UNIVERSITY OF LINCOLN

The basics of statistical testing

PSY9219M - Research Methods and Skills

Dr Matt Craddock

30/10/2018

The five verbs of dplyr

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

Importing, transforming, and summarising your data

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

```
## # A tibble: 301 x 15
##    Participant sex     age victim_crime     H     E     X     A     C     O
##    <chr>       <chr> <dbl> <chr>        <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 R_01TjXgC1~ male     55 yes            3.7   3     3.4   3.9   3.2   3.6
##  2 R_0dN5YeUL~ fema~    20 no             2.5   3.1   2.5   2.4   2.2   3.1
##  3 R_0DPiPYWh~ male     57 yes            2.6   3.1   3.3   3.1   4.3   2.8
##  4 R_0f7bSsH6~ male     19 no             3.5   1.8   3.3   3.4   2.1   2.7
##  5 R_0rov2RoS~ fema~    20 no             3.3   3.4   3.9   3.2   2.8   3.9
##  6 R_0wioqGER~ fema~    20 no             2.6   2.6   3     2.6   2.9   3.4
##  7 R_0wRO8lNe~ male     34 yes            3.2   2.5   3.2   2.8   4     3.2
##  8 R_116nEdFs~ fema~    19 no             2.9   4     3.9   4.2   3.7   1.9
##  9 R_11ZmBd5V~ fema~    19 yes            3.4   3.4   3.3   3.4   3.2   3.2
## 10 R_12i26Qzo~ male     20 no             2.4   2.1   1.8   2.2   3.4   2.9
## # ... with 291 more rows, and 5 more variables: SA <dbl>, TA <dbl>,
## #   OHQ <dbl>, FoC <dbl>, Foc2 <dbl>
```

Importing, transforming, and summarising your data

```
library(readxl)
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 x 4
##   Sex    mean_pop median_pop max_population
##   <chr>     <dbl>      <dbl>          <dbl>
## 1 Female     184.         98            536
## 2 Male       513.       462.           2082
```

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))
```

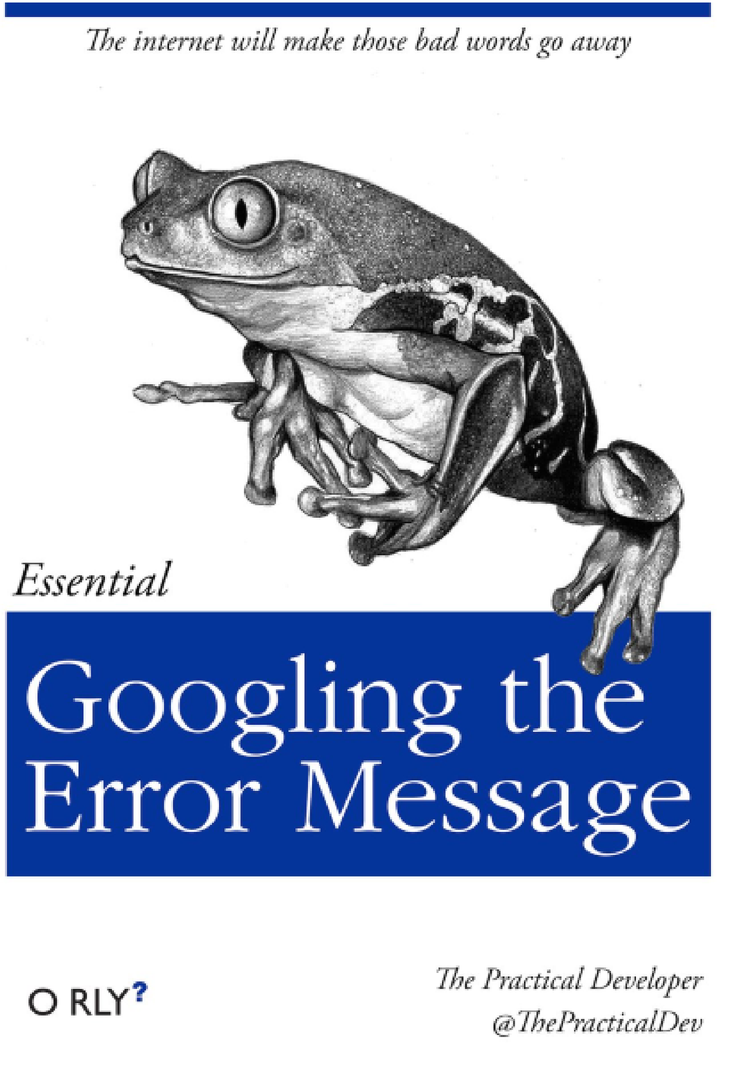
```
## # A tibble: 4 x 5
## # Groups:   sex [2]
##   sex    victim_crime state_anxiety sd_SA var_SA
##   <chr>  <chr>                <dbl> <dbl>  <dbl>
## 1 female no                    1.90 0.518  0.268
## 2 female yes                   1.98 0.643  0.413
## 3 male   no                    2.02 0.553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

