

# The basics of statistical testing

07/11/2023

# dplyr and data wrangling

Function	Effect
<code>select()</code>	Include or exclude variables (columns)
<code>arrange()</code>	Change the order of observations (rows)
<code>filter()</code>	Include or exclude observations (rows)
<code>mutate()</code>	Create new variables (columns)
<code>group_by()</code>	Create groups of observations
<code>summarise()</code>	Aggregate or summarise groups of observations (rows)

# Importing, transforming, and summarising your data

```
library(readr)
crime <- read_csv("data/crime.csv")
crime
```

```
## # A tibble: 301 × 15
##   Participi...1 sex      age victi...2      H      E      X      A      C      O      SA      TA
##   <chr>         <chr> <dbl> <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 R_01TjXg... male     55 yes     3.7    3     3.4    3.9    3.2    3.6    1.15   1.55
## 2 R_0dN5Ye... fema...   20 no      2.5    3.1    2.5    2.4    2.2    3.1    2.05   2.95
## 3 R_0DPiPY... male     57 yes     2.6    3.1    3.3    3.1    4.3    2.8    2     2.6
## 4 R_0f7bSs... male     19 no      3.5    1.8    3.3    3.4    2.1    2.7    1.55   2.1
## 5 R_0rov2R... fema...   20 no      3.3    3.4    3.9    3.2    2.8    3.9    1.3    1.8
## 6 R_0wioqG... fema...   20 no      2.6    2.6    3     2.6    2.9    3.4    2.55   1.5
## 7 R_0wR08l... male     34 yes     3.2    2.5    3.2    2.8    4     3.2    1.85   1.75
## 8 R_116nEd... fema...   19 no      2.9    4     3.9    4.2    3.7    1.9    1.1    2
## 9 R_11ZmBd... fema...   19 yes     3.4    3.4    3.3    3.4    3.2    3.2    2.2    2.9
## 10 R_12i26Q... male     20 no      2.4    2.1    1.8    2.2    3.4    2.9    2.15   2.4
## # ... with 291 more rows, 3 more variables: OHQ <dbl>, FoC <dbl>, Foc2 <dbl>, and
```

# Importing, transforming, and summarising your data

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.2.2
```

```
prison_pop <- read_excel("data/prison-population-data-tool-31-december-2017.xlsx",  
                        sheet = "PT Data")  
  
filter(prison_pop,  
       View == "a Establishment*Sex*Age Group", Date == "2017-09") %>%  
group_by(Sex) %>%  
summarise(mean_pop = mean(Population, na.rm = TRUE),  
          median_pop = median(Population, na.rm = TRUE),  
          max_population = max(Population, na.rm = TRUE))
```

```
## # A tibble: 2 × 4
```

```
##   Sex      mean_pop median_pop max_population
```

```
##   <chr>      <dbl>      <dbl>         <dbl>
```

```
## 1 Female    184.         98          536
```

```
## 2 Male     513         463         2002
```

# Importing, transforming, and summarising your data

```
grouped_crime <- group_by(crime, sex, victim_crime)
summarise(grouped_crime,
           state_anxiety = mean(SA),
           sd_SA = sd(SA),
           var_SA = var(SA))
```

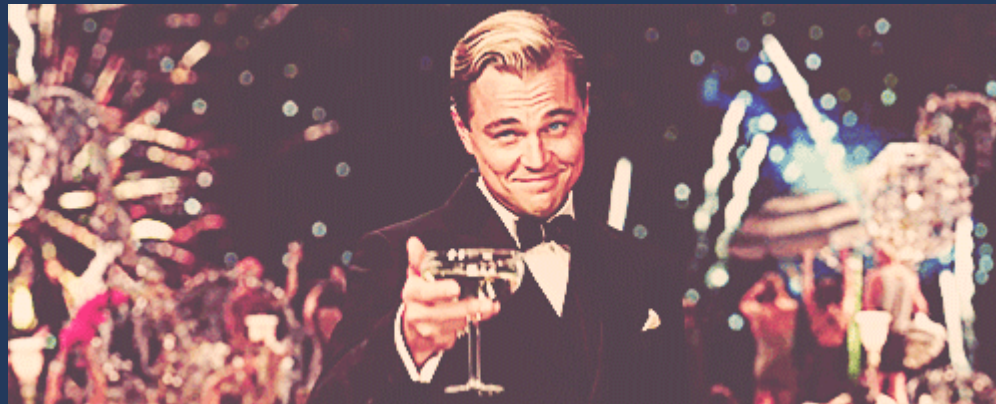
```
## `summarise()` has grouped output by 'sex'. You can override using the `.groups`
## argument.
```

