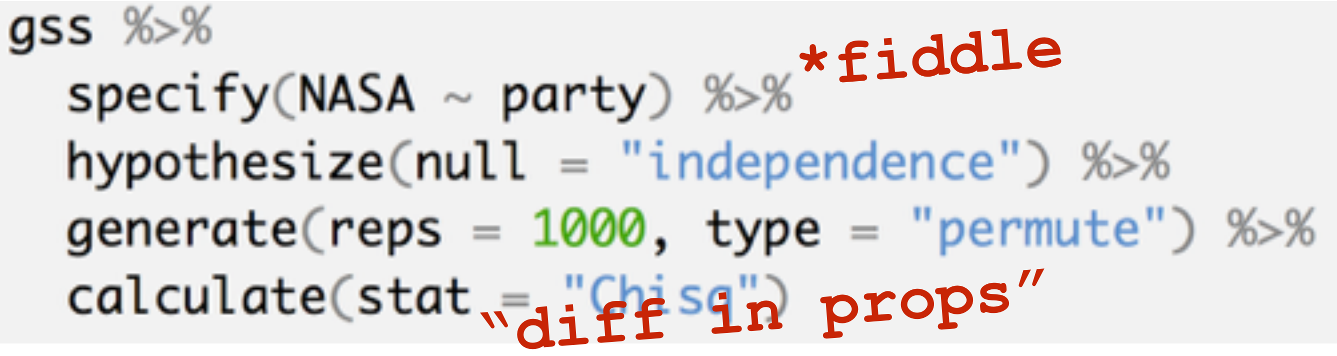
```
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq")
```

Permutation Chi-squared



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

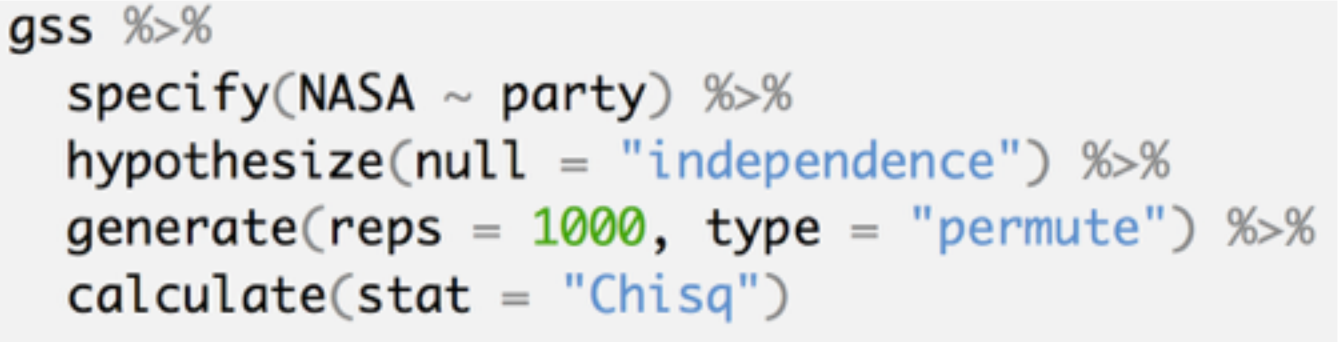
Approximation Chi-squared*



```
gss %>%
  specify(NASA ~ party) %>% *fiddle
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq")
                 “diff in props”
```