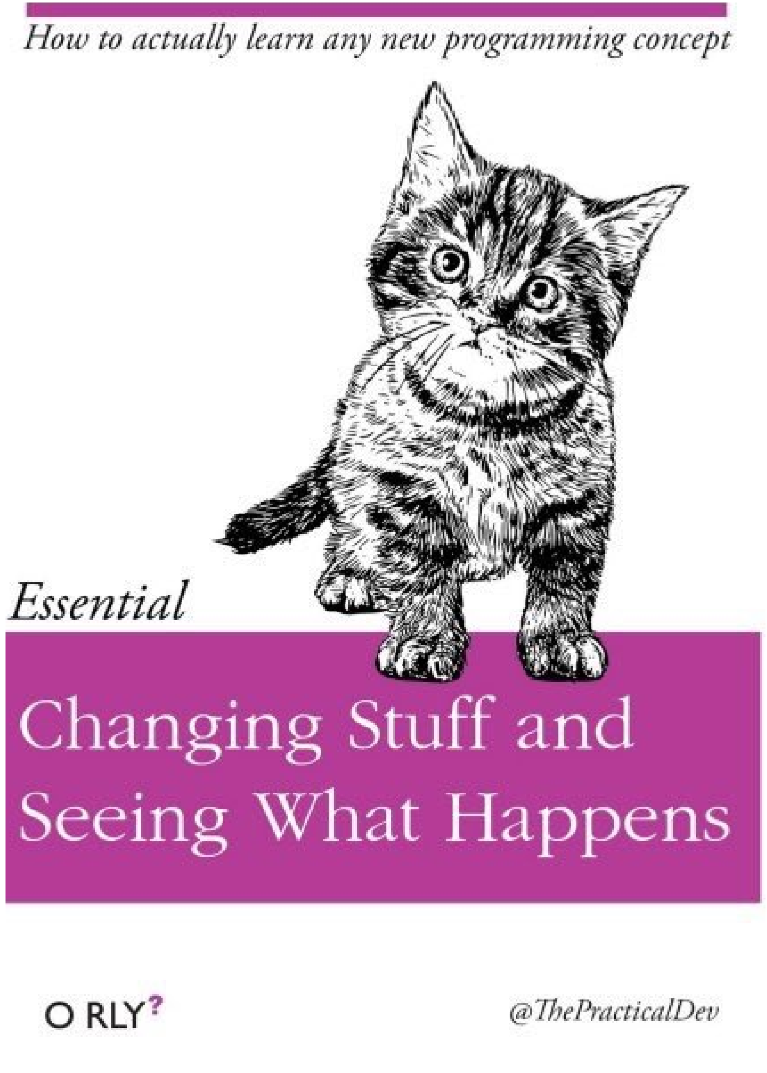
Things we've covered

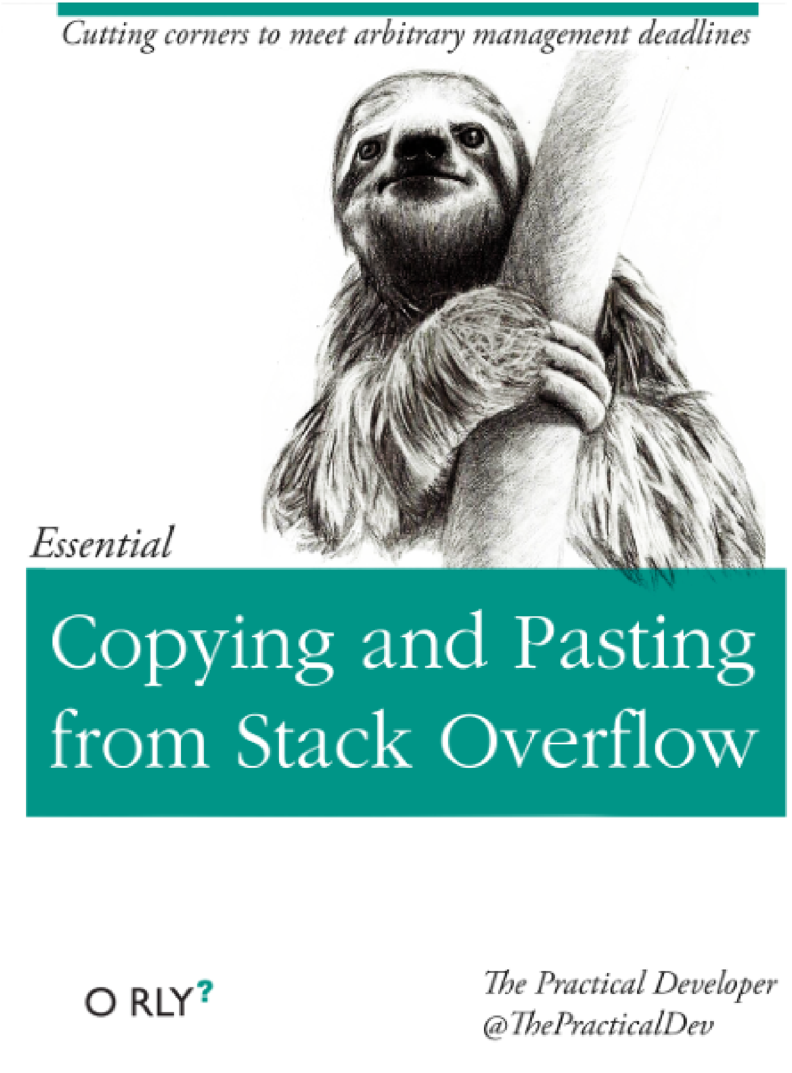
1. How to structure data through data.frames
2. How to handle different types of data (characters, numbers, logicals)
3. How to import different types of data
4. How to visualize your data
5. How to select only the data you need
6. How to summarise data

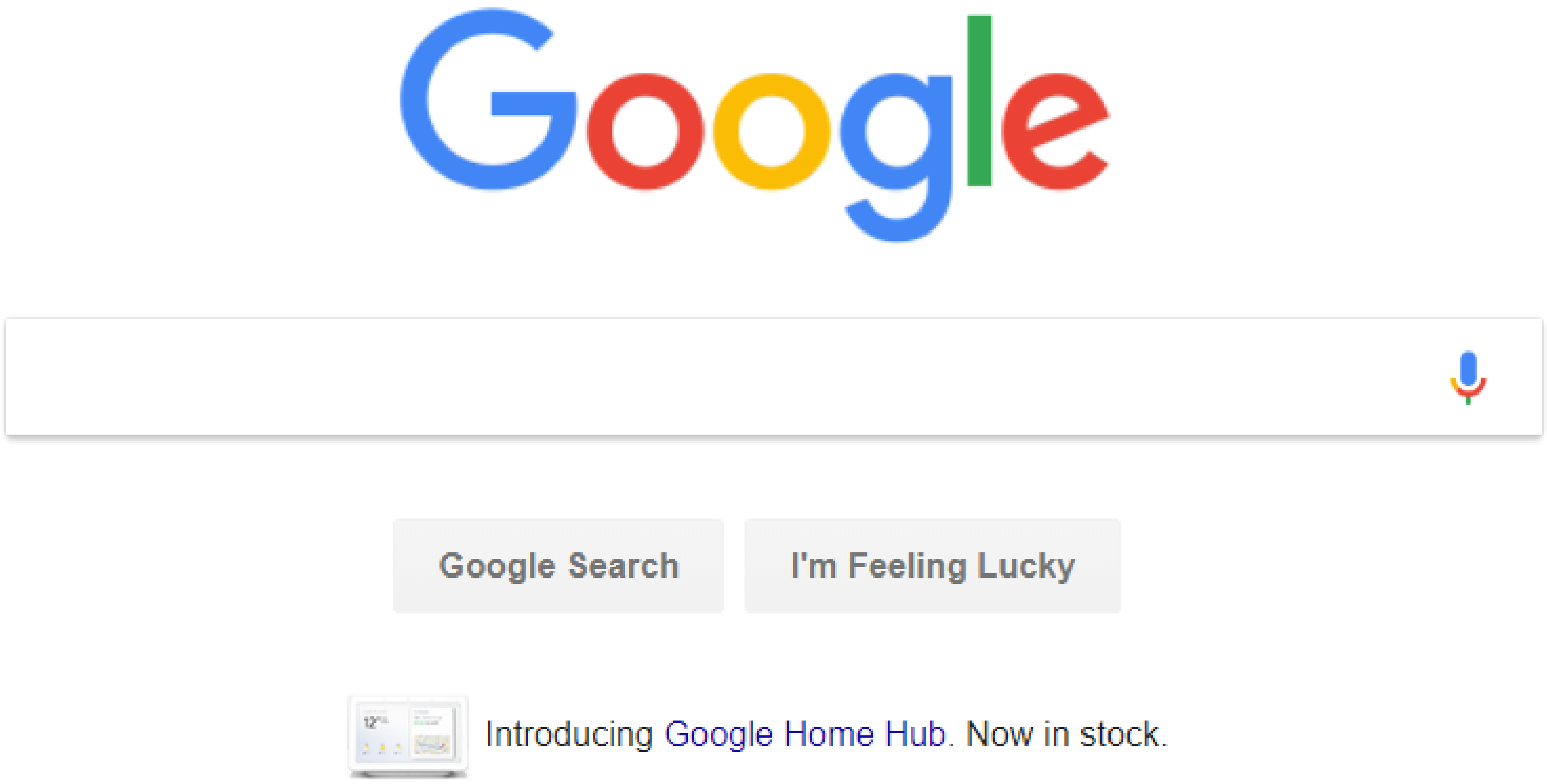
You've come a long way!

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

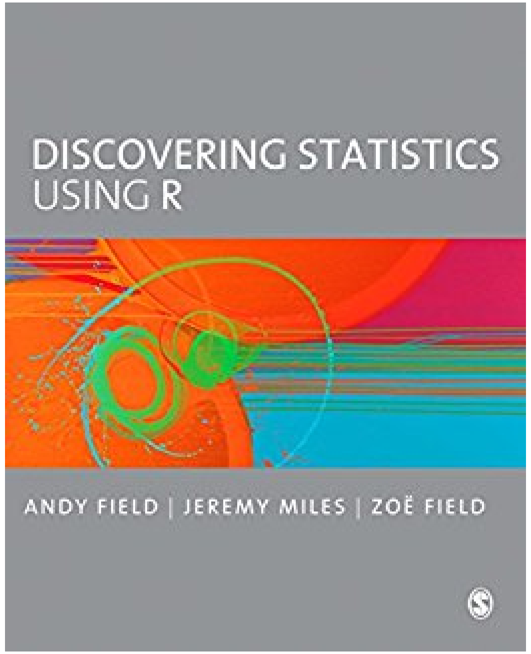

How to actually learn any new programming concept
Essential
Changing Stuff and Seeing What Happens
O RLY?
@ThePracticalDev


Cutting corners to meet arbitrary management deadlines
Essential
Copying and Pasting from Stack Overflow
O RLY?
The Practical Developer
@ThePracticalDev



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.