```
## # A tibble: 4 × 5
## # Groups:   sex [2]
##   sex      victim_crime state_anxiety sd_SA var_SA
##   <chr>    <chr>           <dbl> <dbl> <dbl>
## 1 female no             1.90  0.518  0.268
## 2 female yes            1.98  0.643  0.413
## 3 male   no             2.02  0.553  0.306
## 4 male   yes            1.74  0.472  0.223
```

# Things we've covered

1. How to structure data through data.frames
2. How to handle different types of data (characters, numbers, logicals)
3. How to import different types of data
4. How to visualize your data
5. How to select only the data you need
6. How to summarise data

You've come a long way!



# Some R (and beyond) tips



*The internet will make those bad words go away*



*Essential*

## Googling the Error Message

O RLY?

*The Practical Developer*  
@ThePracticalDev

*How to actually learn any new programming concept*



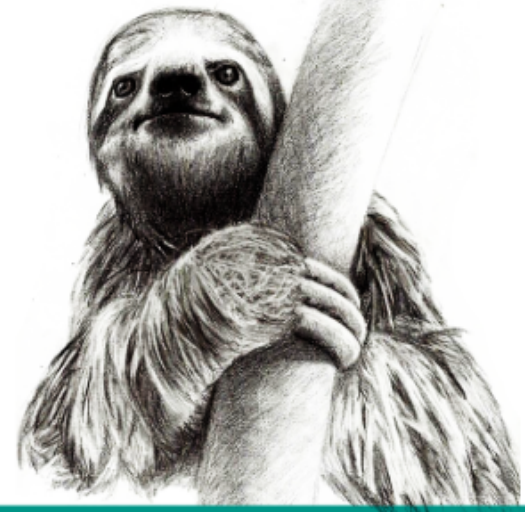
*Essential*

## Changing Stuff and Seeing What Happens

O RLY?

@ThePracticalDev

*Cutting corners to meet arbitrary management deadlines*



*Essential*

## Copying and Pasting from Stack Overflow

O RLY?

*The Practical Developer*  
@ThePracticalDev

# Google



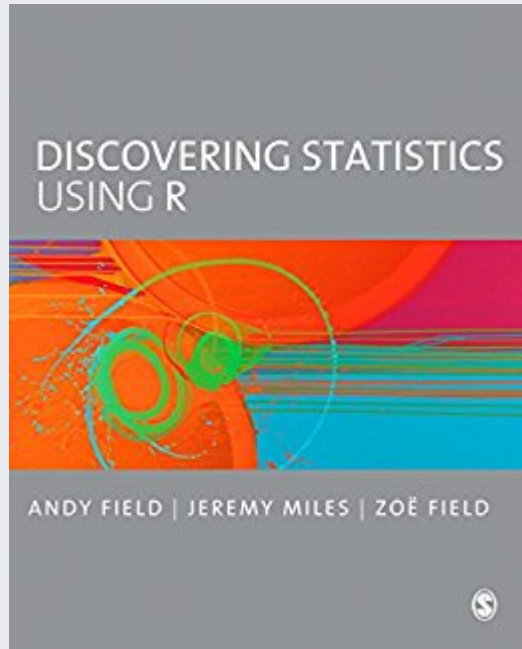
Google Search

I'm Feeling Lucky



Introducing [Google Home Hub](#). Now in stock.

# Some notes about Discovering Statistics Using R



1. Great as a *reference* book for stats. If you want to know more about the underlying maths or rationale of a given test, you'll find it in here.
2. If you don't know how to do a particular test, or what test you need, find it in here.
3. The coding style I've taught you differs. Both styles are fine.
4. There is no need to read it cover to cover.

# Don't despair: be patient!



**Hadley Wickham**

@hadleywickham · [Follow](#)



Just spent 5 minutes staring at this code trying to figure out what was wrong 🤔

```
all_units <- c(  
  setNames(powers, pa  
  setNames(powers. pa  
)
```

6:03 PM · Oct 27, 2019



1.3K



Reply



Copy link

[Read 104 replies](#)

# Basic Research Design and Null Hypothesis Significance Testing

# Research questions

In research, you typically start with a question:

- Do people find object recognition easier if the picture is in colour or in black and white?
- Is the blue pill or red pill better for treating a cold?
- Are people who have been a victim of crime more afraid of crime?

How do we measure the phenomena we talk about and compare them across groups of people?

# Operationalizing your variables

We don't typically have direct access to the *underlying, psychological* phenomena. Thus, we need to work how to measure them.

For example, Ellis & Renouf (2018) used questionnaires to assess people's personality traits and fear of crime.

```
head(select(crime, H:O), 6)
```

```
## # A tibble: 6 × 6
##       H     E     X     A     C     O
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1   3.7    3    3.4   3.9   3.2   3.6
## 2   2.5   3.1   2.5   2.4   2.2   3.1
## 3   2.6   3.1   3.3   3.1   4.3   2.8
## 4   3.5   1.8   3.3   3.4   2.1   2.7
## 5   3.3   3.4   3.9   3.2   2.8   3.9
## 6   2.6   2.6    3    2.6   2.9   3.4
```

```
head(select(crime, FoC, Foc2), 6)
```

```
## # A tibble: 6 × 2
##       FoC  Foc2
##   <dbl> <dbl>
## 1   2.6    3
## 2    2    3
## 3   1.2    2
## 4   3.2    5
## 5   2.8    3
## 6    2    4
```

# The Fear of Crime dataset

Some of the fundamental research questions for the Fear of Crime experiment Ellis & Renouf (2018) were:

1. Do men and women differ in terms of their fear of crime?
2. Are people who have been a victim of crime more fearful of crime?

