# Basic statistical inference

# Packages for this section

```
library(tidyverse)
```

# Inference for means (from STAB57)

Three kinds of inference for means of normally-distributed data:

- **One-sample** $t$: a single sample from a population, estimate that population's mean
- **Two-sample** $t$: one sample from each of 2 populations, estimate difference in population means
- **Matched pairs** $t$: two paired measurements on same (or matched) individuals, estimate population mean difference

Two forms of inference for a population parameter:

- **Confidence interval**: "what is the population parameter?"
- **Hypothesis test**: "could the population parameter be equal to this value?"

# Examples:

- Blue jays attendances (one-sample)
- Kids learning to read (two-sample)
- Pain relief (matched pairs)

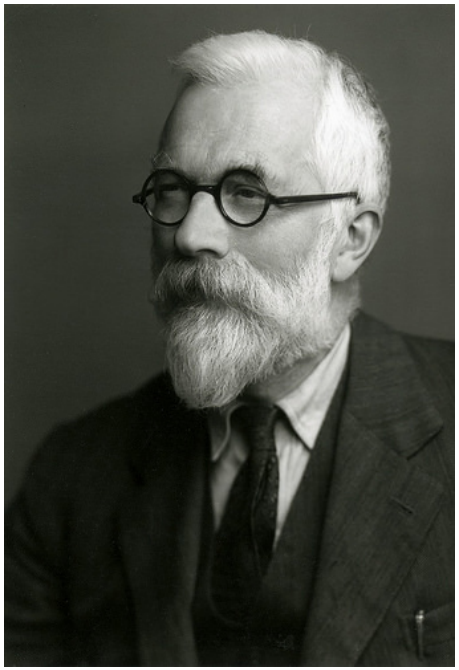
# Confidence interval

- You have a sample from some population
- Imagine repeated sampling from that population
- Procedure that gives an interval containing the true parameter in 95% (or 90% or 99%) of all possible samples

# Hypothesis test

- Null hypothesis gives value for population parameter
- Alternative hypothesis says how you are trying to prove the null hypothesis wrong (not equal, greater, less).
- Test statistic measures "distance" between data and null hypothesis
- P-value gives probability of observing test statistic *as extreme or more extreme*, **if the null hypothesis is true**.
- Reject null hypothesis if P-value small enough (eg smaller than 0.05).

# Why 0.05? This man.



- analysis of variance
- Fisher information
- Linear discriminant analysis
- Fisher's $z$-transformation
- Fisher-Yates shuffle
- Behrens-Fisher problem

Sir Ronald A. Fisher, 1890–1962.

# Why 0.05? (2)

- From The Arrangement of Field Experiments (1926):

the line at about the level at which we can say: "Either there is something in the treatment, or a coincidence has occurred such as does not occur more than once in twenty trials." This level, which we may call the 5 per cent. point, would be indicated, though very roughly, by the greatest chance deviation observed in twenty successive trials. To

- and

If one in twenty does not seem high enough odds, we may, if we prefer it, draw the line at one in fifty (the 2 per cent. point), or one in a hundred (the 1 per cent. point). Personally, the writer prefers to set a low standard of significance at the 5 per cent. point, and ignore entirely all results which fail to reach this level. A scientific fact should be regarded as experimentally established only if a properly designed experiment *rarely fails* to give this level of significance. The very high

# $\alpha$ and errors

- Hypothesis test ends with decision:
  - reject null hypothesis
  - do not reject null hypothesis.
- but decision may be wrong:

| | Decision | |
|---|---|---|
| **Truth** | **Do not reject** | **reject null** |
| **Null true** | Correct | Type I error |
| **Null false** | Type II error | Correct |

- Either type of error is bad, but for now focus on controlling Type I error: write $\alpha = \mathrm{P}(\text{type I error})$, and devise test so that $\alpha$ small, typically 0.05.
- That is, **if null hypothesis true**, have only small chance to reject it (which would be a mistake).
- Worry about type II errors later (when we consider power of test).