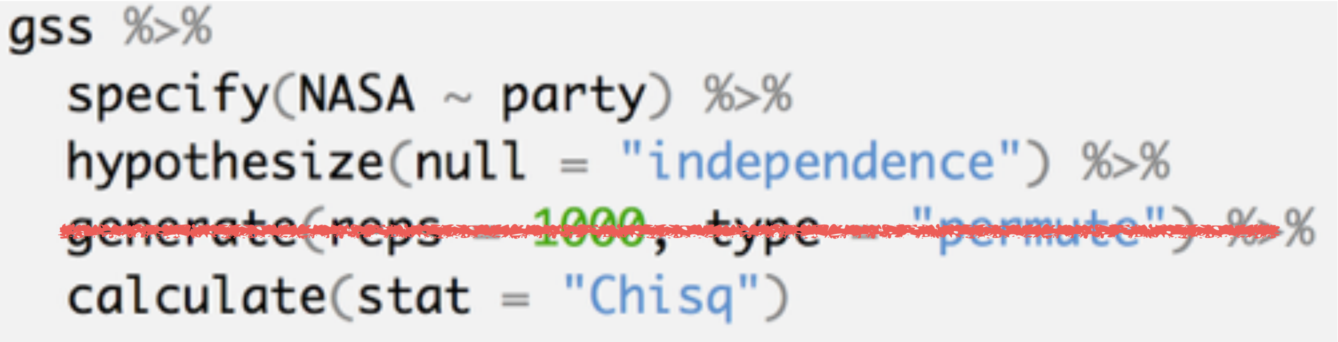
Permutation p1 - p2

Reusable parts



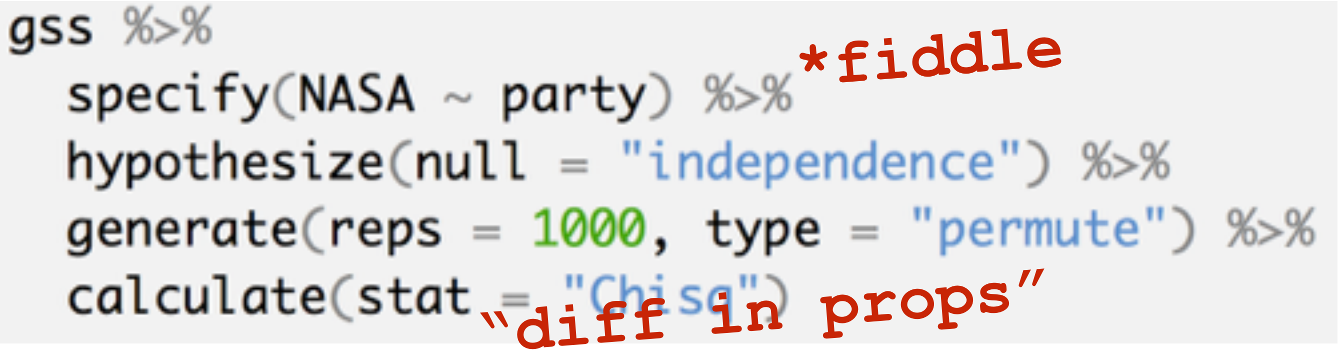
```
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq")
```

