An Introduction to R

2.3 Simple inferential statistics

Dan Navarro (daniel.navarro@adelaide.edu.au)
School of Psychology, University of Adelaide
ua.edu.au/ccs/people/dan
DSRO R Workshop, 29-Apr-2015

Calculating a simple confidence interval

Simple normal confidence intervals

```
> mean( expt$age )
[1] 25.25
```

We already know how to
calculate the mean of a variable

Simple normal confidence intervals

```
> mean( expt$age )
[1] 25.25

> library( lsr )
```

```
> ciMean( expt$age )
    2.5%    97.5%
23.18449 27.31551
```

The lsr package has a simple function ciMean that computes the standard "normal-theory" 95% confidence interval for the mean

Simple normal confidence intervals

```
> mean( expt$age )
[1] 25.25

> library( lsr )

> ciMean( expt$age )
    2.5%    97.5%
23.18449 27.31551
```

```
> ciMean( expt$age, conf=.8 )
     10%      90%
23.97049 26.52951
```

The conf argument lets you specify a different confidence level if you like...

Use aggregate()

```
> aggregate( age ~ treatment, expt, mean )
  treatment   age
1   control 26.00
2     drug1 23.75
3     drug2 26.00
```

We've seen this already... this is one way to calculate means separately for each group

Use aggregate()

```
> aggregate( age ~ treatment, expt, mean )
  treatment   age
1   control 26.00
2     drug1 23.75
3     drug2 26.00
```

```
> aggregate( age ~ treatment, expt, ciMean )
  treatment age.2.5% age.97.5%
1   control 23.74967  28.25033
2     drug1 17.20532  30.29468
3     drug2 19.76912  32.23088
```

We can use exactly the same trick to get confidence intervals for each group mean

Applying ciMean to several variables

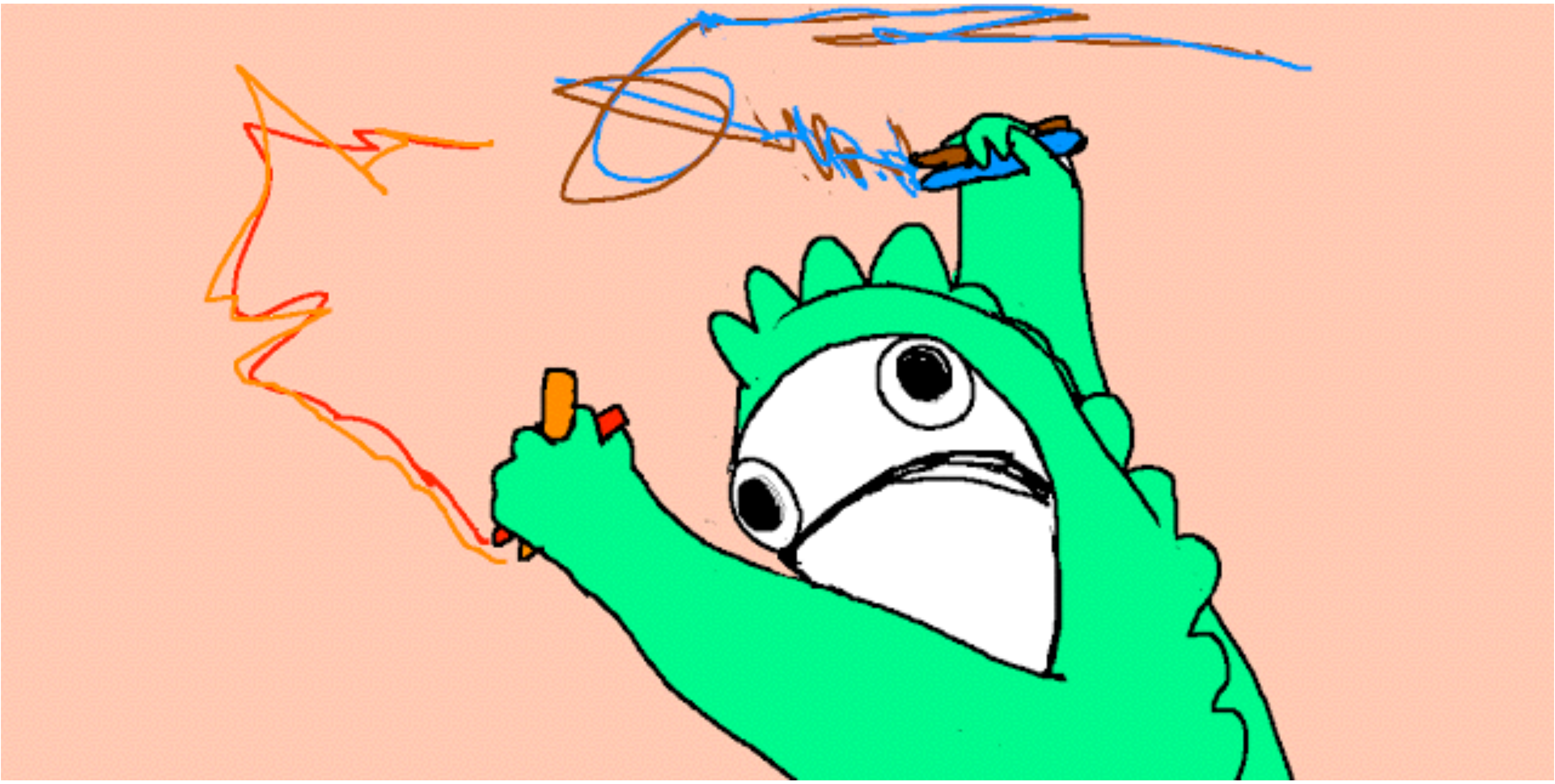
```
> ciMean( expt )

                 2.5%     97.5%
id           4.209142  8.790858
age         23.184493 27.315507
gender*            NA        NA
treatment*         NA        NA
hormone     27.793247 59.390086
happy        3.065630  4.359370
sad          2.404357  4.442309
```

If you input a data frame rather than a single variable, ciMean() calculates confidence intervals separately for each variable.