The who, what and how of research

Research questions

In research, you typically want to ask questions like

- 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 with higher anxiety more likely to have depression?

How do we measure, compare, and assess the phenomena we talk about?

Operationalizing your variables

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

For example, Ellis & Renouf (2018) used scales like the HEXACO-PI-R, or the State-Trait Anxiety Inventory.

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

```
## # A tibble: 6 x 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 x 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?

So who did they collect data from?

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

```
## # A tibble: 301 x 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
```

Populations and samples

A *population* is all the members of a defined group (e.g. all people with anxiety-related disorders; all black and white objects).

For the question "Are people who have been a victim of crime more fearful of crime than people who have not been a victim of crime?", you have two potential *populations*: people who have and people who have not been a victim of crime.

We rarely have the capability to get statistics for an entire population. Thus, we draw *samples* and try to estimate *population* values.

```
crime %>%
  group_by(victim_crime) %>%
  summarise(n())
```

```
## # A tibble: 2 x 2
##   victim_crime `n()`
##   <chr>        <int>
## 1 no             198
## 2 yes            103
```

Populations and samples

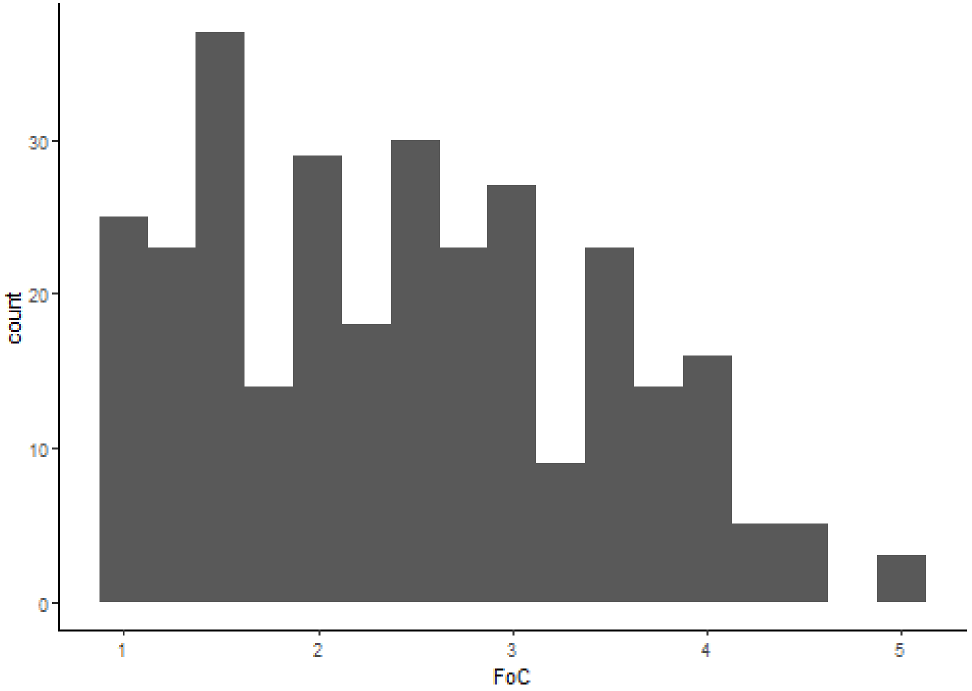
Our question was not just about whether there are people who have and people who haven't been victims of crime.

Our dependent variable is the *fear of crime*.

A starting question would be what are the potential values of the *fear of crime* if the population was simply *everyone*?

We can get an idea by plotting the *sampling distribution* - the distribution of values for our entire sample.

```
ggplot(crime, aes(x = FoC)) +
  geom_histogram(binwidth = 0.25) + theme_cla
```



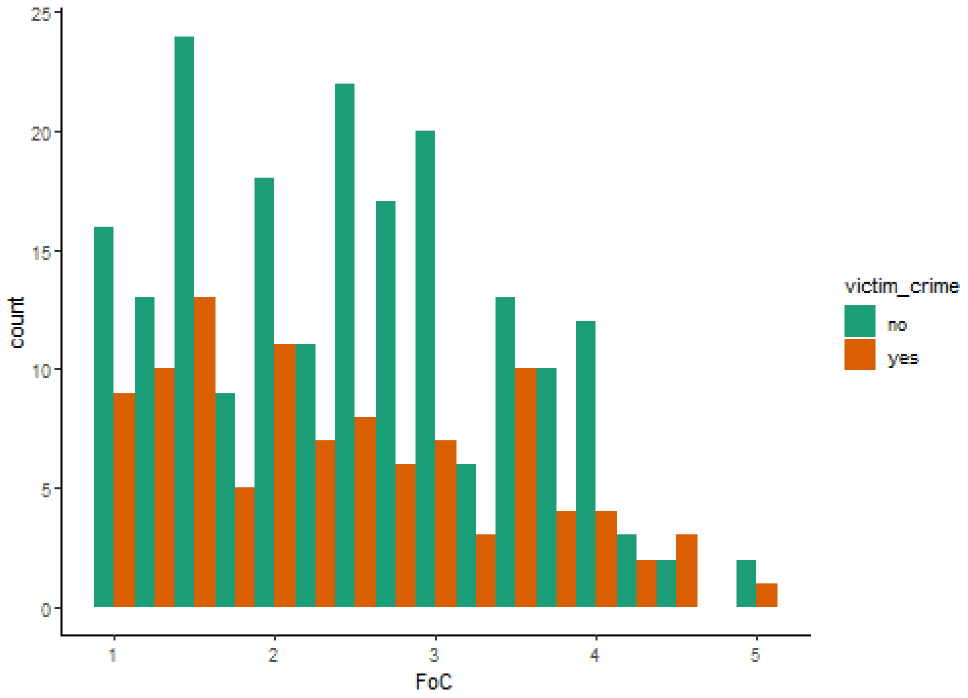
Populations and samples

Our next question is: does the data we have collected support the idea that victims of crime are a separate population from non-victims, with respect to *fear of crime*?

As we can see, the distributions of our two samples overlap substantially.

This suggests that, at least with respect to *fear of crime*, all our participants may be part of *one population*. But we need to follow this up with statistics!

```
ggplot(crime, aes(x = FoC, fill = victim_crim
  geom_histogram(binwidth = 0.25,
                 position = "dodge") +
  scale_fill_brewer(palette = "Dark2") +
  theme_classic()
```



Comparing two samples

Descriptive statistics

We want to estimate what values are *typical* or *characteristic* of a given population from samples.

Then we want to be able to distinguish between different *populations*.

There are a range of different statistics we can calculate that will help describe our samples.

The most obvious statistic to calculate is the *mean*, and its close friend, the *standard deviation* - **sd()**.