```
select(crime, sex, victim_crime, FoC)
```

```
## # A tibble: 301 × 3
##   sex      victim_crime  FoC
##   <chr>    <chr>          <dbl>
## 1 male    yes            2.6
## 2 female  no              2
## 3 male    yes            1.2
## 4 male    no             3.2
## 5 female  no             2.8
## 6 female  no              2
## 7 male    yes            1.6
## 8 female  no              2
## 9 female  yes            3.4
## 10 male   no             1.4
## # ... with 291 more rows
```



# Null Hypothesis Significance Testing (NHST)

*Null Hypothesis Significance Testing* is a statistical framework for answering these types of questions.

Our *hypothesis* is that people who have been a victim of crime will be more afraid of it, so we ask:

- "Are people who have been a victim of crime more fearful of crime?"

What we test is how likely it would be to get the data we have *if the null hypothesis were true*:

- If people who **have not** been a victim of crime are as fearful of crime as people who **have**, how likely are the results we have obtained?

# The $t$ -test family

The statistical test of choice for comparing two groups is the two sample  $t$ -test.

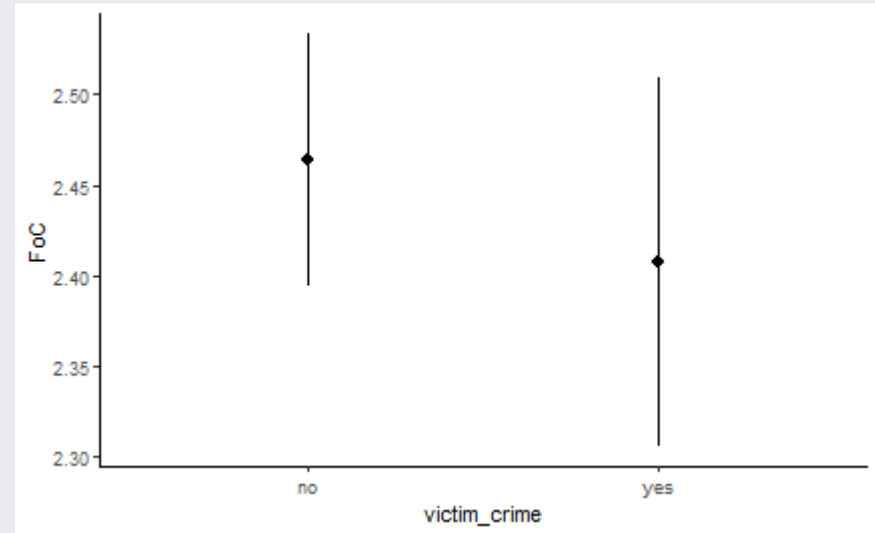
## 1. Independent $t$ -tests

- Compare means across different groups (e.g. groups of people)
- Also called between-subjects

## 2. Paired $t$ -tests

- Compares means across related data (e.g. data from the same people measured twice)
- Also called repeated-measures

We use the `t.test()` function for all of these!



Independent samples

# Performing *t*-tests in R

Let's test whether the mean Fear of Crime differs between victims of crime and non-victims.

Victims and non-victims are two independent groups, so we need an independent (or two-sample *t*-test).

```
t.test(FoC ~ victim_crime,  
       data = crime,  
       paired = FALSE)
```

```
##  
##      Welch Two Sample t-test  
##  
## data:  FoC by victim_crime  
## t = 0.45309, df = 197.48, p-value = 0.651  
## alternative hypothesis: true difference in means between group no and group yes is not equal to 0  
## 95 percent confidence interval:  
##  -0.1873001  0.2990388  
## sample estimates:  
##  mean in group no mean in group yes  
##      2.463636      2.407767
```

# Performing *t*-tests in R

The tilde (~) symbol in R usually means "modelled by". The dependent variable goes before "~".

FoC ~ victim\_crime means FoC modelled by victim\_crime.

data = crime tells R to look in the crime data frame for the data.

paired = FALSE tells R that this is an *independent samples* test.

```
t.test(FoC ~ victim_crime,  
       data = crime,  
       paired = FALSE)
```

```
##  
##      Welch Two Sample t-test  
##  
## data:  FoC by victim_crime  
## t = 0.45309, df = 197.48, p-value = 0.651  
## alternative hypothesis: true difference in means  
## 95 percent confidence interval:  
##  -0.1873001  0.2990388  
## sample estimates:  
##  mean in group no mean in group yes  
##           2.463636           2.407767
```

# Performing *t*-tests in R

The output shows the key statistics in one-line:

**"t = 0.453, df = 197.479, p-value = 0.651"**

A p-value is the probability of obtaining this difference between means if the null hypothesis is true.

Conventionally, we consider a p-value < .05 to be *significant*, allowing us to *reject* the null hypothesis.

```
##  
##      Welch Two Sample t-test  
##  
## data:  FoC by victim_crime  
## t = 0.45309, df = 197.48, p-value = 0.651  
## alternative hypothesis: true difference in means  
## 95 percent confidence interval:  
##  -0.1873001  0.2990388  
## sample estimates:  
##  mean in group no mean in group yes  
##           2.463636           2.407767
```

# Performing *t*-tests in R

There are also *sample estimates*, i.e. the group means, and confidence intervals of the *difference* between means.

Confidence intervals are a measure of *uncertainty*; the broader they are, the more uncertain about the accuracy of the values we have estimated.

Here, they overlap zero; the data is compatible with both negative and positive differences in fear of crime.

```
##  
##      Welch Two Sample t-test  
##  
## data:  FoC by victim_crime  
## t = 0.45309, df = 197.48, p-value = 0.651  
## alternative hypothesis: true difference in means  
## 95 percent confidence interval:  
##    -0.1873001    0.2990388  
## sample estimates:  
## mean in group no mean in group yes  
##          2.463636          2.407767
```

# Visualizing the results

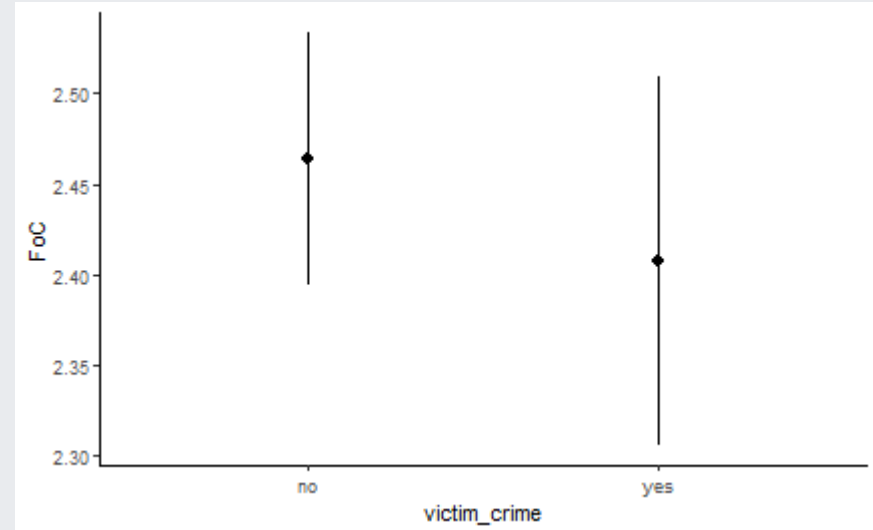
Here I plot a representation of the results using `ggplot()`

`stat_summary()` can be used to calculate and plot summary statistics.

By default, it calculates the *mean and standard error of the mean (SEM)*.

The SEM is another measure of uncertainty, providing a measure of how *accurately* we've estimated the mean.

