Chapter 4: Basic Analyses

Tyson S. Barrett

Summer 2017

Utah State University

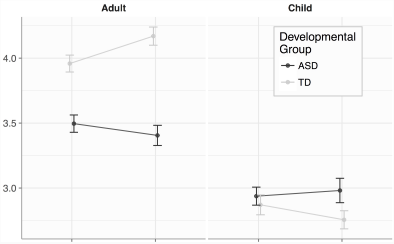
Introduction

Basic Analyses: The analyses taught in the first stats course

These include:

1. T-tests
2. ANOVA
3. Linear Regression

These allow us to assess relationships like that in the figure.



Maybe surprising:\ These all are doing essentially the same thing!

First, **T-TESTS!**

T-tests

1. Simple
2. Independent Samples
3. Paired Samples

Three Types

Each will be demonstrated using:

```r
df <- data.frame("A"=sample(c(0,1), 100, replace = TRUE),
                 "B"=rnorm(100),
                 "C"=rnorm(100))
df
```

```
    A           B            C
1   0 -0.888158433  0.294452230
2   1  0.384032654 -2.022480886
3   0 -0.978200548  0.363196635
4   0  0.597665769  0.306536631
5   0  0.849400438 -0.444227641
6   0 -0.890268979 -1.254064551
7   0 -0.854688613  0.938866598
8   0 -0.148777057 -2.283803888
9   0  1.046238407 -0.149862421
```

Simple

Comparing a mean of a variable with μ.

```r
t.test(df$B, mu = 0)
```

```
    One Sample t-test

data:  df$B
t = 0.29332, df = 99, p-value = 0.7699
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.1803399  0.2429081
sample estimates:
 mean of x
0.03128405
```

Independent Samples

Comparing the means of two groups (dfA is the grouping variable).

```r
t.test(df$B ~ df$A)
```

```

    Welch Two Sample t-test

data:  df$B by df$A
t = 0.59367, df = 89.632, p-value = 0.5542
alternative hypothesis: true difference in means is not equal to
95 percent confidence interval:
 -0.3009458  0.5574410
sample estimates:
mean in group 0 mean in group 1
     0.08514805     -0.04309956
```

Paired Samples

Comparing repeated measures (e.g., Pretest vs. Posttest).

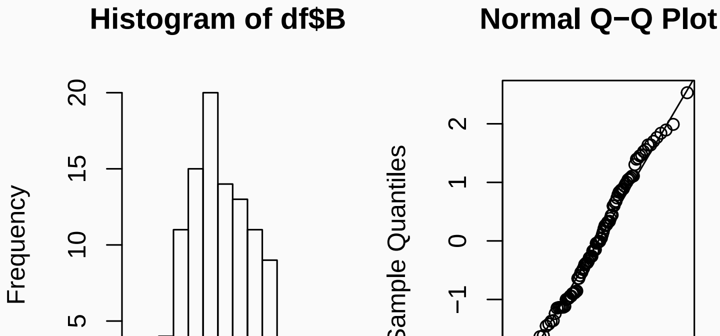
```
t.test(df$B, df$C, paired = TRUE)
```

```

	Paired t-test

data:  df$B and df$C
t = 1.7086, df = 99, p-value = 0.09066
alternative hypothesis: true difference in means is not equal to
95 percent confidence interval:
 -0.03873655  0.51897089
sample estimates:
mean of the differences
              0.2401172
```

Testing Assumptions of T-Tests

T-tests require that the data be normally distributed with approximately the same variance.

```
## Normality
par(mfrow = c(1,2))
hist(df$B)
qqnorm(df$B)
abline(a=0, b=1)
```



ANOVA

Analysis of Variance

The Analysis of Variance (ANOVA) is highly related to t-tests but can handle 2+ groups.

1. Provides the same p-value as t-tests
2. $t^2 = F$

For example:

```
fit_ano = aov(df$B ~ df$A)
summary(fit_ano)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
df$A         1    0.4  0.4007    0.35  0.556
Residuals   98  112.2  1.1450
```

```
t.test(df$B ~ df$A)$p.value
```

```
[1] 0.5542261
```

Analysis of Variance

```
fit_ano = aov(df$B ~ df$A)
summary(fit_ano)
t.test(df$B ~ df$A)$p.value
```

Notice in the code:

- We assigned the `aov()` the name `fit_ano` (which we could have called anything)
- We used the `summary()` function to see the F and p values.
- We pulled the p-value right out of the `t.test()` function.

1. One-Way
2. Two-Way (Factorial)
3. Repeated Measures
4. A combination of Factorial and Repeated Measures

Types

We will use the following data set for the examples:

```
library(tidyverse)
df <- data.frame("A"=sample(c(0,1), 100, replace = TRUE) %>% fac
                 "B"=rnorm(100),
                 "C"=rnorm(100),
                 "D"=sample(c(1:4), 100, replace = TRUE) %>% fac
df
```

```
    A          B           C D
1   1  0.393203143  0.166533149 3
2   1  0.325532077  0.121702597 4
3   0 -0.716154628  0.208602755 3
4   0 -0.544302202  0.033837841 2
5   0 -1.040692589 -0.353863739 1
6   1  1.086275696 -0.249269105 2
7   0 -0.304352002 -1.226131107 2
8   1 -0.825585271 -1.068904014 4
```

One-Way

A One-Way ANOVA can be run using `aov()`.

```
fit1 = aov(B ~ D, data = df)
summary(fit1)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
D            3   1.55  0.5152   0.499  0.684
Residuals   96  99.12  1.0325
```

Two-Way

A Two-Way ANOVA uses essentially the exact same code with a minor change—including the other variable in an interaction.

```
fit2 = aov(B ~ D * A, data = df)
summary(fit2)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
D            3   1.55  0.5152   0.502  0.682
A            1   1.28  1.2818   1.250  0.266
D:A          3   3.50  1.1679   1.139  0.338
Residuals   92  94.34  1.0254
```

The `D:A` line highlights the interaction term whereas the others show the main effects.