# One sample: the Blue Jays attendances

- The Toronto Blue Jays' average home attendance in part of 2015 season was 25,070 (up to May 27 2015, from baseball-reference.com).
- Does that mean the attendance at every game was exactly 25,070? Certainly not. Actual attendance depends on many things, eg.:
  - how well the Jays are playing
  - the opposition
  - day of week
  - weather
  - random chance

# Reading the attendances

...as a `.csv` file:

```
my_url <- "http://ritsokiguess.site/datafiles/jays15-home.csv"
jays <- read_csv(my_url)
jays
```

```
# A tibble: 25 x 21
     row  game date   box   team  venue opp   result  runs Oppruns innings
   <dbl> <dbl> <chr>  <chr> <chr> <lgl> <chr> <chr>  <dbl>   <dbl>   <dbl>
 1    82     7 Monda~ boxs~ TOR   NA    TBR   L          1       2      NA
 2    83     8 Tuesd~ boxs~ TOR   NA    TBR   L          2       3      NA
 3    84     9 Wedne~ boxs~ TOR   NA    TBR   W         12       7      NA
 4    85    10 Thurs~ boxs~ TOR   NA    TBR   L          2       4      NA
 5    86    11 Frida~ boxs~ TOR   NA    ATL   L          7       8      NA
 6    87    12 Satur~ boxs~ TOR   NA    ATL   W-wo       6       5      10
 7    88    13 Sunda~ boxs~ TOR   NA    ATL   L          2       5      NA
 8    89    14 Tuesd~ boxs~ TOR   NA    BAL   W         13       6      NA
 9    90    15 Wedne~ boxs~ TOR   NA    BAL   W          4       2      NA
10    91    16 Thurs~ boxs~ TOR   NA    BAL   W          7       6      NA
# i 15 more rows
# i 9 more variables: position <dbl>, gb <chr>, winner <chr>, loser <chr>,
#   save <chr>, `game time` <time>, Daynight <chr>, attendance <dbl>
```