```
ggplot(crime, aes(x = victim_crime, y = FoC)) +  
  stat_summary(fun.data = mean_se) +  
  theme_classic()
```





# Descriptive statistics

We can also calculate these summary statistics directly.

The SEM is the *standard deviation* divided by the *square root* - `sqrt()` - of the sample size.

$$\sigma^M = \frac{\sigma}{\sqrt{N}}$$

```
crime %>%  
  group_by(victim_crime) %>%  
  summarise(n = n(),  
            mu = mean(FoC),  
            std = sd(FoC),  
            sem = std/sqrt(n))
```

```
## # A tibble: 2 × 5  
##   victim_crime      n    mu   std   sem  
##   <chr>        <int> <dbl> <dbl> <dbl>  
## 1 no             198  2.46  0.980 0.0696  
## 2 yes            103  2.41  1.03  0.102
```

# Paired samples

# Performing *t*-tests in R

None of the measures in the `crime` dataset are *repeated* or *paired*, so for a moment we need to switch to another dataset.

`crossmod` is a dataset from a cognitive experiment in which participants identified objects using by touch. They did this twice. Sometimes the objects changed size; sometimes they stayed the same. The hypothesis was that changing size would slow down naming.

```
crossmod <- read_csv("data/crossmod.csv")
crossmod
```

```
## # A tibble: 48 × 3
##   participant Size      RT
##         <dbl> <chr> <dbl>
## 1             1 DS    2608.
## 2             1 SS    2195
## 3             2 DS    2551
## 4             2 SS    2213
## 5             3 DS    2900.
## 6             3 SS    2788.
## 7             4 DS    2646.
## 8             4 SS    2390.
## 9             5 DS    3486.
## 10            5 SS    2844.
## # ... with 38 more rows
```

# Performing *t*-tests in R

Since the data contains repeated measurements from the same participants, we need to run a paired/repeated-measures *t*-test. So we simply change `paired = FALSE` to `paired = TRUE`!