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")
```

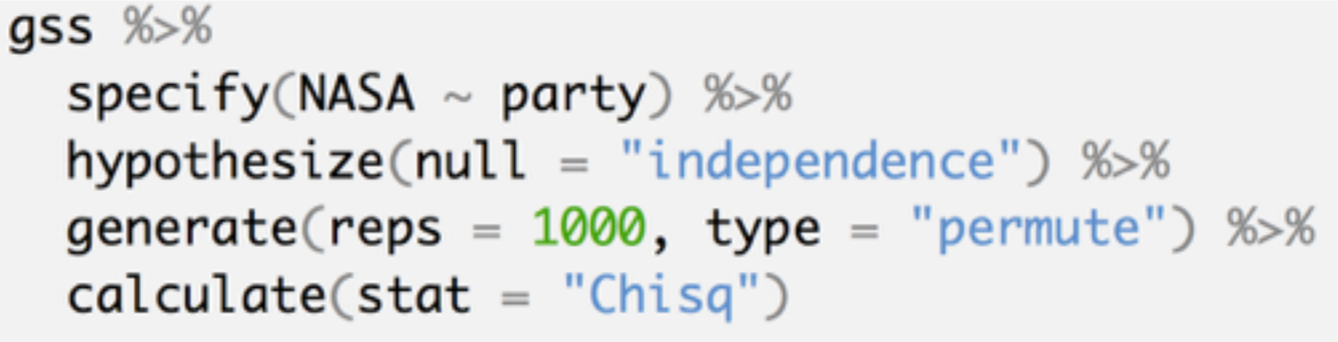
*fiddle

“diff in props”

Permutation p1 - p2

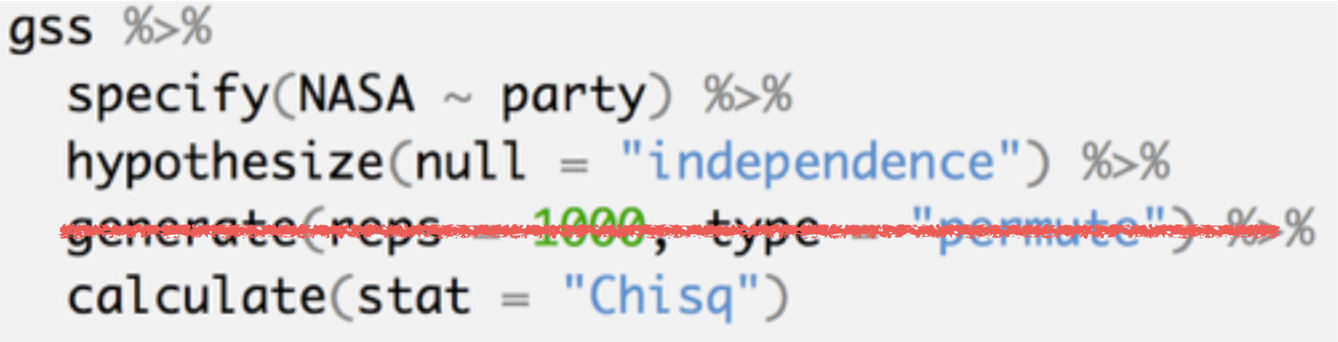
Confidence interval for p1 - p2

Reusable parts



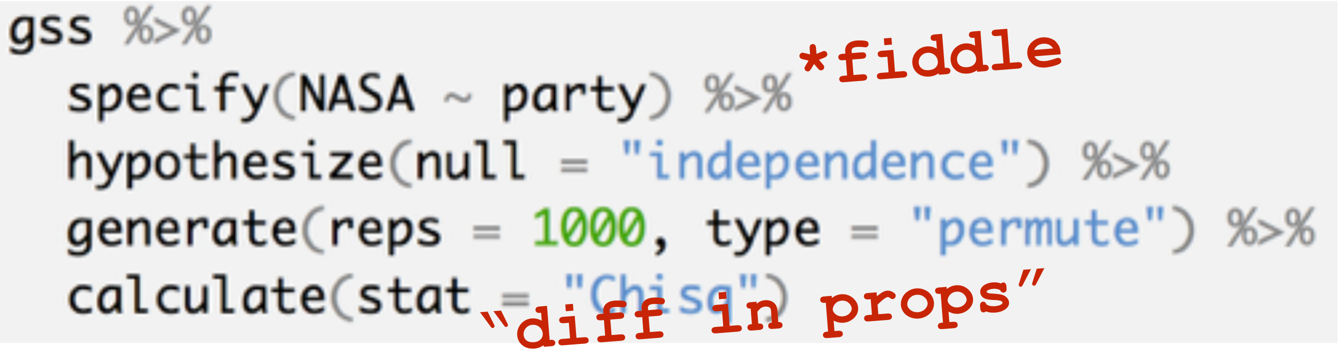
```
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq")
```

Permutation Chi-squared



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

Approximation Chi-squared*

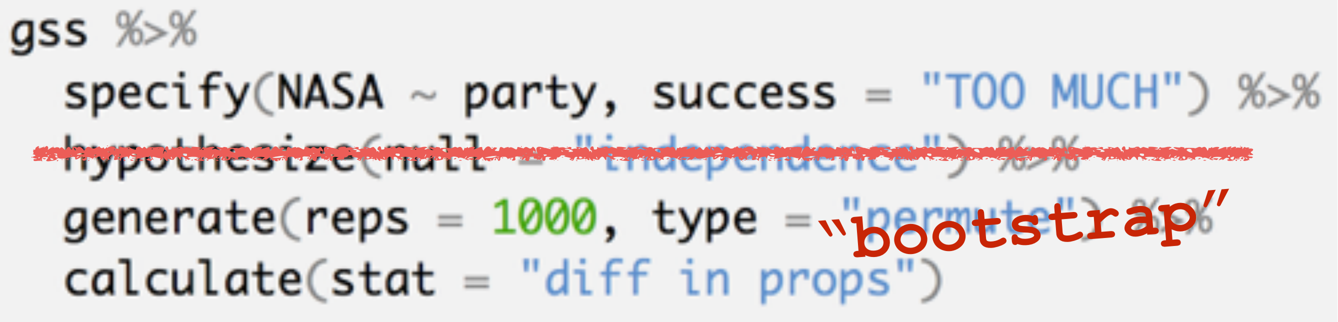