# Another way

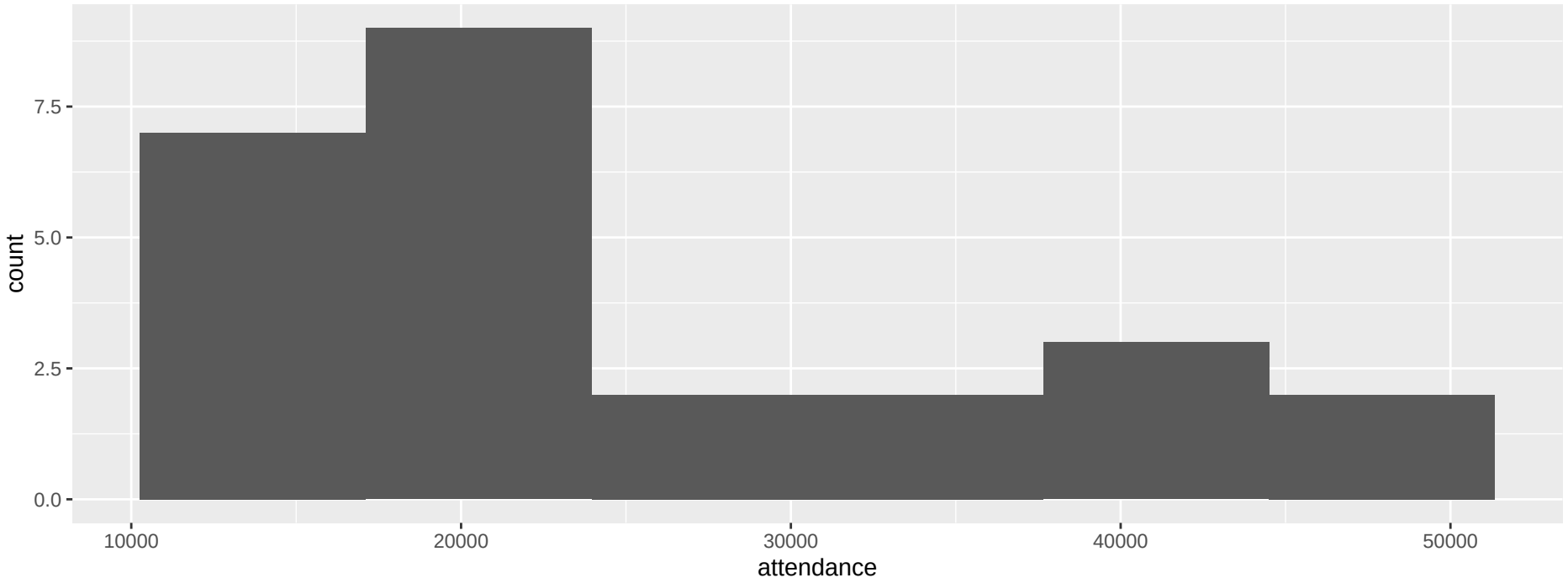
- This is a "big" data set: only 25 observations, but a lot of *variables*.
- To see the first few values in all the variables, can also use `glimpse`:

```
glimpse(jays)
```

```
Rows: 25
Columns: 21
$ row          <dbl> 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96~
$ game         <dbl> 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 27, 28, 29, 30, 31, 3~
$ date         <chr> "Monday, Apr 13", "Tuesday, Apr 14", "Wednesday, Apr 15", ~
$ box          <chr> "boxscore", "boxscore", "boxscore", "boxscore", "boxscore"~
$ team         <chr> "TOR", "TOR", "TOR", "TOR", "TOR", "TOR", "TOR", "TOR", "T~
$ venue        <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
$ opp          <chr> "TBR", "TBR", "TBR", "TBR", "ATL", "ATL", "ATL", "BAL", "B~
$ result       <chr> "L", "L", "W", "L", "L", "W-wo", "L", "W", "W", "W", "W", ~
$ runs         <dbl> 1, 2, 12, 2, 7, 6, 2, 13, 4, 7, 3, 3, 5, 7, 7, 3, 10, 2, 3~
$ Oppruns      <dbl> 2, 3, 7, 4, 8, 5, 5, 6, 2, 6, 1, 6, 1, 0, 1, 6, 6, 3, 4, 4~
$ innings      <dbl> NA, NA, NA, NA, NA, 10, NA, NA, NA, NA, NA, NA, NA, NA, NA~
$ wl           <chr> "4-3", "4-4", "5-4", "5-5", "5-6", "6-6", "6-7", "7-7", "8~
$ position     <dbl> 2, 3, 2, 4, 4, 3, 4, 2, 2, 1, 4, 5, 3, 3, 3, 3, 5, 5, 5, 5~
$ gb           <chr> "1", "2", "1", "1.5", "2.5", "1.5", "1.5", "2", "1", "Tied~
$ winner       <chr> "Odorizzi", "Geltz", "Buehrle", "Archer", "Martin", "Cecil~
$ loser        <chr> "Dickey", "Castro", "Ramirez", "Sanchez", "Cecil", "Marimo~
$ save         <chr> "Boxberger", "Jepsen", NA, "Boxberger", "Grilli", NA, "Gri~
```

# Attendance histogram

```
ggplot(jays, aes(x = attendance)) + geom_histogram(bins = 6)
```



# Comments

- Attendances have substantial variability, ranging from just over 10,000 to around 50,000.
- Distribution somewhat skewed to right (but no outliers).
- These are a sample of "all possible games" (or maybe "all possible games played in April and May"). What can we say about mean attendance in all possible games based on this evidence?