```
t.test(RT ~ Size, data = crossmod, paired = TRUE)
```

```
##
##      Paired t-test
##
## data:  RT by Size
## t = 3.512, df = 23, p-value = 0.001872
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##    99.91035 386.29799
## sample estimates:
## mean difference
##      243.1042
```

This time, there *was* a significant difference!

# Checking assumptions

# Assumptions of the $t$ -test

Before we get too carried away with our results, we have to check our assumptions!

1. The dependent variable must be continuous (and *interval* or *ratio*).
2. The independent variable must be categorical.
3. The distribution of the data should be approximately normal.

## Independent $t$ -tests have a couple of extra assumptions:

1. The data in each sample should be *independent* (i.e. come from different people)
2. The variance of the data in each group should be the same (*Homogeneity of variance*).

# Levels of measurement





# MEN'S 100M



RESULT - FINAL



WIND +1.5M/S

1	JAM		USAIN BOLT	OR 9.63
2	JAM		YOHAN BLAKE	9.75
3	USA		JUSTIN GATLIN	9.79
4	USA		TYSON GAY	9.80
5	USA		RYAN BAILEY	9.88
6	NED		CHURANDY MARTINA	9.94
7	TRI		RICHARD THOMPSON	9.98
8	JAM		ASAFA POWELL	11.99

Ω OMEGA




# Levels of measurement











## Nominal data

Nominal data is categorical data that has no natural order.

For example, the runners' names (e.g. Usain Bolt, Asafa Powell, Tyson Gay) and nationalities (e.g. Jamaica, USA) are **nominal**.



A screenshot of a sports broadcast showing the final results of the Men's 100m race. The header includes a runner icon, the event name 'MEN'S 100M', the subtitle 'RESULT - FINAL', and the Olympic rings logo. A wind speed indicator shows 'WIND +1.5M/S'. The results are listed in a table with columns for rank, nationality (with flag), name, and time. Usain Bolt is the winner with a time of 9.63 seconds, marked as a world record (WR). Asafa Powell is in 8th place with a time of 11.99 seconds.


		<b>MEN'S 100M</b>	
		<b>RESULT - FINAL</b>	 WIND +1.5M/S
1	JAM 	USAIN BOLT	WR 9.63
2	JAM 	YOHAN BLAKE	9.75
3	USA 	JUSTIN GATLIN	9.79
4	USA 	TYSON GAY	9.80
5	USA 	RYAN BAILEY	9.88
6	NED 	CHURANDY MARTINA	9.94
7	TRI 	RICHARD THOMPSON	9.98
8	JAM 	ASAFA POWELL	11.99

# Levels of measurement

## Ordinal data

Ordinal data is also categorical, but is *ordered*. The gaps between the categories are not necessarily equal.

e.g. finishing position is ordinal, but the gap between first and second is bigger than the gap between second and third!



A screenshot of a digital display for the Men's 100m final race. It shows a list of eight athletes with their finishing positions, country flags, names, and times. The background is dark blue with white and yellow text. The Olympic rings logo is in the top right corner. The wind speed is indicated as +1.5M/S. The first three positions are highlighted with a red background.

MEN'S 100M		RESULT - FINAL		WIND +1.5M/S	
1	JAM	USAIN BOLT	OR	9.63	
2	JAM	YOHAN BLAKE		9.75	
3	USA	JUSTIN GATLIN		9.79	
4	USA	TYSON GAY		9.80	
5	USA	RYAN BAILEY		9.88	
6	NED	CHURANDY MARTINA		9.94	
7	TRI	RICHARD THOMPSON		9.98	
8	JAM	ASAFA POWELL		11.99	


# Levels of measurement

## Interval data

Interval data is data with equally spaced intervals. (e.g. the gap between 9 seconds and 10 seconds is the same as the gap between 12 seconds and 13 seconds)

## Ratio data

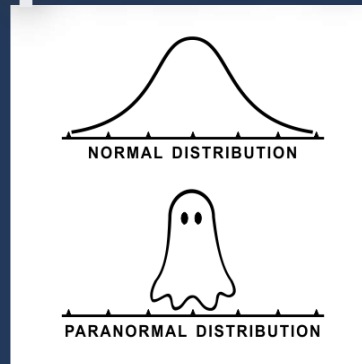
Ratio data is similar to interval data, but has a meaningful boundary at zero (e.g. a finishing time cannot be below zero.)



A screenshot of a sports broadcast showing the final results of the Men's 100m race. The header includes a runner icon, the event name 'MEN'S 100M', and the Olympic rings. Below the header, it says 'RESULT - FINAL' and 'WIND +1.5M/S'. The results are listed in a table with columns for rank, country, athlete name, and time. The background is dark blue with white and yellow text.

MEN'S 100M			
RESULT - FINAL			
WIND +1.5M/S			
1	JAM	USAIN BOLT	9.63
2	JAM	YOHAN BLAKE	9.75
3	USA	JUSTIN GATLIN	9.79
4	USA	TYSON GAY	9.80
5	USA	RYAN BAILEY	9.88
6	NED	CHURANDY MARTINA	9.94
7	TRI	RICHARD THOMPSON	9.98
8	JAM	ASAFA POWELL	11.99

# The assumption of *normality*

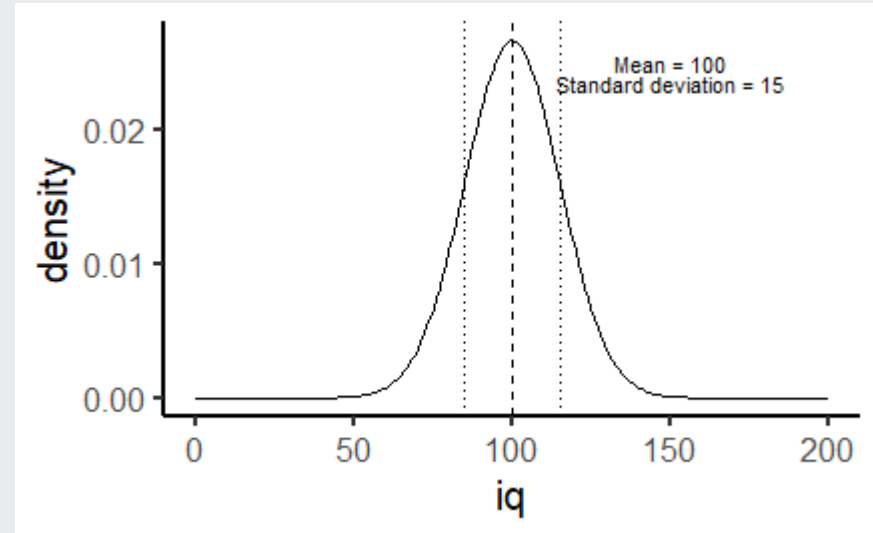


# The assumption of *normality*

A normal distribution can be easily described by two parameters: the mean -  $\mu$  - and the standard deviation -  $\sigma$ .

The normal distribution is symmetrical.

For example, Mean Intelligence Quotient is 100; the standard deviation is 15. Thus, it's easy to draw what the distribution of IQ should look like.



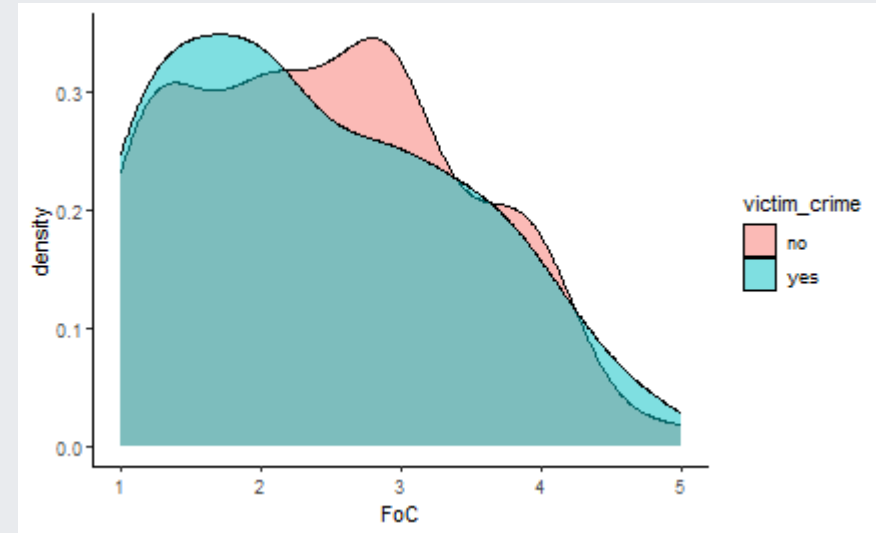
# Checking normality

Plotting our data is a simple way to check normality!

Our data are clearly *skewed* - more values are to the left than to the right.