(Non-numeric variables are indicated with *, and no CI is computed for those)

Try it yourself (Exercise 2.3.1)



Comparing two means with t-tests

Comparing two means using t.test()

- The t.test() function
  - Runs several different kinds of t test
  - Handles data formatted in different ways
  - I'll only show a few of the more important ones

Independent samples t-test

```
t.test( hormone ~ gender, data = expt )
```

Formula stating that:

- hormone is the outcome (DV)
- gender is the predictor (IV)

Independent samples t-test

```
t.test( hormone ~ gender, data = expt )
```

Tells R to look for these variables
inside the data frame `expt`

```
> t.test( hormone ~ gender, data = expt )

   Welch Two Sample t-test

data:  hormone by gender
t = 0.6852, df = 7.49, p-value = 0.5138
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -24.25544  44.42211
sample estimates:
mean in group female   mean in group male
            48.63333             38.55000
```

Independent samples t-test without assuming equality of variances

```
> t.test( hormone ~ gender, data = expt )

	Welch Two Sample t-test

data:  hormone by gender
t = 0.6852, df = 7.49, p-value = 0.5138
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -24.25544  44.42211
sample estimates:
mean in group female   mean in group male
            48.63333             38.55000
```

Reminds us what variables we ran the test on

```
> t.test( hormone ~ gender, data = expt )

   Welch Two Sample t-test

data:  hormone by gender
t = 0.6852, df = 7.49, p-value = 0.5138
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -24.25544  44.42211
sample estimates:
mean in group female   mean in group male
            48.63333             38.55000
```

Test statistic, degrees of freedom, p-value

```
> t.test( hormone ~ gender, data = expt )

	Welch Two Sample t-test

data:  hormone by gender
t = 0.6852, df = 7.49, p-value = 0.5138
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -24.25544  44.42211
sample estimates:
mean in group female   mean in group male
            48.63333             38.55000
```

Two sided test

```
> t.test( hormone ~ gender, data = expt )

   Welch Two Sample t-test

data:  hormone by gender
t = 0.6852, df = 7.49, p-value = 0.5138
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -24.25544  44.42211
sample estimates:
mean in group female   mean in group male
            48.63333             38.55000
```

Confidence interval for the
difference between the two means

```
> t.test( hormone ~ gender, data = expt )

	Welch Two Sample t-test

data:  hormone by gender
t = 0.6852, df = 7.49, p-value = 0.5138
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -24.25544  44.42211
sample estimates:
mean in group female   mean in group male
            48.63333             38.55000
```

The group means

```
> t.test( hormone ~ gender, data = expt )

	Welch Two Sample t-test

data:  hormone by gender
t = 0.6852, df = 7.49, p-value = 0.5138
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -24.25544  44.42211
sample estimates:
mean in group female   mean in group male
            48.63333             38.55000
```

Variations on the t-test

```
> # one-sided test
> t.test( hormone ~ gender, data = expt, alternative = "greater" )
```

The alternative argument has three options...

- “two.sided”
- “greater”
- “less”

Variations on the t-test

```
> # one-sided test
> t.test( hormone ~ gender, data = expt, alternative = "greater" )

> # Student t-test (assumes equal variance)
> t.test( hormone ~ gender, data = expt, var.equal = TRUE )
```