# CI for mean attendance

- `t.test` function does CI and test. Look at CI first:

```
t.test(jays$attendance)
```

```
    One Sample t-test

data:  jays$attendance
t = 11.389, df = 24, p-value = 3.661e-11
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 20526.82 29613.50
sample estimates:
mean of x
 25070.16
```

- From 20,500 to 29,600.

# Or, 90% CI

- by including a value for conf.level:

```
t.test(jays$attendance, conf.level = 0.90)
```

```
    One Sample t-test

data:  jays$attendance
t = 11.389, df = 24, p-value = 3.661e-11
alternative hypothesis: true mean is not equal to 0
90 percent confidence interval:
 21303.93 28836.39
sample estimates:
mean of x
 25070.16
```

- From 21,300 to 28,800. (Shorter, as it should be.)

# Comments

- Need to say "column attendance within data frame `jays`" using `$`.
- 95% CI from about 20,000 to about 30,000.
- Not estimating mean attendance well at all!
- Generally want confidence interval to be shorter, which happens if:
  - SD smaller
  - sample size bigger
  - confidence level smaller
- Last one is a cheat, really, since reducing confidence level increases chance that interval won't contain pop. mean at all!

# Another way to access data frame columns

```
with(jays, t.test(attendance))
```

```
    One Sample t-test

data:  attendance
t = 11.389, df = 24, p-value = 3.661e-11
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 20526.82 29613.50
sample estimates:
mean of x
 25070.16
```

# Hypothesis testing for Blue Jays attendances

- Previous year's mean attendance was 29,327, so test to see whether the mean is different from that in any way (two-sided test):

```
t.test(jays$attendance, mu = 29327)
```

```
    One Sample t-test

data:  jays$attendance
t = -1.9338, df = 24, p-value = 0.06502
alternative hypothesis: true mean is not equal to 29327
95 percent confidence interval:
 20526.82 29613.50
sample estimates:
mean of x
 25070.16
```

- See test statistic $-1.93$, P-value 0.065.

# Another example: learning to read

- You devised new method for teaching children to read.
- Guess it will be more effective than current methods.
- To support this guess, collect data.
- Want to generalize to "all children in Canada".
- So take random sample of all children in Canada.
- Or, argue that sample you actually have is "typical" of all children in Canada.
- Randomization (1): whether or not a child in sample or not has nothing to do with anything else about that child.
- Randomization (2): randomly choose whether each child gets new reading method (t) or standard one (c).

# Reading in data

- File at http://ritsokiguess.site/datafiles/drp.txt.
- Proper reading-in function is `read_delim` (check file to see)
- Read in thus:

```
my_url <- "http://ritsokiguess.site/datafiles/drp.txt"
kids <- read_delim(my_url," ")
```