```
gss %>%
  specify(NASA ~ party) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq")
```

*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

What’s to come

What’s to come

- Generalize input to `calculate()`

What’s to come

- Generalize input to `calculate()`
    - For example, `calculate(trimmed_mean)`

What’s to come

- Generalize input to `calculate()`
  - For example, `calculate(trimmed_mean)`
  - Support for more advanced regression models

What’s to come

- Generalize input to `calculate()`
  - For example, `calculate(trimmed_mean)`
  - Support for more advanced regression models
- Spruce up `visualize()`

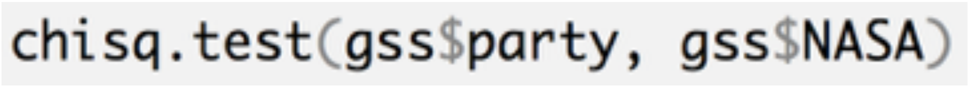
What’s to come

- Generalize input to `calculate()`
    - For example, `calculate(trimmed_mean)`
    - Support for more advanced regression models
- Spruce up `visualize()`
- Add list-columns to `generate()`

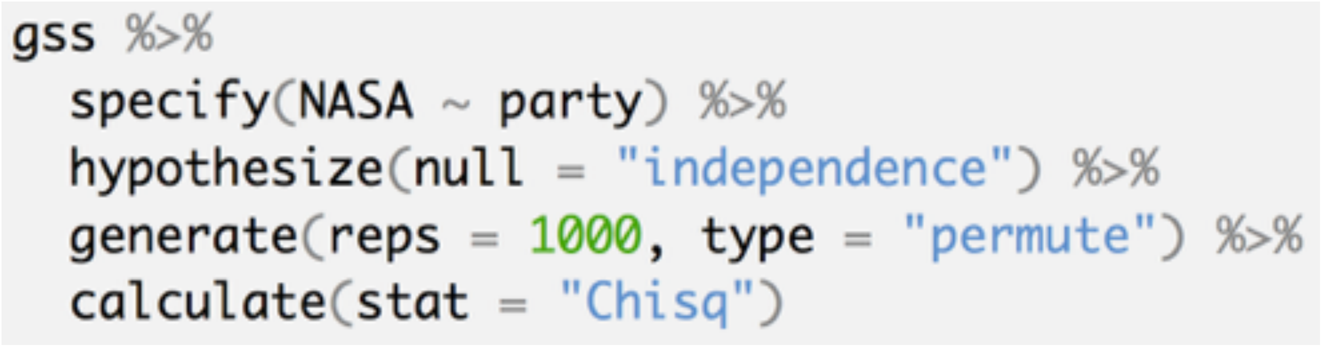
What’s to come

- Generalize input to `calculate()`
    - For example, `calculate(trimmed_mean)`
    - Support for more advanced regression models
- Spruce up `visualize()`
- Add list-columns to `generate()`
- Wrapper functions: `t_test, chisq_test,` etc.

infer makes ~~p-values~~ statistical inference ~~easier to compute.~~ tidy and transparent.



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



```
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
- slides: http://bit.ly/2DYoBOz