```
crime %>%
  group_by(victim_crime) %>%
  summarise(n(),
            mean(FoC),
            std.dev = sd(FoC))
```

```
## # A tibble: 2 x 4
##   victim_crime `n()` `mean(FoC)` std.dev
##   <chr>        <int>       <dbl>   <dbl>
## 1 no             198        2.46   0.980
## 2 yes            103        2.41   1.03
```

Descriptive statistics

The mean and standard deviation describe the characteristics of our samples.

But remember that these are samples from potential populations. So one question is how precisely our sample captures the true mean value of the *population*.

For that, we need the *standard error of the mean* (or SEM).

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

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

```
## # A tibble: 2 x 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
```

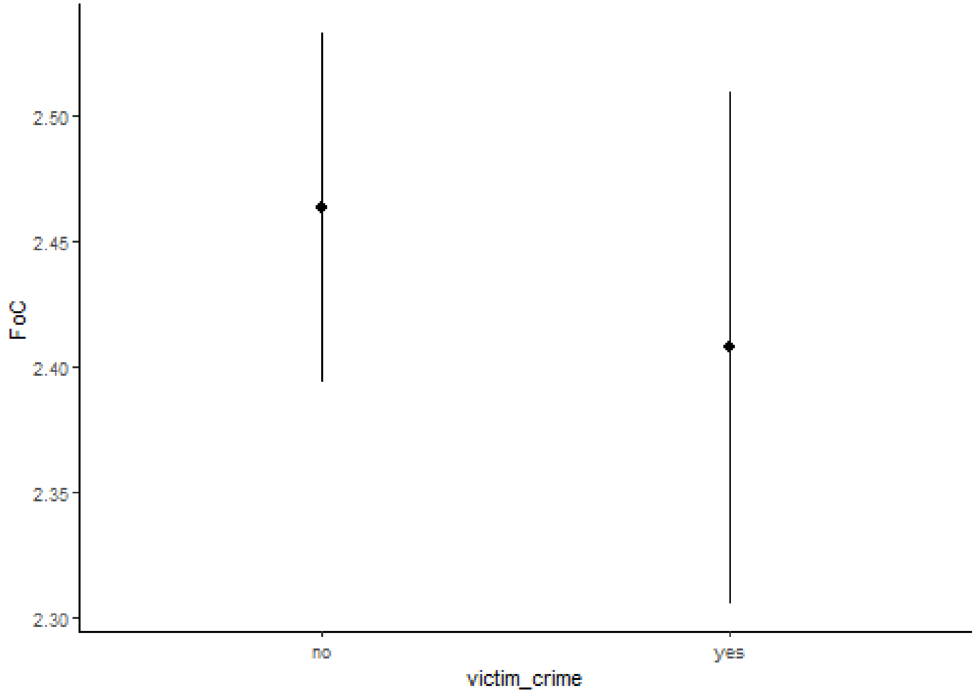
Descriptive statistics

The SEM gives us a measure of how well we've estimated the population mean from our sample. An easy way to visualize this is with error bars.

Here I add a **stat_summary()** to a **ggplot()**.

The large overlap between the error bars tells us that, at least with these samples, we have little evidence that there is a difference in Fear of Crime between these two groups.

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



Null Hypothesis Significance Testing

Null Hypothesis Significance Testing (NHST)

Think back to our original questions:

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?

The basis of NHST is to phrase these questions as:

If there is only one population, how likely is it that our two samples have values this different from each other?

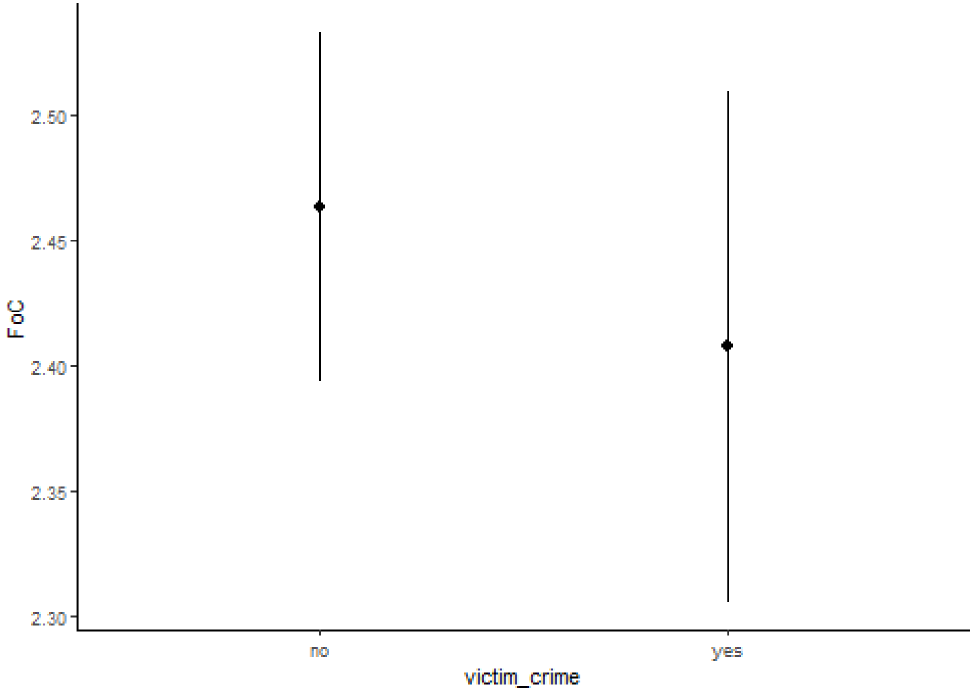
We answer this question with *test statistics* and *p-values*.

The *t*-test family

The statistical test of choice for comparing two means is the *t*-test.

1. One-sample t-tests
   - Compares one sample of means to a specific value (e.g. 0)
2. Independent t-tests
   - Compares means across different groups (e.g. groups of people)
3. Paired t-tests
   - Compares means across related data (e.g. data from the same people measured twice)

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



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).

```r
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 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"

`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.

```r
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 i
## 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 = 0.46069, df = 299, p-value = 0.6454

The p-value is above .05, so there was no significant difference in fear of crime between people who were victims and people who were not.

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

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

```
## 
##      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 i
## 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