```
ggplot(crime, aes(x = FoC,  
                  fill= victim_crime)) +  
  geom_density(alpha = 0.5) +  
  theme_classic()
```

If you need to test normality formally, use the Shapiro-Wilks test - **shapiro.test()**.

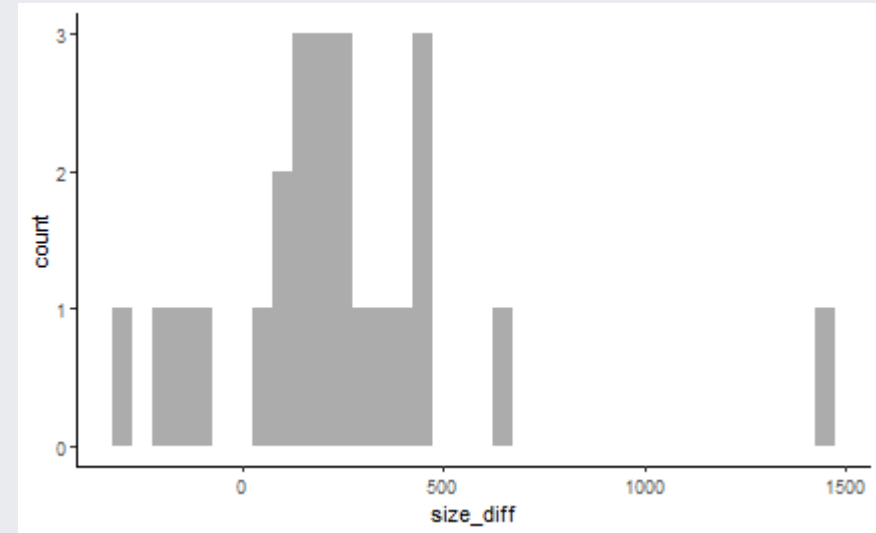


# Checking normality

With a repeated-measures design, we care about the normality of the *differences* between pairs.

```
crossmod_wide <-  
  crossmod %>%  
  spread(Size, RT) %>%  
  mutate(size_diff = DS - SS)
```

```
ggplot(crossmod_wide,  
       aes(x = size_diff)) +  
  geom_histogram(alpha = 0.5,  
                binwidth = 50) +  
  theme_classic()
```



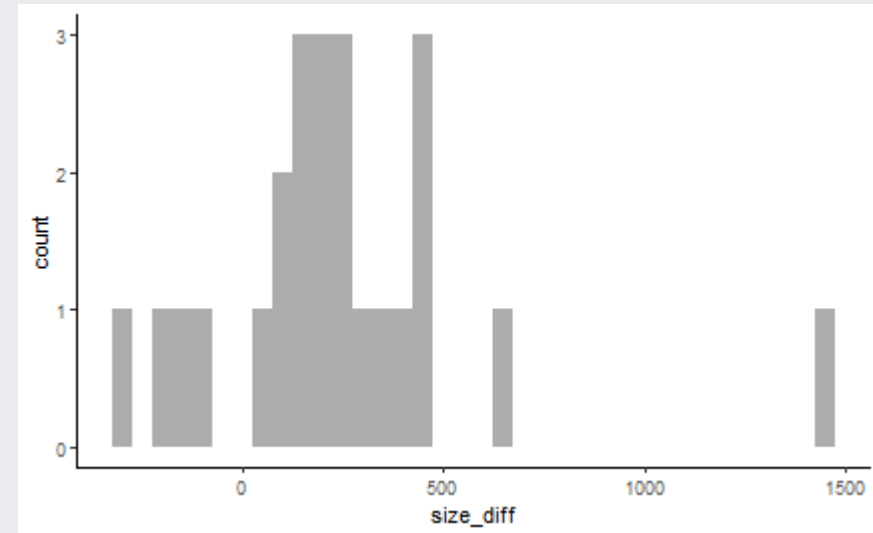
# Checking normality

This looks a little suspicious. There's probably an *outlier* at the far right side of the plot (more on these next session).