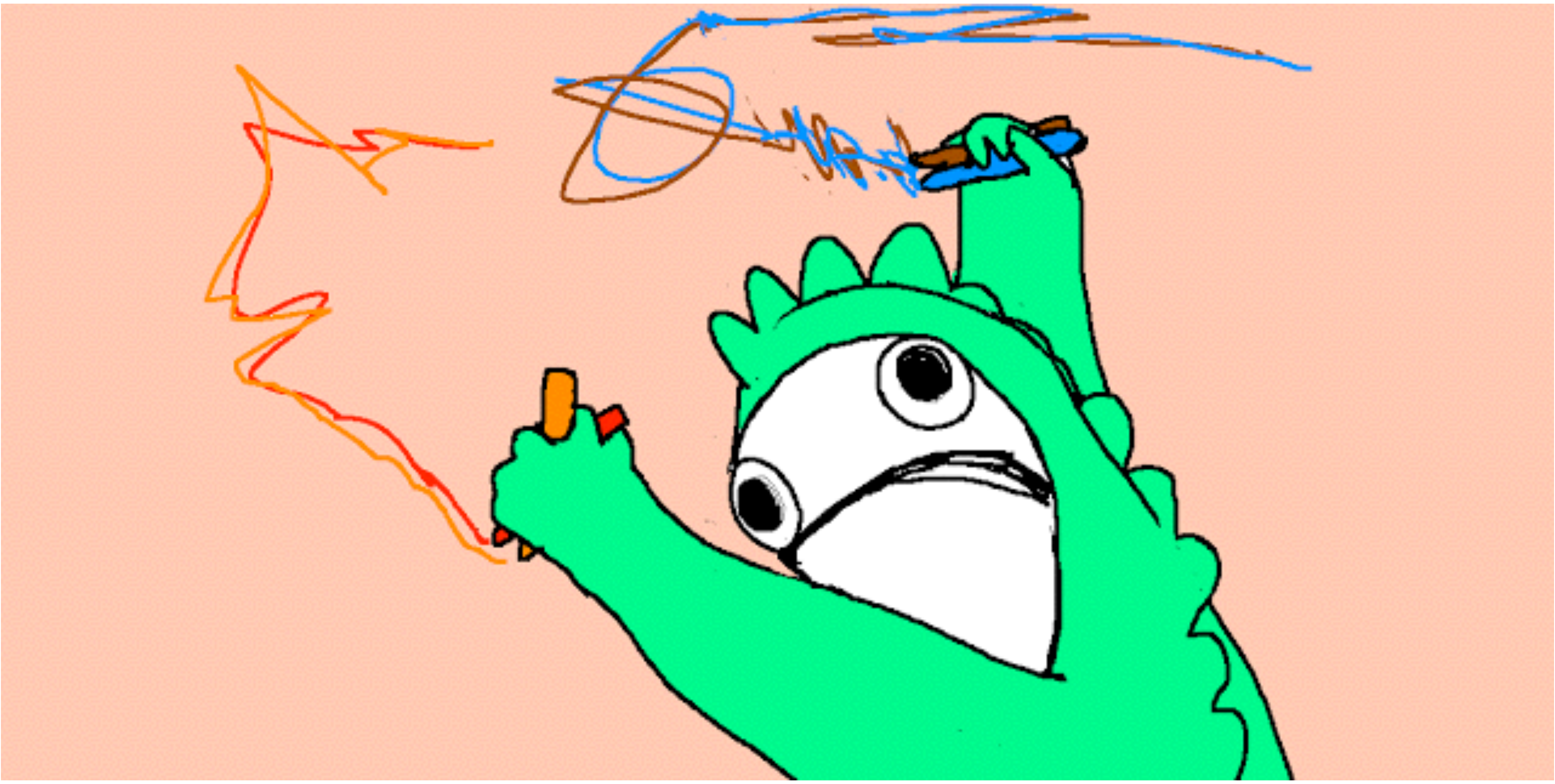
This tells R that you're happy to trust the equal variance assumption

Effect size?

- The cohensD() function (lsr package)
  - Syntax is very similar to t.test()
  - It can calculate several variants of Cohen’s d
  - Default is to use pooled standard deviation estimate

```
> library( lsr )
> cohensD( hormone ~ gender, data = expt )
[1] 0.395626
```

Try it yourself (Exercise 2.3.2)



Other kinds of t-tests

Paired samples t-test

```
> quickdata <- data.frame(
     id = c("s1","s2","s3","s4","s5"),
     time1 = c(2,3,5,9,11),
     time2 = c(1,4,7,9,13)
  )
```

I need some data appropriate for a paired samples test... manually specify a data set with 5 people tested on something at two time points

Paired samples t-test

```
> quickdata <- data.frame(
     id = c("s1","s2","s3","s4","s5"),
     time1 = c(2,3,5,9,11),
     time2 = c(1,4,7,9,13)
  )

> quickdata
  id time1 time2
1 s1     2     1
2 s2     3     4
3 s3     5     7
4 s4     9     9
5 s5    11    13
```

... and here it is!

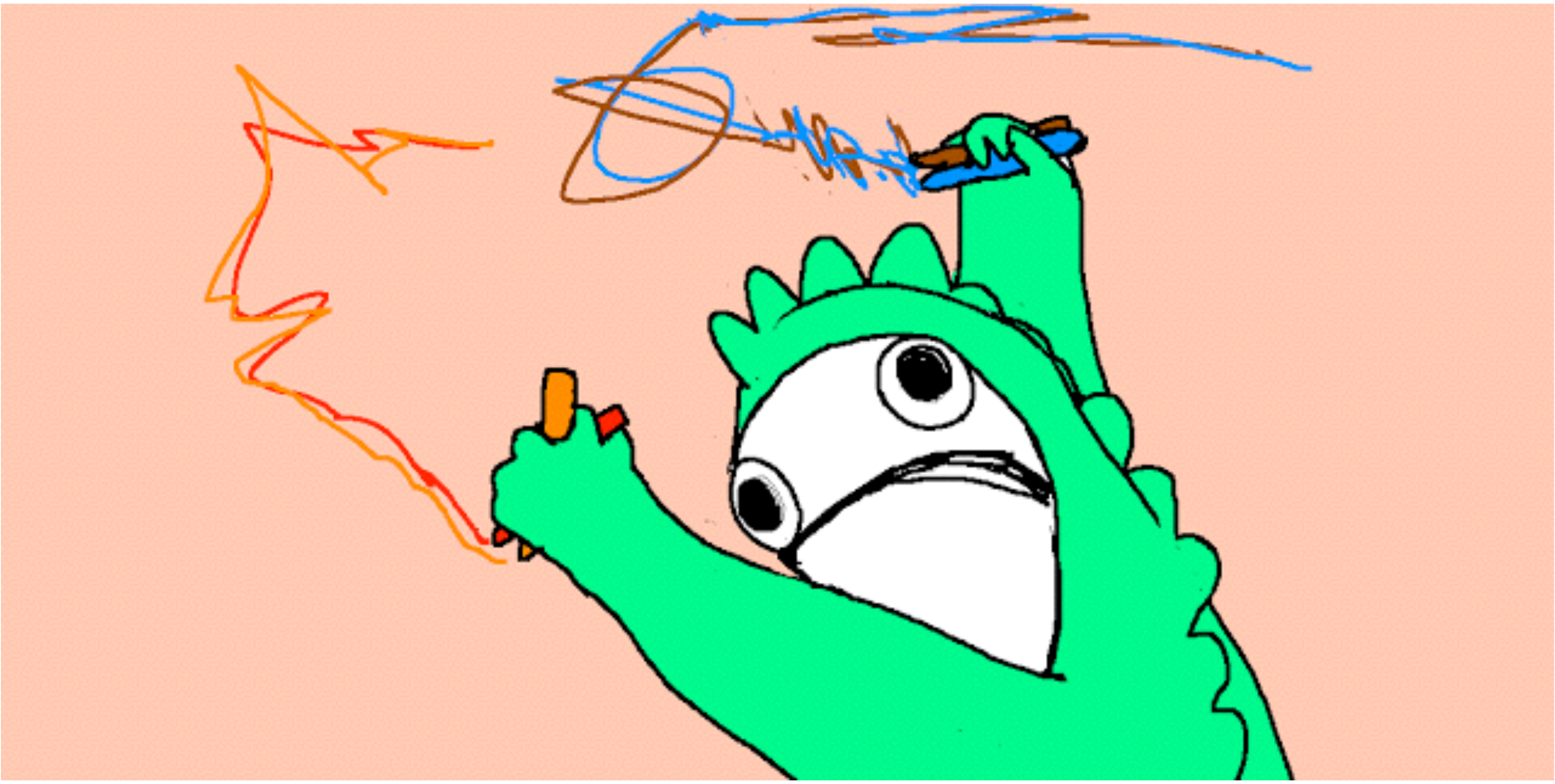
Paired samples t-test

```
> t.test( quickdata$time1, quickdata$time2, paired=TRUE )

   Paired t-test

data:  quickdata$time1 and quickdata$time2
t = -1.372, df = 4, p-value = 0.242
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.4189318  0.8189318
sample estimates:
mean of the differences
                   -0.8
```