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 x 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 difference in means is not equal to 0
## 95 percent confidence interval:
##   99.91035 386.29799
## sample estimates:
## mean of the differences
##                243.1042
```

Checking assumptions

Assumptions of the *t*-test

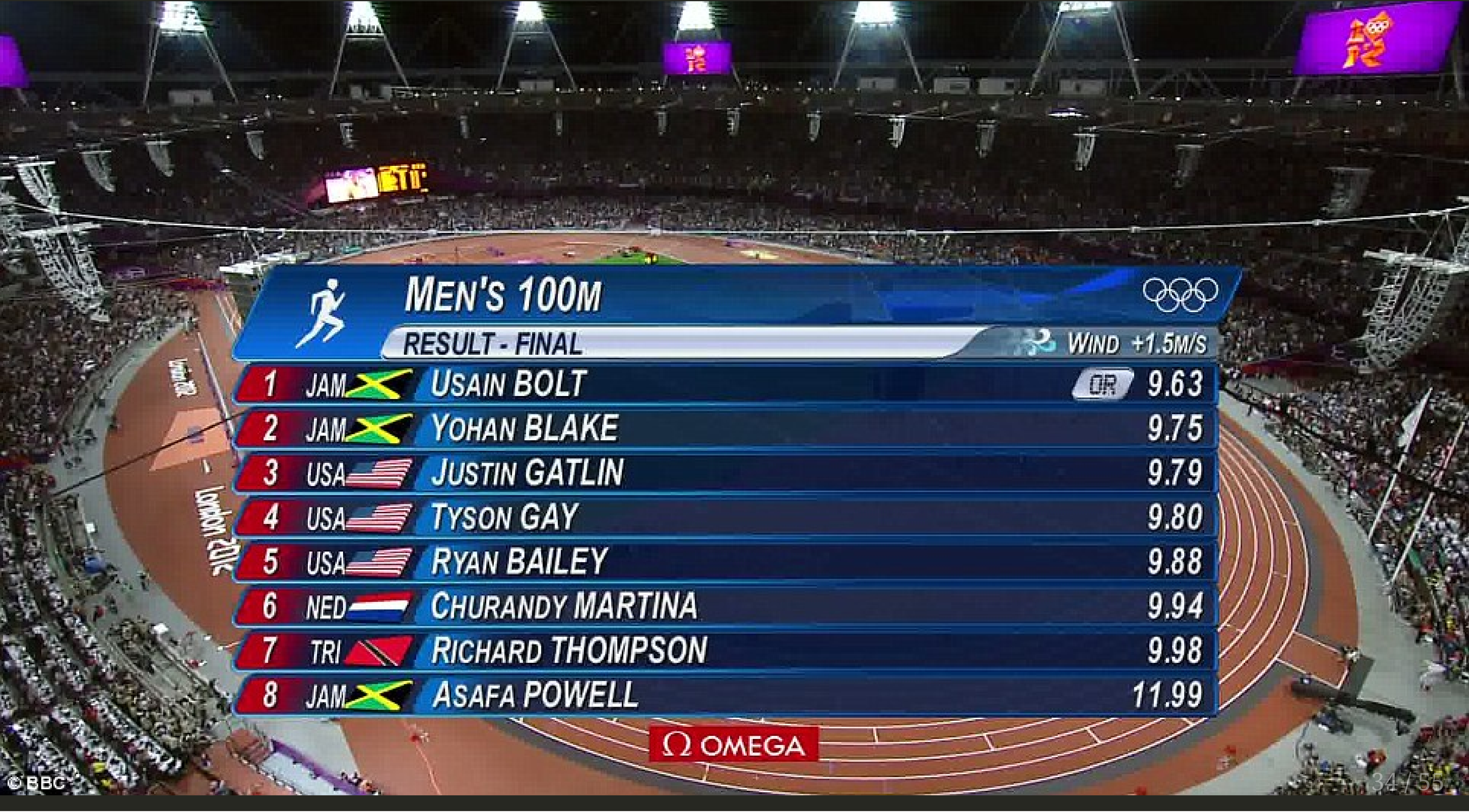
Before we get too carried away with our results, we have to check our assumptions!

1. The dependent variable must be continuous (and *interval* or *ratio*).
2. The independent variable must be categorical.
3. The distribution of data for each sample 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

OMEGA
London 2012

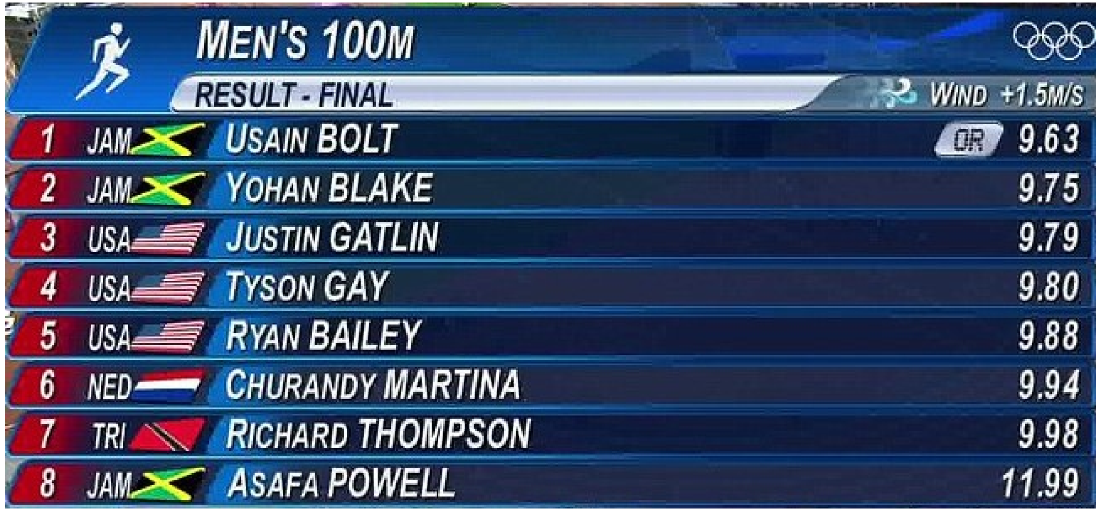
© BBC

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**.



MEN'S 100M

RESULT - FINAL

WIND +1.5M/S

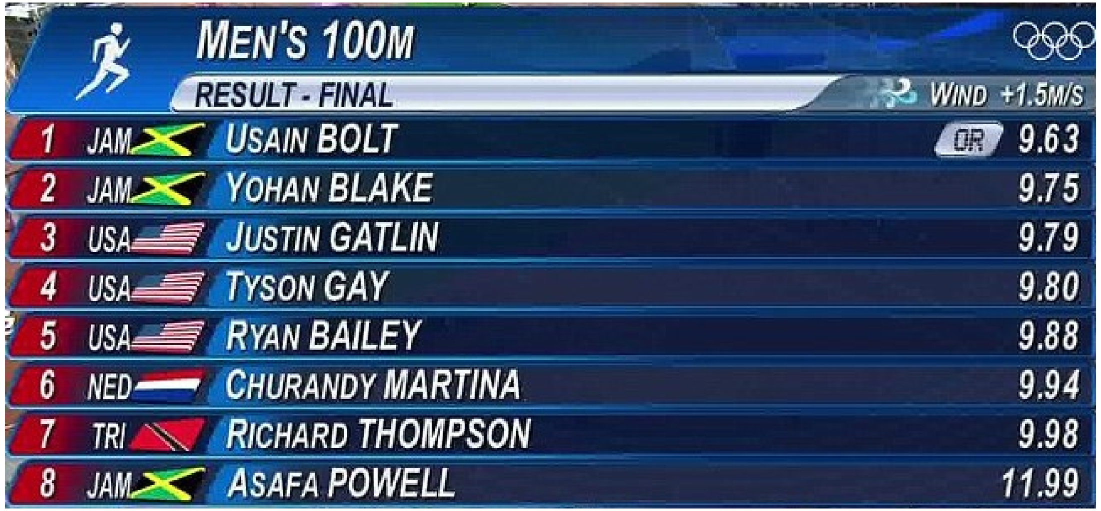
1	JAM	USAIN BOLT	OR	9.63
2	JAM	YOHAN BLAKE		9.75
3	USA	JUSTIN GATLIN		9.79
4	USA	TYSON GAY		9.80
5	USA	RYAN BAILEY		9.88
6	NED	CHURANDY MARTINA		9.94
7	TRI	RICHARD THOMPSON		9.98
8	JAM	ASAFA POWELL		11.99