```
shapiro.test(crossmod_wide$size_diff)
```

```
##  
##      Shapiro-Wilk normality test  
##  
## data:  crossmod_wide$size_diff  
## W = 0.83796, p-value = 0.001318
```

The Shapiro-Wilks test is significant, suggesting that normality is violated.





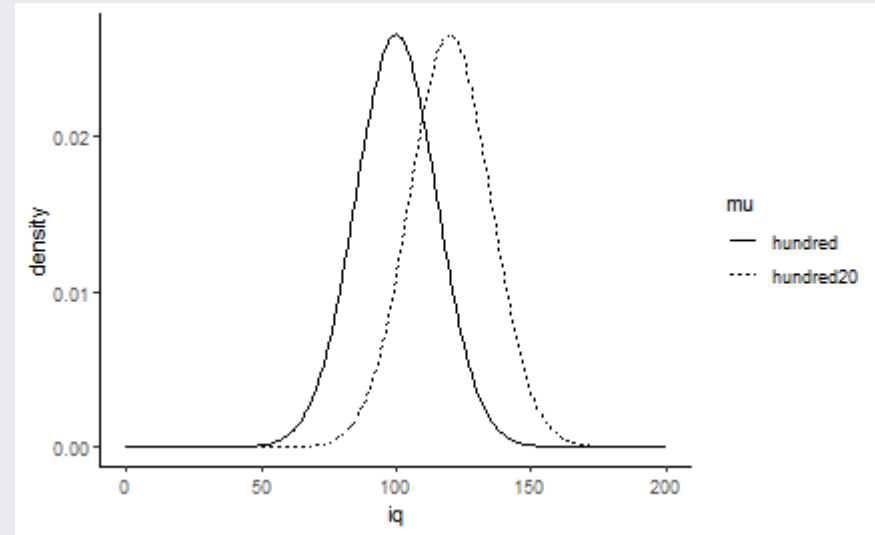
# Homogeneity of variance

# Homogeneity of variance

Remember how a normal distribution can be described by two parameters: the mean and the standard deviation.

As we change the mean, the distribution moves along the x-axis.

With a  $t$ -test we only want to pick up differences in the location of the *means* along the x-axis representing our dependent variable.



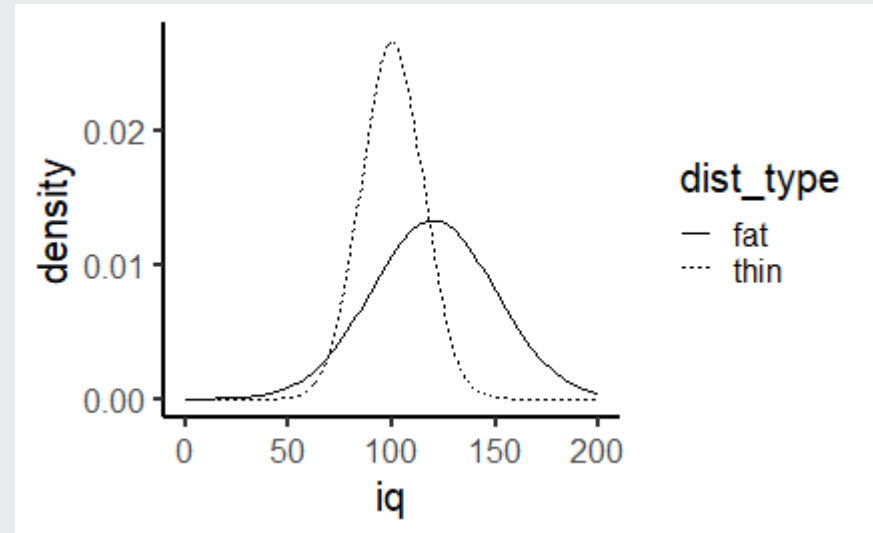
# Homogeneity of variance

Our ability to detect differences in the location of the means is hampered if the *standard deviation* (or the *variance*, which is  $sd^2$ ) differs across groups.

As the standard deviation increases, the variability around the mean increases, and the distribution of values gets *broader*.

Here, I doubled the standard deviation of the second distribution to 30.

Differences in variance across groups can *bias* your statistics in complex ways.



# Homogeneity of variance

Let's look again at our distributions for Fear of Crime.

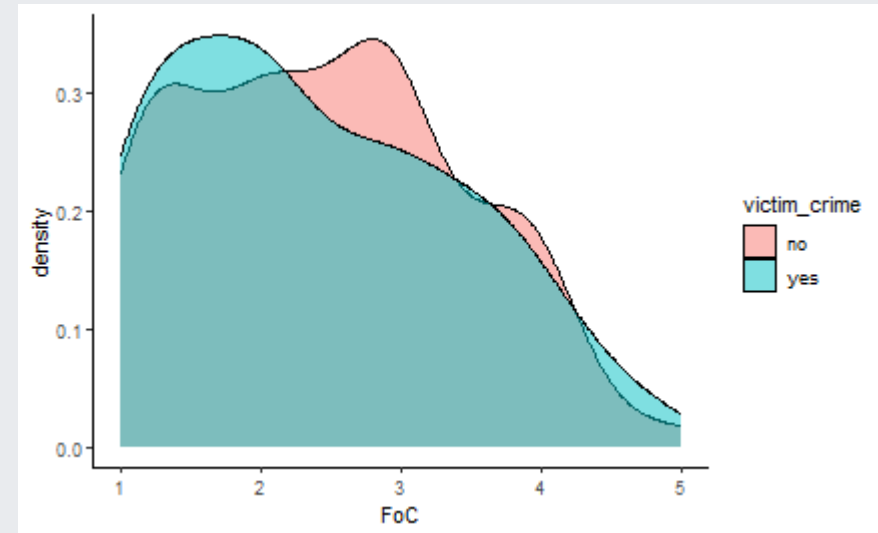
Although they're skewed, it looks like they're similarly variable.

We can test this statistically using Levene's test from the **car** package:

```
library(car)
leveneTest(FoC ~ victim_crime, data = crime)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value Pr(>F)
## group    1  0.3669 0.5451
##           299
```

It's not significant, consistent with our visual impressions.



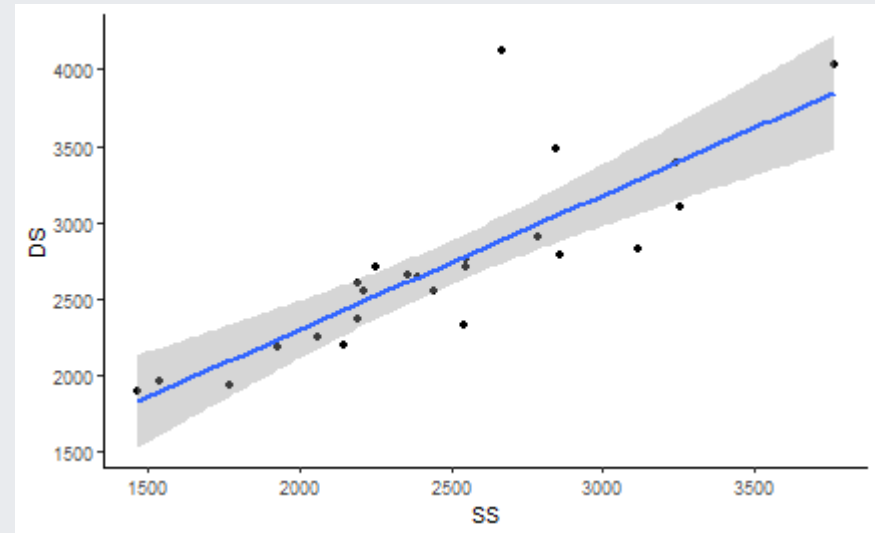
# The assumption of independence

From the `crossmod` dataset, RTs across the two conditions have a strong positive relationship.

Values generated by the same people on repeated occasions tend to be correlated.

This one is easy to assess: we know from the design whether the data is independent or not!