Try it yourself (Exercise 2.3.3)



One sample t-test

```
> t.test( x = expt$hormone, mu = 50 )

   One Sample t-test

data:  expt$hormone
t = -0.8928, df = 11, p-value = 0.3911
alternative hypothesis: true mean is not equal to 50
95 percent confidence interval:
 27.79325 59.39009
sample estimates:
mean of x
 43.59167
```

```
> oneSampleTTest( expt$age, mu=20 )

   One sample t-test

Data variable:   expt$age

Descriptive statistics:
                  age
   mean        25.250
   std dev.     3.251

Hypotheses:
   null:        population mean equals 20
   alternative: population mean not equal to 20

Test results:
   t-statistic:  5.594
   degrees of freedom:  11
   p-value:  <.001

Other information:
   two-sided 95% confidence interval:  [23.184, 27.316]
   estimated effect size (Cohen's d):  1.615
```

The lsr package also has functions “oneSampleTTest”, “pairedSamplesTTest” and “independentSamplesTTest” which present information in a format that is easier for undergrads.

Nonparametric tests:
one and two sample Wilcoxon

(Two-sample) Wilcoxon rank sum test

```
> wilcox.test( hormone ~ gender, data = expt )

	Wilcoxon rank sum test

data:  hormone by gender
W = 11, p-value = 0.3095
alternative hypothesis: true location shift is not equal to 0
```

The syntax for this is basically the same thing as what we did for an independent samples t-test

(One-sample) Wilcoxon signed rank test

```
> wilcox.test( x = expt$hormone, mu = 50 )

	Wilcoxon signed rank test

data:  expt$hormone
V = 25, p-value = 0.3013
alternative hypothesis: true location is not equal to 50
```

The syntax for this is basically the same thing as what we did for a one sample t-test

Goodness of fit test +
Chi-square test of association

The chisq.test() function

- The chisq.test() function runs two different tests:
  - One variable tested against fixed probabilities
  - Two variables tested for association

- Effect size
  - cramersV() [lsr package]

The GOF test

```
> freq.gender <- table( expt$gender )
> freq.gender

female   male
     6      6
```

First, we need the frequency table

```
> null.probs <- c(.5, .5)
```

And a vector specifying the null hypothesis (e.g., males and females equally common)