Levels of measurement

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!



MEN'S 100M

RESULT - FINAL

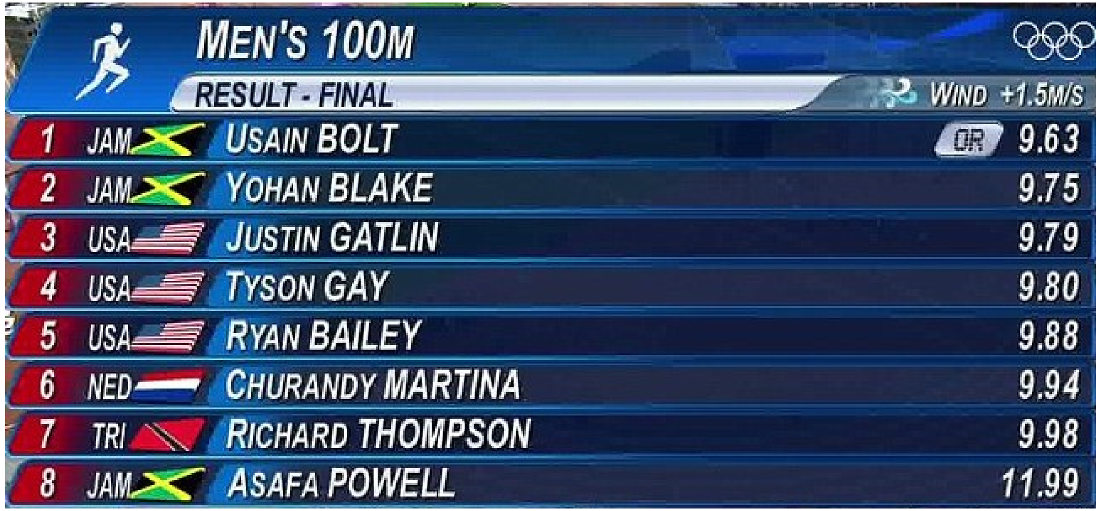
WIND +1.5M/S

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)



MEN'S 100M

RESULT - FINAL

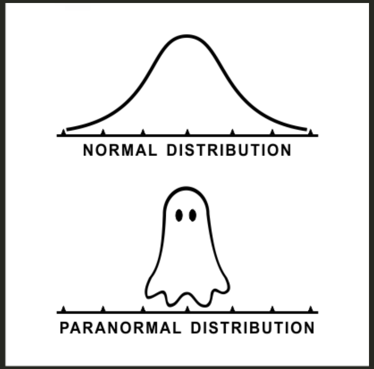
WIND +1.5M/S

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

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.)

The assumption of *normality*

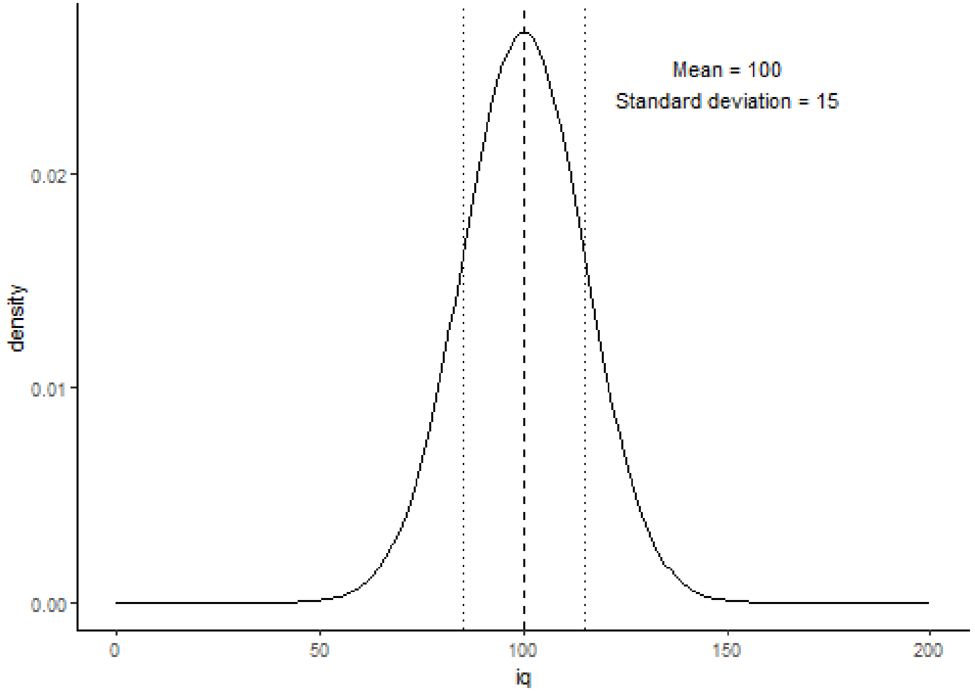


The assumption of *normality*

A normal distribution can be easily described by two parameters: the mean - μ - and the standard deviation - σ.

The normal distribution is approximately symetrically distributed about its central values.

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



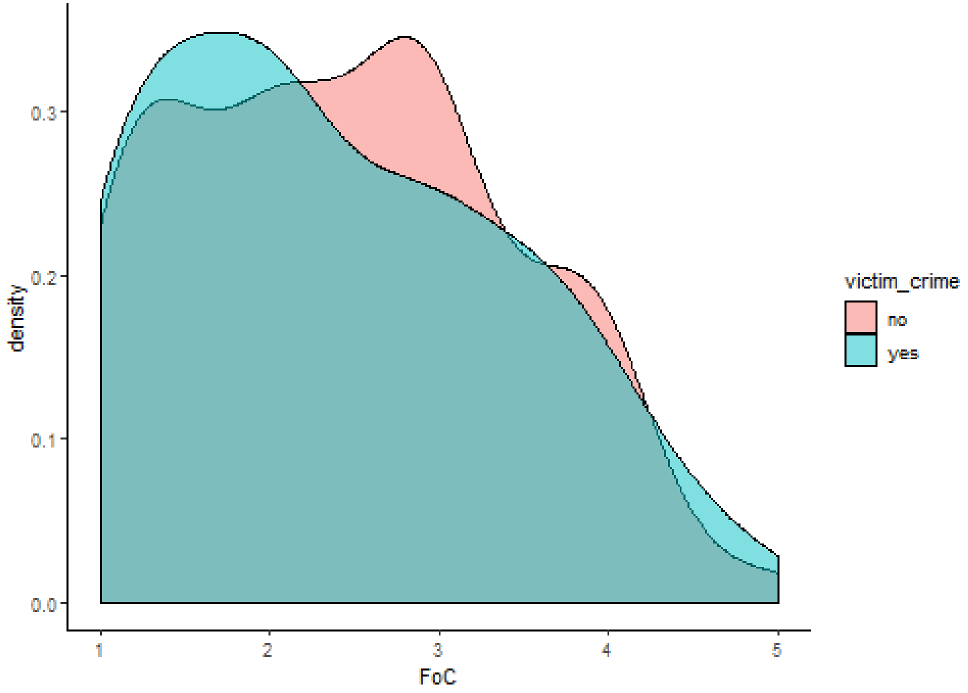
Checking normality

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()**.



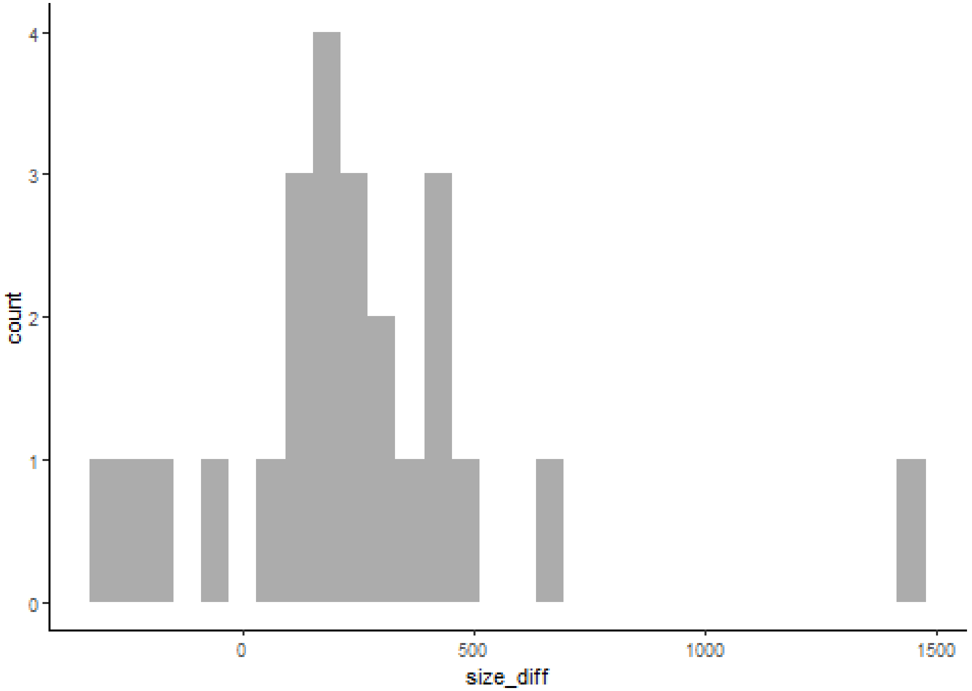
Checking normality

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) +
  theme_classic()
```

```
## `stat_bin()` using `bins = 30`. Pick better value
```