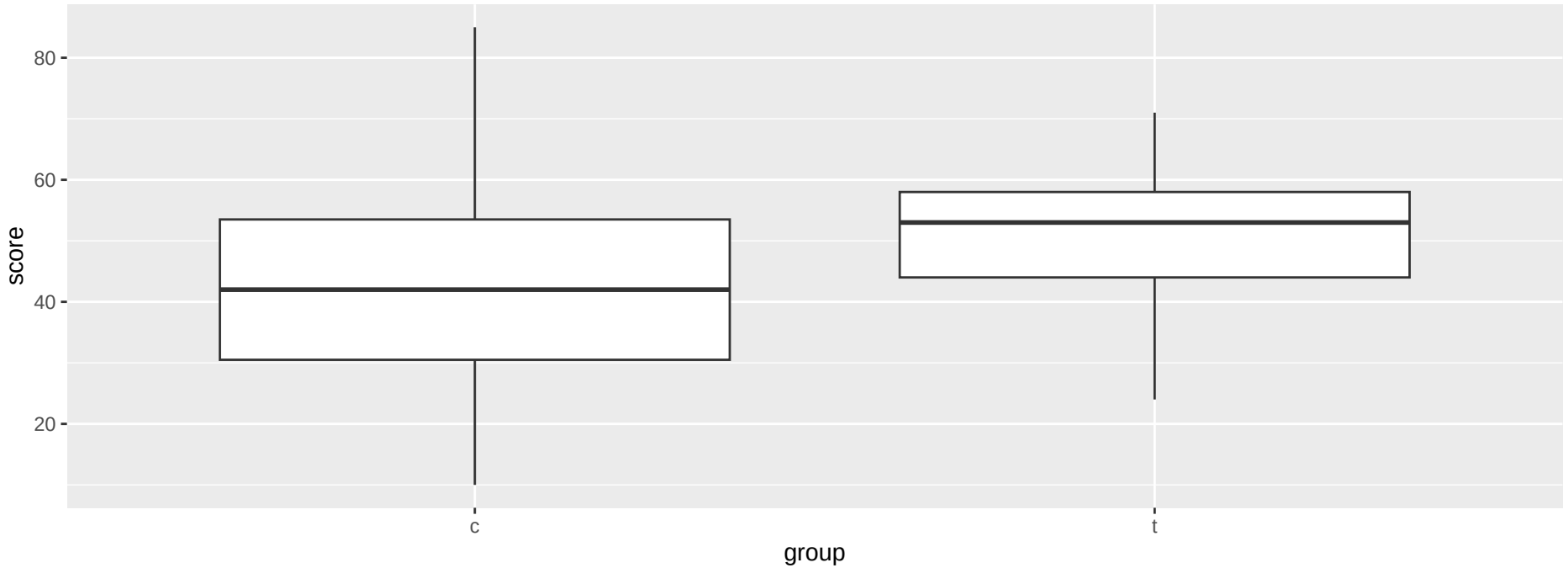
# The data (some)

```
kids
```

```
# A tibble: 44 x 2
   group score
   <chr> <dbl>
 1 t        24
 2 t        61
 3 t        59
 4 t        46
 5 t        43
 6 t        44
 7 t        52
 8 t        43
 9 t        58
10 t        67
# i 34 more rows
```

# Boxplots

```
ggplot(kids, aes(x = group, y = score)) + geom_boxplot()
```



# Two kinds of two-sample t-test

- Do the two groups have same spread (SD, variance)?
  - If yes (shaky assumption here), can use pooled t-test.
  - If not, use Welch-Satterthwaite t-test (safe).
- Pooled test derived in STAB57 (easier to derive, but assumes equal variances).
- Welch-Satterthwaite does not assume equality of variances.
- Assess (approx) equality of spreads using boxplot.

# The (Welch-Satterthwaite) t-test

- `c` (control) before `t` (treatment) alphabetically, so proper alternative is "less".
- R does Welch-Satterthwaite test by default
- Answer to "does the new reading program really help?"
- (in a moment) how to get R to do pooled test?

# Welch-Satterthwaite

```
t.test(score ~ group, data = kids, alternative = "less")
```

```
    Welch Two Sample t-test

data:  score by group
t = -2.3109, df = 37.855, p-value = 0.01319
alternative hypothesis: true difference in means between group
95 percent confidence interval:
      -Inf -2.691293
sample estimates:
mean in group c mean in group t
       41.52174        51.47619
```

# The pooled t-test

```
t.test(score ~ group, data = kids,
       alternative = "less", var.equal = TRUE)
```

```
    Two Sample t-test

data:  score by group
t = -2.2666, df = 42, p-value = 0.01431
alternative hypothesis: true difference in means between group
95 percent confidence interval:
      -Inf -2.567497
sample estimates:
mean in group c mean in group t
       41.52174        51.47619
```

# Two-sided test; CI

- To do 2-sided test, leave out `alternative`:

```
t.test(score ~ group, data = kids)
```

```
    Welch Two Sample t-test

data:  score by group
t = -2.3109, df = 37.855, p-value = 0.02638
alternative hypothesis: true difference in means between group
95 percent confidence interval:
 -18.67588  -1.23302
sample estimates:
mean in group c mean in group t
       41.52174        51.47619
```

# Comments:

- P-values for pooled and Welch-Satterthwaite tests very similar (even though the pooled test seemed inferior): 0.013 vs. 0.014.
- Two-sided test also gives CI: new reading program increases average scores by somewhere between about 1 and 19 points.
- Confidence intervals inherently two-sided, so do 2-sided test to get them.