```
## `geom_smooth()` using formula = 'y ~ x'
```



**What to do about violated assumptions**

# What to do about violated assumptions

## 1. When homogeneity of variance is violated

- Welch's t-test does not assume equality/homogeneity of variance. By default, R uses Welch's t-test for independent samples.
- This is only a problem for independent samples t-tests.

## 2. When independence is violated

- Use a paired/repeated-measures t-test.

## 3. When normality is violated

- Often...nothing is done, or the data is *transformed* (we'll cover that in detail next term).
- Consider non-parametric tests

# Non-parametric t-tests

Consider using non-parametric statistics, which make fewer assumptions.

The most frequently used tests are the Wilcoxon rank-sum test (sometimes called the Mann-Whitney U test) for independent samples data, and the Wilcoxon signed-rank test for paired samples data.

Simply substitute `t.test()` for `wilcox.test()`!

```
wilcox.test(FoC ~ victim_crime, data = crime, paired = FALSE)
```

```
##  
##      Wilcoxon rank sum test with continuity correction  
##  
## data:  FoC by victim_crime  
## W = 10623, p-value = 0.5517  
## alternative hypothesis: true location shift is not equal to 0
```



# Reporting the results of a *t*-test

```
t.test(FoC ~ victim_crime, data = crime)
```

```
##
##      Welch Two Sample t-test
##
## data:  FoC by victim_crime
## t = 0.45309, df = 197.48, p-value = 0.651
## alternative hypothesis: true difference in means between group no and group yes is not equal to 0
## 95 percent confidence interval:
##  -0.1873001  0.2990388
## sample estimates:
##  mean in group no mean in group yes
##           2.463636           2.407767
```

```
crime %>%
  group_by(victim_crime) %>%
  summarise(means = mean(FoC),
            sem = sd(FoC) / sqrt(n()))
```

```
## # A tibble: 2 × 3
##   victim_crime means    sem
##   <fct>      <dbl> <dbl>
## 1 no         2.46 0.0696
## 2 yes        2.41 0.102
```

# Reporting the results of a *t*-test

"An independent sample *t*-test was conducted to test whether being victims (or not) victims of crime affect fear of crime. On average, participants who had been victims of crime did not have significantly higher Fear of Crime ( $M = 2.41, SE = .10$ ) than participants who had not ( $M = 2.46, SE = .07$ ),  $t(197.48) = .453, p = .651$ ."

Means and standard errors are typically reported to two decimal places.

*t*-values are usually reported to three decimal places.

Exact *p-values* should be reported down to three decimal places; if the *p*-value is below .001, report " $p < .001$ ".

Remember to always specify, *somewhere*, what type of *t*-test you used.

(for further guidance, see Field et al., Discovering statistics using R)

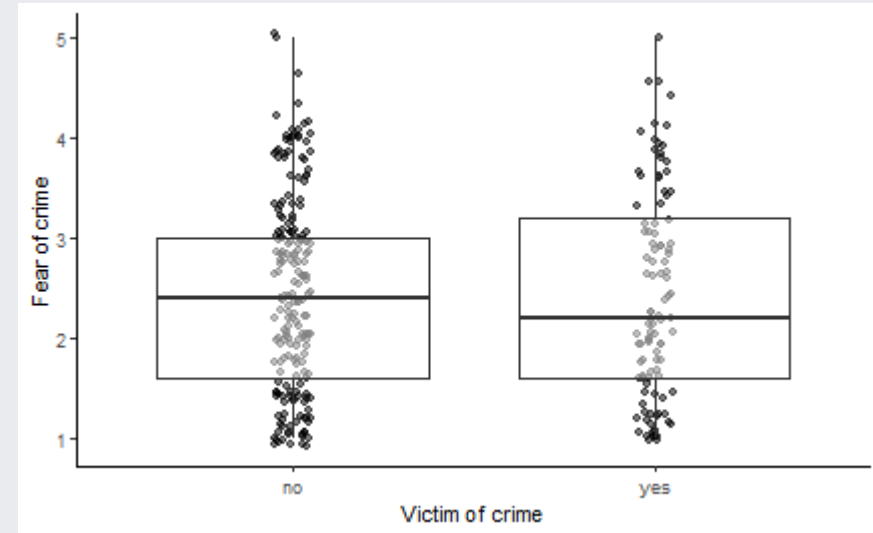
# A plotting suggestion

One way to compare distributions is graphically.

Here we plot the data from each sample using boxplots, with individual data overlaid as points.

Each point is the score for an individual.

```
crime %>%  
  ggplot(aes(x = victim_crime, y = FoC)) +  
  geom_jitter(width = 0.05, alpha = 0.5) +  
  geom_boxplot(alpha = 0.5) +  
  theme_classic() +  
  labs(y = "Fear of crime", x = "Victim of
```



# Next session

Next week we'll look into **regression** and **correlation**.

Chapters 6 (Correlation) and 7 (Regression) of Discovering Statistics Using R.

## Reminder – don't feel you have to read every word!

Look at the introductory sections of each chapter, refer back to the rest *as necessary*.