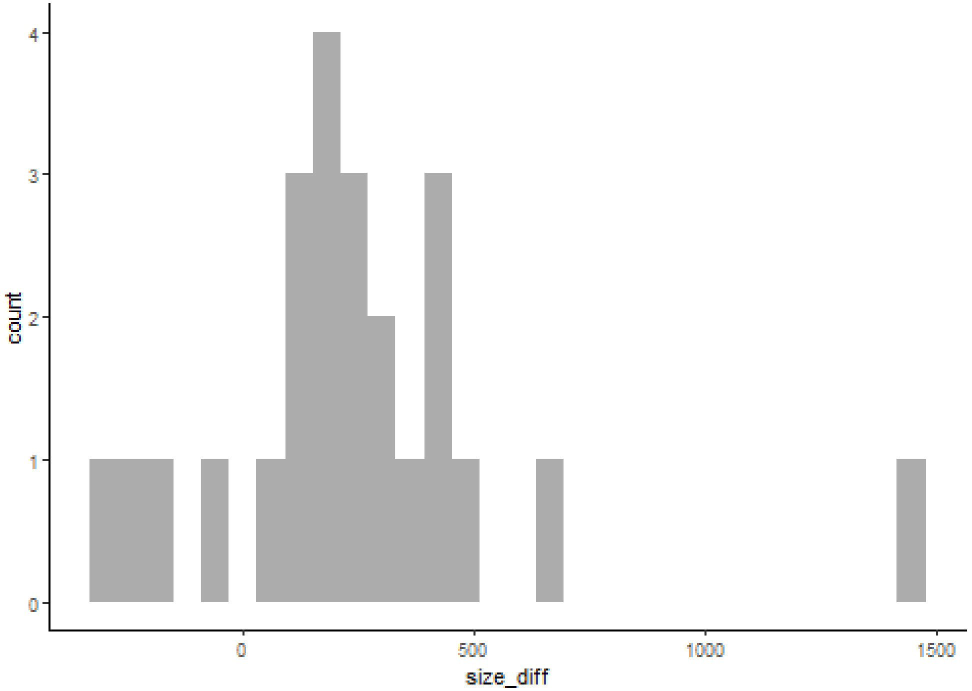
Checking normality

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 suggests that normality is violated.



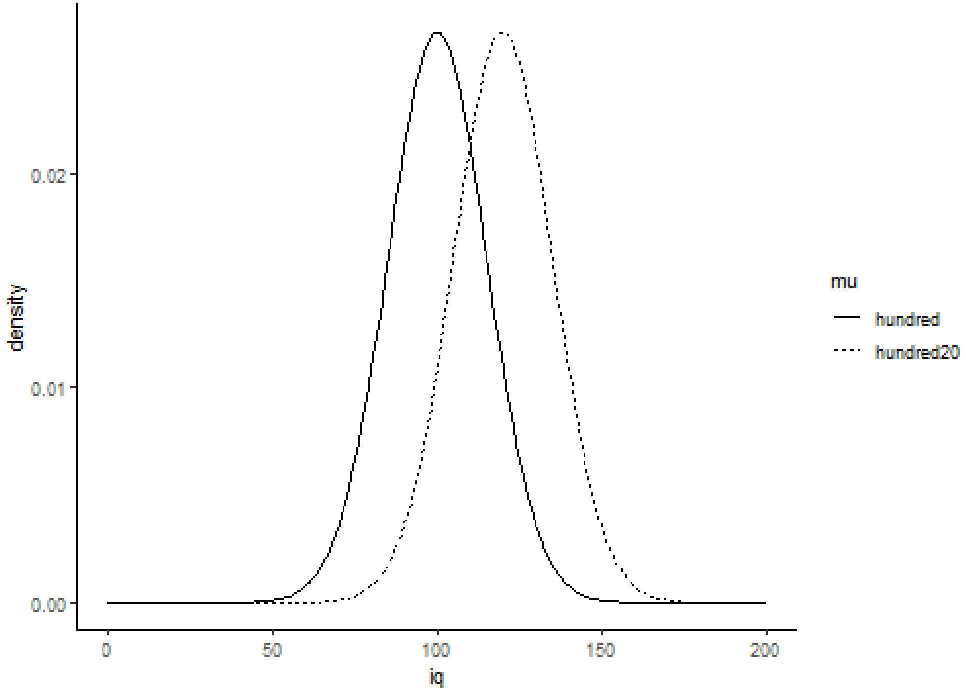
Homogeneity of variance

Homogeneity of variance

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

As we increase the mean, the distribution shifts further along the x-axis (or vice versa!)

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.