Repeated Measures

To show this, we will add a fake ID variable to our already fake data set df.

```
df$ID = 1:100
```

And change our data to long (Can you remember how to do it?)

```
library(tidyverse)
df_long = gather(df, "var", "value", 2:3)
df_long
```

```
    A D  ID var        value
1   1 3   1   B  0.393203143
2   1 4   2   B  0.325532077
3   0 3   3   B -0.716154628
4   0 2   4   B -0.544302202
5   0 1   5   B -1.040692589
6   1 2   6   B  1.086275696
7   0 2   7   B -0.304352002
```

Repeated Measures

The repeated measures, besides using a long-form of the data, is very similar in code. In addition to our usual formula (e.g., `something ~ other + stuff`), we have the `Error()` function. This function tells `R` how the repeated measures are clustered. In general, you'll provide the subject ID.

The next slide highlights this.

Repeated Measures

```
fit3 = aov(value ~ var + Error(ID), data = df_long)
summary(fit3)
```

```
Error: ID
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  1  2.222   2.222

Error: Within
           Df Sum Sq Mean Sq F value Pr(>F)
var         1   1.95  1.9474   2.101  0.149
Residuals 197 182.57  0.9267
```

Here, `value` was the value of the repeated measures where `var` is the time. That means our oucome is testing if there were any differences from pre-test to post-test across all the groups.

Combination

To take the repeated measures a step further, we can do a Three-Way Repeated Measures ANOVA.

```
fit4 = aov(value ~ var * D * A + Error(ID), data = df_long)
summary(fit4)
```

The output is on the next slide...

Combination

```
Error: ID
  Df Sum Sq Mean Sq
D  1  2.222   2.222

Error: Within
           Df Sum Sq Mean Sq F value Pr(>F)
var         1   1.95  1.9474   2.088  0.150
D           3   1.43  0.4780   0.512  0.674
A           1   0.76  0.7593   0.814  0.368
var:D       3   1.26  0.4202   0.450  0.717
var:A       1   0.66  0.6576   0.705  0.402
D:A         3   2.72  0.9055   0.971  0.408
var:D:A     3   5.04  1.6794   1.800  0.149
Residuals 183 170.70  0.9328
```

Checking Assumptions

Of course, as with any statistical analysis, there are assumptions.

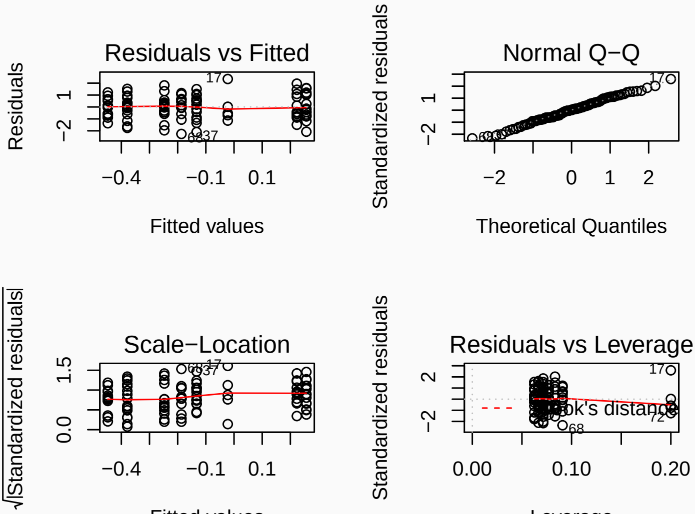
Many of these we can test.

Using our `fitX` objects from our ANOVAs above, we can look at our assumptions:

```
par(mfrow = c(2,2))
plot(fit2)
```

Again, the output is on the next slide...

Checking Assumptions



Checking Assumptions

They don't fit great on the slides but trust me that normality looks good. The assumption of homogeneity of variance looks good as well.

But, if you wanted to test it, you could.

```r
library(car)
leveneTest(fit2)
```

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  7  0.2113 0.9821
      92
```

Large p-value here is a good thing: `emo::ji("smile")` [1]

[1] This shows a smiley in 'R', just not on these slides—from the 'emo' package on GitHub.

Linear Regression

Linear Regression

Once again, linear regression is essentially the more flexible twin of ANOVA and t-tests.[2]

It can:

1. Handle continuous and categorical predictors (i.e., independent variables)
2. Less stringent assumption of equality of variances
3. Is what many other methods are built on (Chapter 5 and 6 will talk about some of these)

[2] It mainly only differs from ANOVA in the way it takes a dummy code rather than an effect code of the categorical variables.

Linear Regression

We will use `lm()` (Linear Model) to fit these models.

```
fit5 = lm(B ~ A, data = df)
summary(fit5)
```

```
Call:
lm(formula = B ~ A, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-2.2232 -0.5694 -0.0839  0.6275  2.5542

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.2291     0.1540  -1.488    0.140
A1            0.1765     0.2039   0.865    0.389

Residual standard error: 1.01 on 98 degrees of freedom
```