Summarising and Plotting Data in R

Analysing Data in R

Glenn Williams

University of Sunderland

2021-10-27 (updated: 2021-10-27)

Some Background

Data analysis is surprisingly one of the easiest parts of working with R.

- Once your data is in the correct (long) format, analysis using any test is highly consistent.

- We rely on a formula interface like this:

```
DV ~ IV
```

- Our dependent variable/predicted variable goes to the left of the ~ (tilde), while our independent variables or predictors go to the right.

- After this we specify our data:

```
DV ~ IV, data
```

We then apply a function to our formula which is the name of our test. There's some minor options we can choose within tests, but that's pretty much it!

Correlations

The Data

Let's check out the **starwars** data set again. We'll use this for our tests.

```
starwars <- starwars %>% filter(mass < 500)
```

We will use the height and mass columns, looking at whether mass is associated with height.

```
## Rows: 58
## Columns: 14
## $ name       <chr> "Luke Skywalker", "C-3PO", "R2-D2", "Darth Vader", "Lei
## $ height     <int> 172, 167, 96, 202, 150, 178, 165, 97, 183, 182, 188, 22
## $ mass       <dbl> 77.0, 75.0, 32.0, 136.0, 49.0, 120.0, 75.0, 32.0, 84.0,
## $ hair_color <chr> "blond", NA, NA, "none", "brown", "brown, grey", "brown
## $ skin_color <chr> "fair", "gold", "white, blue", "white", "light", "light
## $ eye_color  <chr> "blue", "yellow", "red", "yellow", "brown", "blue", "bl
## $ birth_year <dbl> 19.0, 112.0, 33.0, 41.9, 19.0, 52.0, 47.0, NA, 24.0, 57
## $ sex        <chr> "male", "none", "none", "male", "female", "male", "fema
## $ gender     <chr> "masculine", "masculine", "masculine", "masculine", "fe
## $ homeworld  <chr> "Tatooine", "Tatooine", "Naboo", "Tatooine", "Alderaan"
```

Correlation

- Here, we aren't predicting any one variable from the other, so both variables go to the right of the tilde.

- We add multiple variables with a +.

- We choose the type of correlation we want (e.g. Pearson, Spearman) with the method.

```
cor.test(~ height + mass, starwars, method = "pearson")
```

```
## 
##      Pearson's product-moment correlation
## 
## data:  height and mass
## t = 8.7853, df = 56, p-value = 4.018e-12
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6260700 0.8520232
## sample estimates:
##       cor 
## 0.7612612
```

Tests of Difference

- We'll use some different data here on out.
- Let's assume this data looks at giving people a placebo or drug, and tests the effect of that drug at two different time points.
- We care about improvements in reaction times.

```r
mixed_data <- read_csv(here("data", "mixed_factorial.csv"))
head(mixed_data)
```

```
## # A tibble: 6 x 4
##   id    drug    time          rt
##   <chr> <chr>   <chr>      <dbl>
## 1 S001  control daylater    431.
## 2 S001  control monthlater  421.
## 3 S002  control daylater    372.
## 4 S002  control monthlater  350.
## 5 S003  control daylater    393.
## 6 S003  control monthlater  368.
```

t-tests

One-sample t-test

- We have only one variable here, so we don't even need a formula.
- We compare the mean of this variable against a specified baseline mean (here 400).

```
t.test(mixed_data$rt, mu = 400)
```

```
##
##      One Sample t-test
##
## data:  mixed_data$rt
## t = -9.5311, df = 479, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 400
## 95 percent confidence interval:
##  376.3267 384.4193
## sample estimates:
## mean of x
##   380.373
```

t-tests

Independent-samples t-test

- Do reaction times vary depending on the drug given to participants?
- We test reaction times predicted by drug, with a regular t-test where variances are assumed to be equal (`var.equal = TRUE`).

```
t.test(rt ~ drug, mixed_data, var.equal = TRUE)
```

```
##
##      Two Sample t-test
##
## data:  rt by drug
## t = 3.732, df = 478, p-value = 0.0002128
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   7.18135 23.15251
## sample estimates:
##   mean in group control mean in group treatment
##                387.9564                372.7895
```

t-tests

Paired t-test

- Do reaction times vary over time (i.e. practice)?

- We test reaction times predicted by the time of testing. This is a paired test (`paired = TRUE`) and a regular t-test where variances are assumed to be equal (`var.equal = TRUE`).

```
t.test(rt ~ time, mixed_data, paired = TRUE, var.equal = TRUE)
```

```
##
##      Paired t-test
##
## data:  rt by time
## t = 18.348, df = 239, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  49.10644 60.91970
## sample estimates:
## mean of the differences
```

One-way ANOVA

Between-subjects

- What if we had **more than two groups** for the drug condition? We use an ANOVA.

- We simply change the test function to `aov()` (**A**nalysis **O**f **V**ariance)

- We need to summarise the model results here to get a regular ANOVA output.

```
summary(aov(rt ~ drug, mixed_data))
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)    
## drug          1  27604   27604   13.93 0.000213 ***
## Residuals   478 947379    1982                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

One-way ANOVA

Within-subjects

- What if we have more than two groups and a **within-subjects design**?
- We do the same as before, but need to add an **Error term** to the formula. This states that we adjust our errors to account for the fact scores in each group belong to the same participant (i.e. **id** in our data).