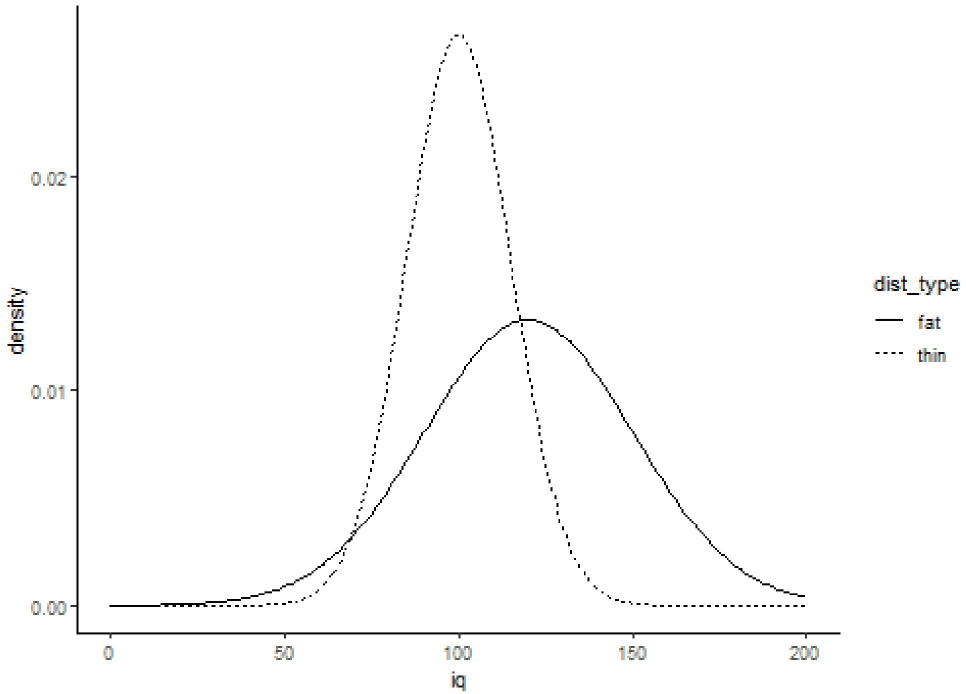
Homogeneity of variance

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

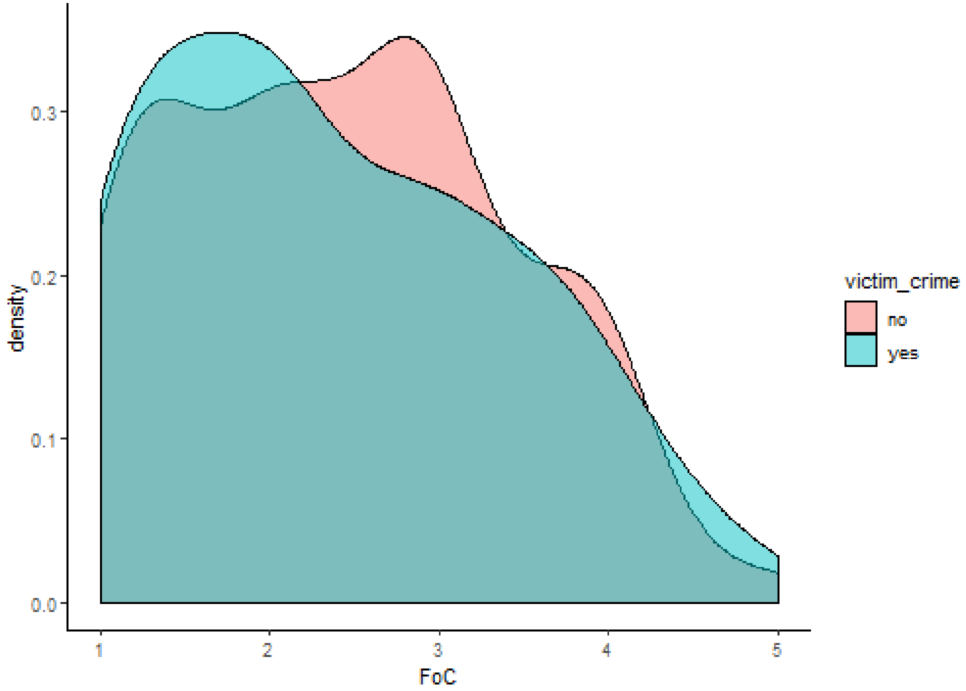
Let's look again at our distributions for Fear of Crime.

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

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

```r
library(car)
```

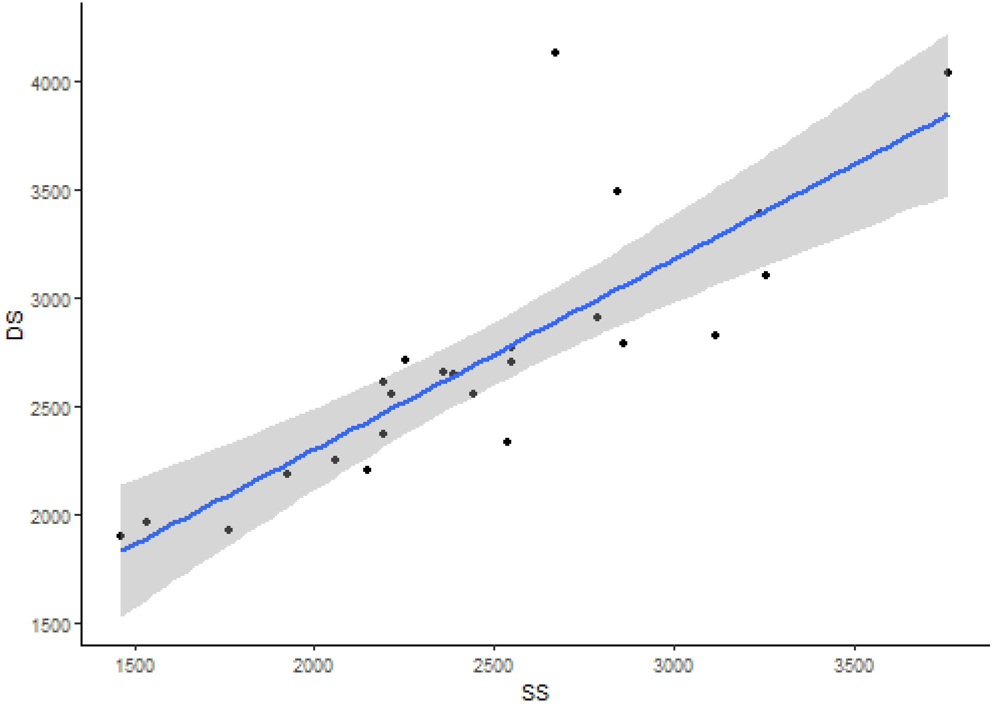
```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
```



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 occassions tend to be correlated.



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 or repeated-measures t-test.
3. When normality is violated
    - Often…nothing is done, or the data is *transformed*. But see next slide…

What to do about violated assumptions

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()*!

```r
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 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 x 3
##   victim_crime means    sem
##   <chr>        <dbl>  <dbl>
## 1 no            2.46 0.0696
## 2 yes           2.41 0.102
```

Reporting the results of a *t*-test

Report to TWO decimal places, typically.

"On average, victims of crime participants did not have significanly higher Fear of Crime (*M* = 2.41, *SE* = .10) than male participants (*M* = 2.46, *SE* = .07), *t*(197.48) = .453, *p* = .65."

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

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

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

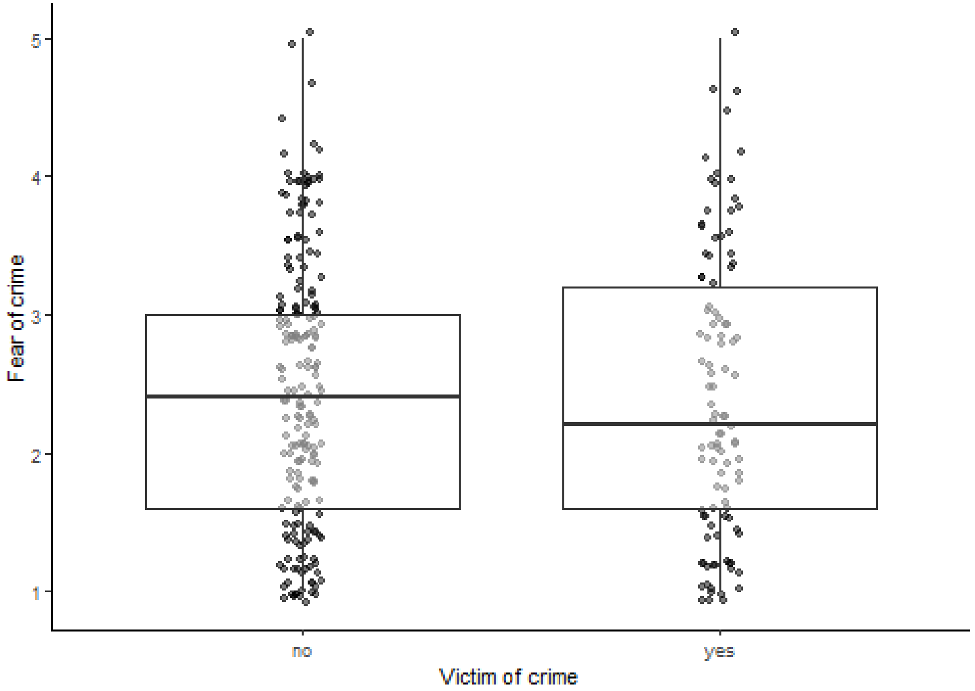
A plotting suggestion

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 cr
```



Next session

Enjoy next week's reading week…!

The week after that 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*.

UNIVERSITY OF LINCOLN