The Pearson GOF test

```
> chisq.test( x = freq.gender, p = null.probs )
```

The data

The null hypothesis probabilities

The Pearson GOF test

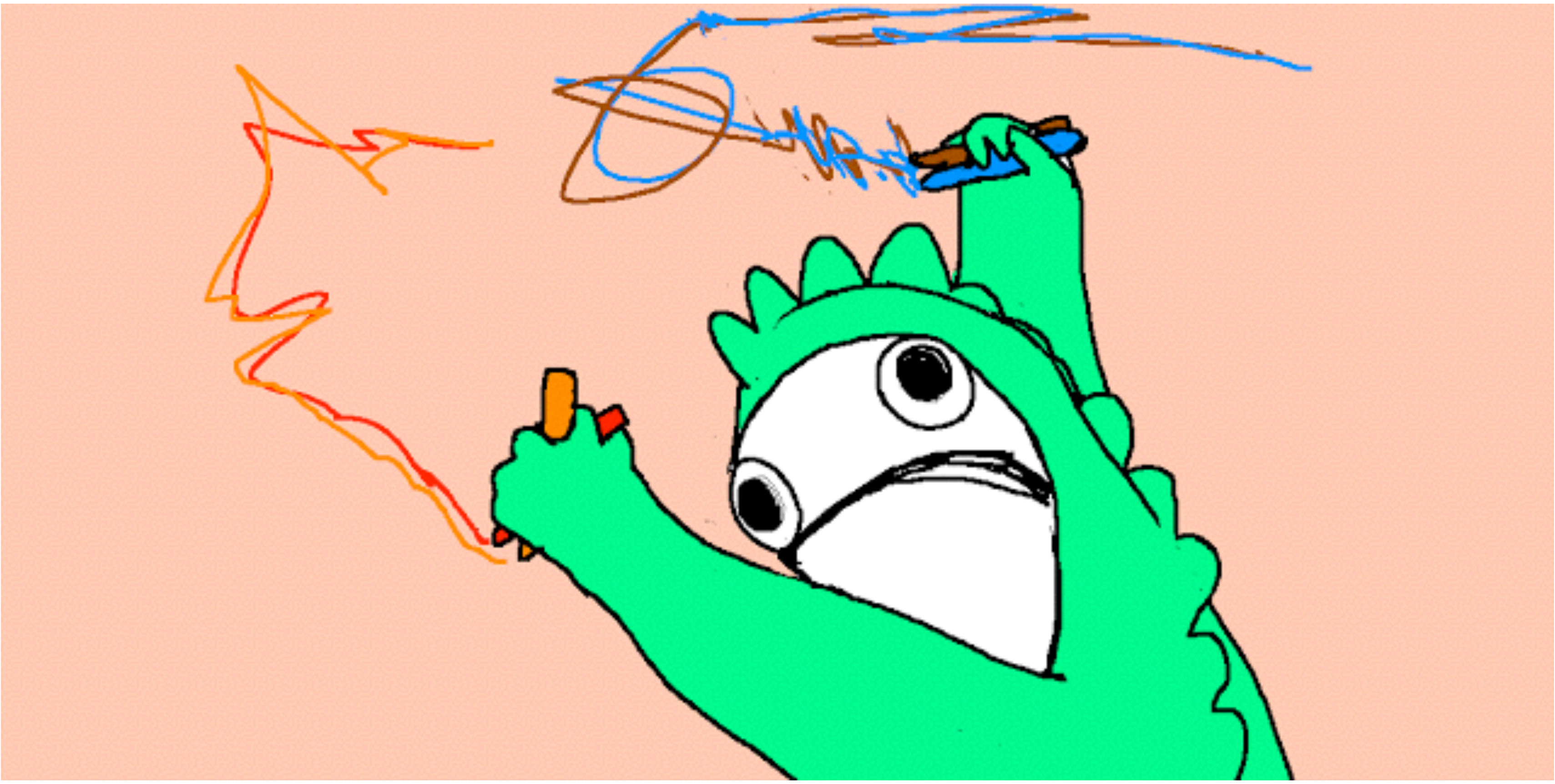
```
> chisq.test( x = freq.gender, p = null.probs )

	Chi-squared test for given probabilities

data:  freq.gender
X-squared = 0, df = 1, p-value = 1
```

Ask a stupid question, get a stupid answer... Obviously you're not going to reject the null given those data!

Try it yourself (Exercise 2.3.4)



chi-square test of association

```
> cross.tabs <- xtabs( ~ gender + age, data= expt)
> cross.tabs
        age
gender   19 21 23 24 25 26 28 29 30
  female  1  1  0  0  1  1  0  1  1
  male    0  0  1  1  2  0  2  0  0
```

Again, we start with
the frequency table

chi-square test of association

```
> chisq.test( x = cross.tabs )

	Pearson's Chi-squared test

data:  cross.tabs
X-squared = 9.3333, df = 8, p-value = 0.315

Warning message:
In chisq.test(x = cross.tabs) : Chi-squared approximation may
be incorrect
```

chi-square test of association

```
> chisq.test( x = cross.tabs )

	Pearson's Chi-squared test

data:  cross.tabs
X-squared = 9.3333, df = 8, p-value = 0.315

Warning message:
In chisq.test(x = cross.tabs) : Chi-squared approximation may
be incorrect
```

What test have we run?

chi-square test of association

```
> chisq.test( x = cross.tabs )

	Pearson's Chi-squared test

data:  cross.tabs
X-squared = 9.3333, df = 8, p-value = 0.315

Warning message:
In chisq.test(x = cross.tabs) : Chi-squared approximation may
be incorrect
```

On what data was it run?

chi-square test of association

```
> chisq.test( x = cross.tabs )

   Pearson's Chi-squared test

data:  cross.tabs
X-squared = 9.3333, df = 8, p-value = 0.315

Warning message:
In chisq.test(x = cross.tabs) : Chi-squared approximation may
be incorrect
```

What was the outcome of the test?

chi-square test of association

```
> chisq.test( x = cross.tabs )

	Pearson's Chi-squared test

data:  cross.tabs
X-squared = 9.3333, df = 8, p-value = 0.315

Warning message:
In chisq.test(x = cross.tabs) : Chi-squared approximation may
be incorrect
```

A polite warning that the expected cell counts are probably too small for the chi-square test to be safe