# Pain relief

Some data:

| subject | druga | drugb |
|---|---|---|
| 1 | 2.0 | 3.5 |
| 2 | 3.6 | 5.7 |
| 3 | 2.6 | 2.9 |
| 4 | 2.6 | 2.4 |
| 5 | 7.3 | 9.9 |
| 6 | 3.4 | 3.3 |
| 7 | 14.9 | 16.7 |
| 8 | 6.6 | 6.0 |
| 9 | 2.3 | 3.8 |
| 10 | 2.0 | 4.0 |
| 11 | 6.8 | 9.1 |
| 12 | 8.5 | 20.9 |

# Matched pairs data

- Data are comparison of 2 drugs for effectiveness at reducing pain.
  - 12 subjects (cases) were arthritis sufferers
  - Response is #hours of pain relief from each drug.
- In reading example, each child tried only one reading method.
- But here, each subject tried out both drugs, giving us two measurements.
  - Possible because, if you wait long enough, one drug has no influence over effect of other.
  - Advantage: focused comparison of drugs. Compare one drug with another on same person, removes a lot of variability due to differences between people.
  - Matched pairs, requires different analysis.
- Design: randomly choose 6 of 12 subjects to get drug A first, other 6 get drug B first.

# Paired t test: reading the data

Values aligned in columns:

```
my_url <- "http://ritsokiguess.site/datafiles/analgesic.txt"
pain <- read_table(my_url)
```

# The data

```
pain
```

```
# A tibble: 12 x 3
   subject druga drugb
     <dbl> <dbl> <dbl>
 1       1   2     3.5
 2       2   3.6   5.7
 3       3   2.6   2.9
 4       4   2.6   2.4
 5       5   7.3   9.9
 6       6   3.4   3.3
 7       7  14.9  16.7
 8       8   6.6   6
 9       9   2.3   3.8
10      10   2     4
11      11   6.8   9.1
12      12   8.5  20.9
```

# Paired *t*-test

```
with(pain, t.test(druga, drugb, paired = T))
```

```
    Paired t-test

data:  druga and drugb
t = -2.1677, df = 11, p-value = 0.05299
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -4.29941513  0.03274847
sample estimates:
mean difference
      -2.133333
```

- P-value is 0.053.
- Not quite evidence of difference between drugs.

# t-testing the differences

- Likewise, you can calculate the differences yourself and do a 1-sample t-test on them.
- First calculate a column of differences:

```
(pain %>% mutate(diff=druga-drugb) -> pain)
```

```
# A tibble: 12 x 4
   subject druga drugb    diff
     <dbl> <dbl> <dbl>   <dbl>
 1       1   2     3.5  -1.5
 2       2   3.6   5.7  -2.1
 3       3   2.6   2.9  -0.300
 4       4   2.6   2.4   0.200
 5       5   7.3   9.9  -2.6
 6       6   3.4   3.3   0.100
 7       7  14.9  16.7  -1.80
 8       8   6.6   6     0.600
 9       9   2.3   3.8  -1.5
10      10   2     4    -2
11      11   6.8   9.1  -2.3
12      12   8.5  20.9 -12.4
```

# t-test on the differences

- then throw them into t.test, testing that the mean is zero, with same result as before:

```
with(pain, t.test(diff, mu=0))
```

```
    One Sample t-test

data:  diff
t = -2.1677, df = 11, p-value = 0.05299
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -4.29941513  0.03274847
sample estimates:
mean of x
-2.133333
```