```
summary(aov(rt ~ time + Error(id), mixed_data))
```

```
##
## Error: id
##            Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 239 353969    1481
##
## Error: Within
##            Df Sum Sq Mean Sq F value Pr(>F)
## time        1 363173  363173   336.6 <2e-16 ***
## Residuals 239 257842    1079
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Two-Way ANOVA

Mixed

```
summary(aov(rt ~ time * drug + Error(id), mixed_data))
```

```
## 
## Error: id
##              Df Sum Sq Mean Sq F value   Pr(>F)    
## drug          1  27604   27604   20.13 1.13e-05 ***
## Residuals   238 326365    1371                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Error: Within
##              Df Sum Sq Mean Sq F value Pr(>F)    
## time          1 363173  363173   806.3 <2e-16 ***
## time:drug     1 150636  150636   334.4 <2e-16 ***
## Residuals   238 107206     450                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Throw Away the Alphabet Soup

All of the statistical tests you know (e.g. *t*-tests, ANOVA, chi-square) are just extensions of the **general linear model**. This is the most important thing you can learn to use in statistics.

Learn the mean and *variance* of some measurement by using an additive combination of other measurements.

- The **geocentric model of applied statistics**: used wisely, can be useful. But we shouldn't read too much into the numbers produced. They're almost certainly wrong because we can't (and shouldn't) model all sources of variance.
- Predict a **linear relationship** between one or more variable(s) and a continuous (e.g. scale) dependent variable.
- Predictor variables can be continuous or categorical.

Linear Regression

Takes the general form:

$$Y = \alpha + \beta X + e$$

- **Outcome** Y = intercept + (slope $\times$ X) + residual error
- **Residuals** e = distance of observed values from predicted values
- *Note*: We do not fit a perfect model, hence the error term. This is a good thing, otherwise we are probably **overfitting** to our data; relying too much on our observed sample to draw infferences.

Linear Regression

Takes the general form:

$$Y = \alpha + \beta X + e$$

- The **intercept**, α, is usually the point on the y-axis at the lowest value of X (usually 0).
- The **slope**, β, corresponds to how much Y increases by for every increment in X.
- The **error**, e, corresponds to a constant by which to add to our estimates accounting for additional variation from other sources that we do not model.

Linear Regression

Fit the model predicting height from weight from the starwars data.

$$Y = \alpha + \beta X + e$$

$$height = intercept + slope \times mass + error$$

```
starlm <- lm(height ~ mass, starwars)
summary(starlm)
```

```
##
## Call:
## lm(formula = height ~ mass, data = starwars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.369  -6.816   2.042  13.851  44.719
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 103.5133     8.5937  12.045  < 2e-16 ***
## mass          0.9327     0.1062   8.785 4.02e-12 ***
## ---
```

Comparing tests we know...

Correlation

```
broom::tidy(cor.test(~ height + mass, starwars, method = "pearson"))
```

```
## # A tibble: 1 x 8
##   estimate statistic  p.value parameter conf.low conf.high method    alter
##      <dbl>     <dbl>    <dbl>     <int>    <dbl>     <dbl> <chr>     <chr>
## 1    0.761      8.79 4.02e-12        56    0.626     0.852 Pearson'… two.s
```

```
broom::tidy(lm(height ~ mass, starwars, method = "pearson"))
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  104.        8.59      12.0  3.53e-17
## 2 mass           0.933     0.106      8.79 4.02e-12
```

t statistics match exactly.

Comparing tests we know...

t-tests

```
broom::tidy(t.test(rt ~ drug, mixed_data, var.equal = TRUE))
```

```
## # A tibble: 1 x 10
##   estimate estimate1 estimate2 statistic  p.value parameter conf.low conf.
##      <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>    <dbl>     <
## 1     15.2      388.      373.      3.73 0.000213       478     7.18
## # … with 2 more variables: method <chr>, alternative <chr>
```

```
broom::tidy(summary(lm(rt ~ drug, mixed_data)))
```

```
## # A tibble: 2 x 5
##   term          estimate std.error statistic  p.value
##   <chr>            <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)      388.       2.87    135.   0
## 2 drugtreatment    -15.2      4.06     -3.73 0.000213
```

t statistics match exactly.

Comparing tests we know…

ANOVA

```
summary(aov(rt ~ drug, mixed_data))
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## drug          1  27604   27604   13.93 0.000213 ***
## Residuals   478 947379    1982
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
broom::tidy(lm(rt ~ drug, mixed_data))
```

```
## # A tibble: 2 x 5
##   term          estimate std.error statistic  p.value
##   <chr>            <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)      388.       2.87    135.   0
## 2 drugtreatment    -15.2      4.06     -3.73 0.000213
```

t to F is just t squared. So, 3.732 squared = 13.93…

Bye!

Effect sizes are easily handled by the {effectsize} package. Super-easy ANOVAs are done using the {afex} package.