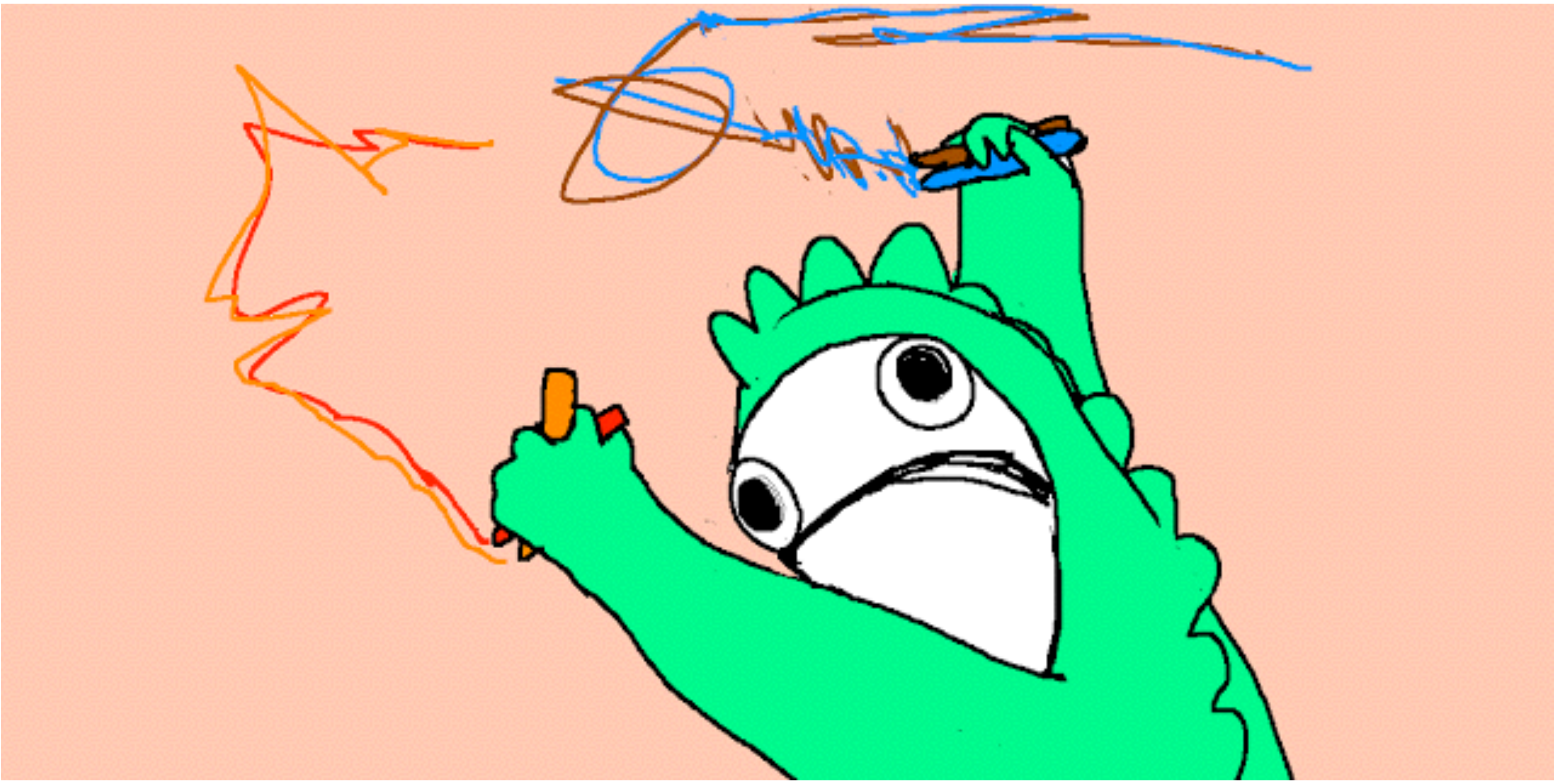
Effect size for the association

```
> library(lsr)
> cramersV( cross.tabs )

[1] 0.8819171
Warning message:
In chisq.test(...) : Chi-squared approximation may be incorrect
```

Same structure as the `chisq.test()` function itself.

Try it yourself (Exercise 2.3.5)



Fisher exact test for small samples

Problem: cell counts too small?

```
> cross.tabs
        age
gender   19 21 23 24 25 26 28 29 30
  male    0  0  1  1  2  0  2  0  0
  female  1  1  0  0  1  1  0  1  1
```

Running a chi-square test on a table like this is a bad idea! The expected cell counts are way too small for the chi-square approximation to be valid

Problem: cell counts too small?

```
> cross.tabs
        age
gender   19 21 23 24 25 26 28 29 30
  male    0  0  1  1  2  0  2  0  0
  female  1  1  0  0  1  1  0  1  1
```

```
> fisher.test( cross.tabs )

	Fisher's Exact Test for Count Data

data:  cross.tabs
p-value = 0.5455
alternative hypothesis: two.sided
```

Problem solved!

Testing the significance of a correlation

Testing a single correlation

```
> cor.test( x=expt$age, y=expt$hormone )

   Pearson's product-moment correlation

data:  expt$age and expt$hormone
t = 0.7765, df = 10, p-value = 0.4555
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.3886303  0.7145684
sample estimates:
      cor
0.2384554
```

Testing a single correlation

```
> cor.test( x=expt$age, y=expt$hormone )

	Pearson's product-moment correlation

data:  expt$age and expt$hormone
t = 0.7765, df = 10, p-value = 0.4555
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.3886303  0.7145684
sample estimates:
      cor
0.2384554
```

What kind of correlation
are we testing?

Testing a single correlation

```
> cor.test( x=expt$age, y=expt$hormone )

	Pearson's product-moment correlation

data:  expt$age and expt$hormone
t = 0.7765, df = 10, p-value = 0.4555
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.3886303  0.7145684
sample estimates:
      cor
0.2384554
```

What are the variables that are being correlated?

Testing a single correlation

```
> cor.test( x=expt$age, y=expt$hormone )

   Pearson's product-moment correlation

data:  expt$age and expt$hormone
t = 0.7765, df = 10, p-value = 0.4555
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.3886303  0.7145684
sample estimates:
      cor
0.2384554
```

The t-statistic, the degrees of freedom and the p-value

Testing a single correlation

```
> cor.test( x=expt$age, y=expt$hormone )

   Pearson's product-moment correlation

data:  expt$age and expt$hormone
t = 0.7765, df = 10, p-value = 0.4555
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.3886303  0.7145684
sample estimates:
      cor
0.2384554
```

Reminder that it's a two-sided test

Testing a single correlation

```
> cor.test( x=expt$age, y=expt$hormone )

   Pearson's product-moment correlation

data:  expt$age and expt$hormone
t = 0.7765, df = 10, p-value = 0.4555
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.3886303  0.7145684
sample estimates:
      cor
0.2384554
```

95% confidence interval for the correlation

Testing a single correlation

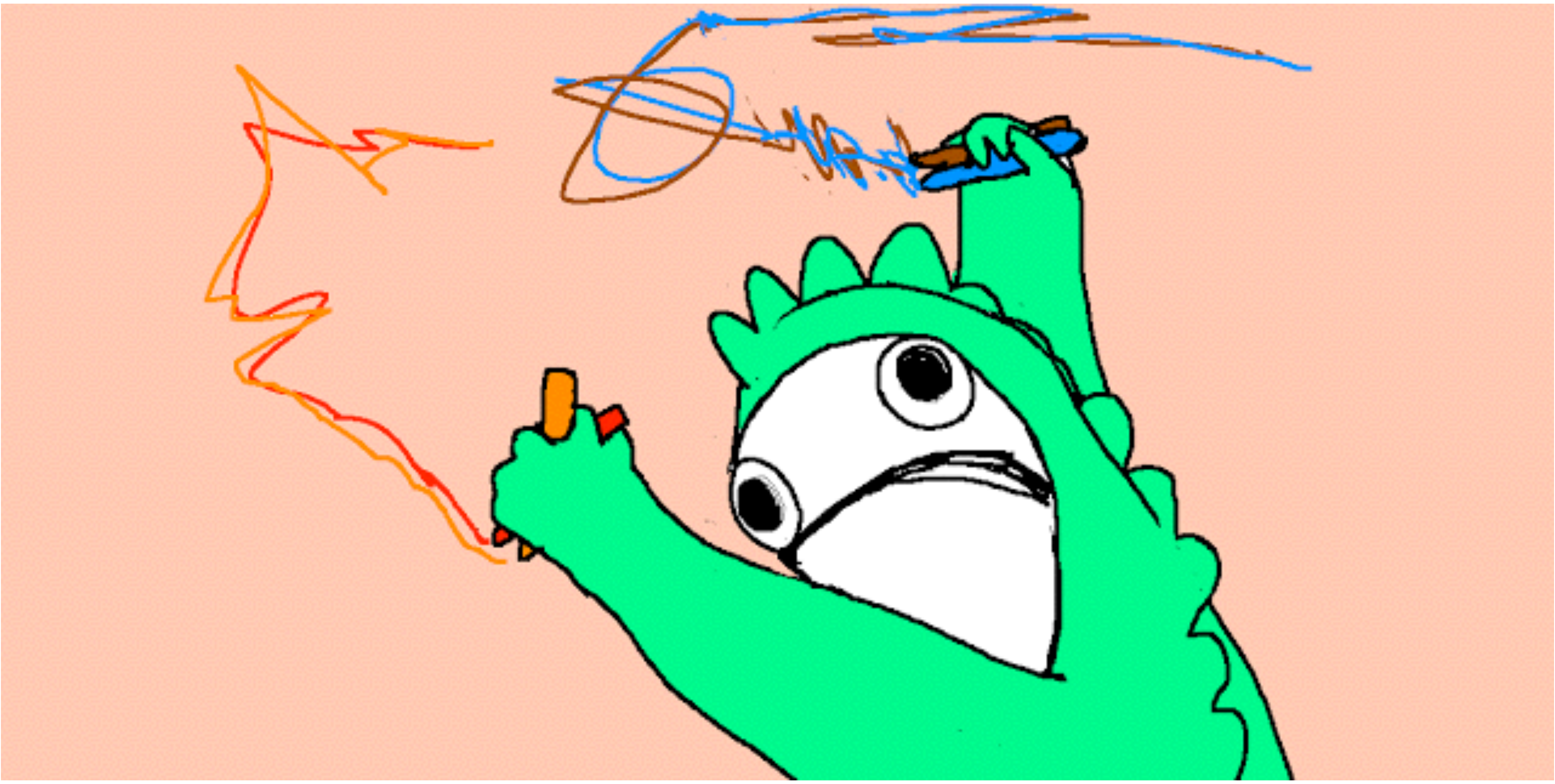
```
> cor.test( x=expt$age, y=expt$hormone )

   Pearson's product-moment correlation

data:  expt$age and expt$hormone
t = 0.7765, df = 10, p-value = 0.4555
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.3886303  0.7145684
sample estimates:
      cor
0.2384554
```

The correlation calculated
from the sample data

Try it yourself (Exercise 2.3.6)



Also works for Spearman correlations

```
> cor.test( x=expt$age, y=expt$hormone, method="spearman" )

   Spearman's rank correlation rho

data:  expt$age and expt$hormone
S = 218.4065, p-value = 0.4596
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.2363408

Warning message:
In cor.test.default(x = expt$age, y = expt$hormone, method =
"spearman") :
  Cannot compute exact p-value with ties
```

Testing all pairwise correlations?

```
> library( lsr )
> correlate( expt, test=TRUE )

CORRELATIONS
============
- correlation type:  pearson
- correlations shown only when both variables are numeric

              id       age    gender    treatment    hormone     happy       sad
id             .    -0.012         .            .      0.316     0.120    -0.486
age       -0.012         .         .            .      0.238     0.203    -0.265
gender         .         .         .            .          .         .         .
treatment      .         .         .            .          .         .         .
hormone    0.316     0.238         .            .          .      0.880** -0.952***
happy      0.120     0.203         .            .       0.880**      .    -0.733.
sad       -0.486    -0.265         .            .     -0.952*** -0.733.          .

---
Signif. codes: . = p < .1, * = p<.05, ** = p<.01, *** = p<.001
```

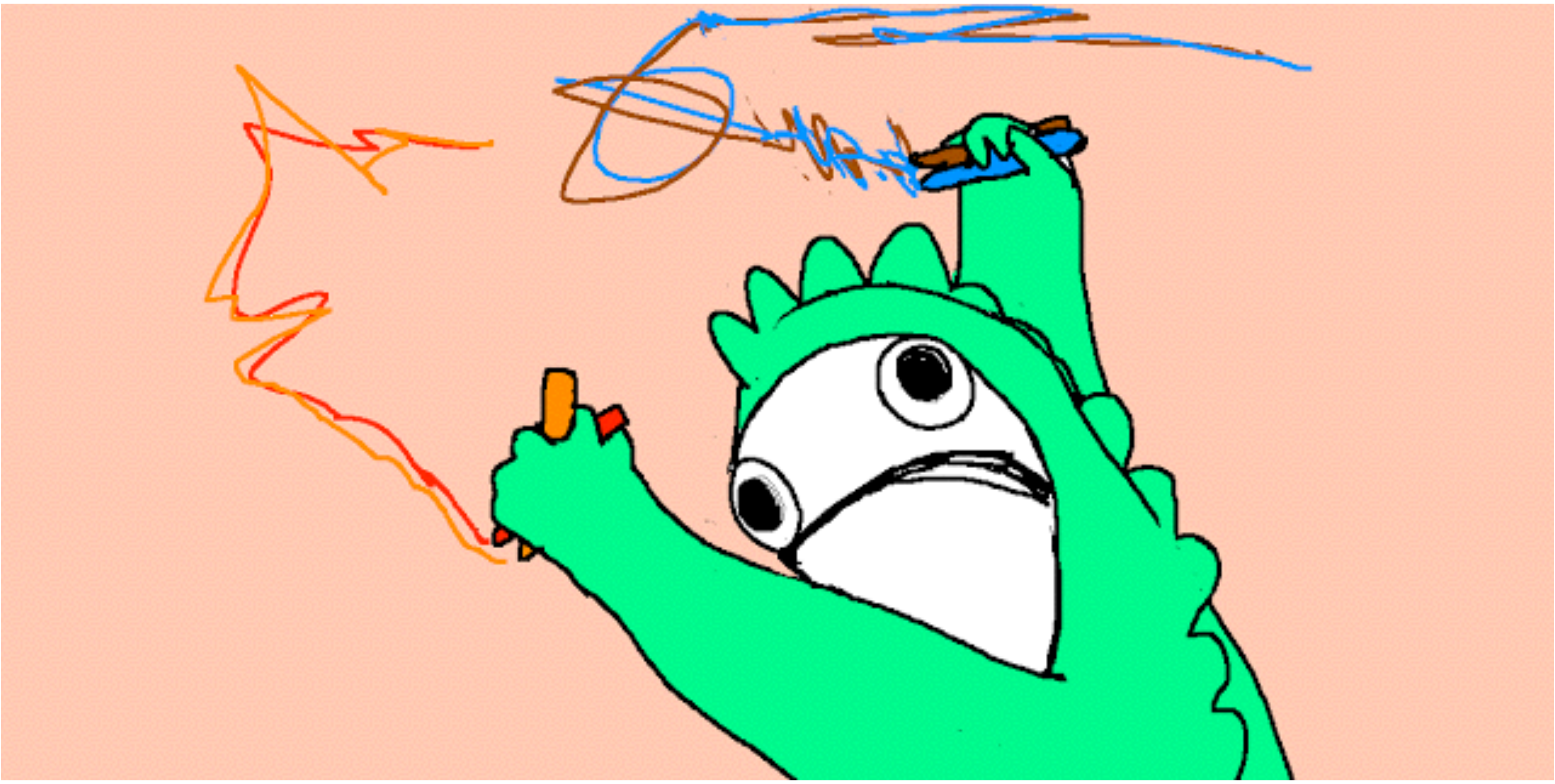
Testing all pairwise correlations?

You can override this to remove the correction for multiple tests, but I can’t think of a situation where it’s statistically justifiable to do so.

```
p-VALUES
========
- total number of tests run:  10
- correction for multiple testing:  holm

             id   age gender treatment hormone happy   sad
id            . 1.000      .         .   1.000 1.000 0.765
age       1.000     .      .         .   1.000 1.000 1.000
gender        .     .      .         .       .     .     .
treatment     .     .      .         .       .     .     .
hormone   1.000 1.000      .         .       . 0.001 0.000
happy     1.000 1.000      .         .   0.001     . 0.053
sad       0.765 1.000      .         .   0.000 0.053     .
```

Try it yourself (Exercise 2.3.7)



End of this section