An Introduction to Statistical Data Analysis in R[1]

Mark Andrews
Psychology Department, Nottingham Trent University

[1] These slides are not intended to be self-contained and comprehensive, but just aim to provide some of the workshop's content. Much more will be provided in the workshop itself.

Linear regression

- Predict `ACT` as a linear function of `education` in the `sat_act` data frame.

```
sat_act <- read_csv('../data/sat_act.csv')
M <- lm(ACT ~ education, data=sat_act)
summary(M)
```

```
##
## Call:
## lm(formula = ACT ~ education, data = sat_act)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.9371  -3.4251   0.5389   3.5389   9.1108
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  26.8892     0.4391   61.23   < 2e-16 ***
## education     0.5240     0.1265    4.14 0.0000389 ***
## ---
```

Predictions in linear regression

- On the basis of our fitted model `M`, we can make predictions about possible values of the predictor variable.

```
hypothetical_data <- data.frame(education = c(1, 2, 5, 10, 15))
predict(M, newdata=hypothetical_data)
```

```
##        1        2        3        4        5
## 27.41314 27.93710 29.50898 32.12878 34.74858
```

Multiple linear regression

- We can add as many predictor variables as we like.

```
M <- lm(ACT ~ education + age + gender, data=sat_act)
summary(M)
```

```
##
## Call:
## lm(formula = ACT ~ education + age + gender, data = sat_act)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.2458  -3.2133   0.7769   3.5921   9.2630
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 27.41706    0.82140  33.378  < 2e-16 ***
## education    0.47890    0.15235   3.143  0.00174 **
## age          0.01623    0.02278   0.712  0.47650
## gender      -0.48606    0.37984  -1.280  0.20110
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

Collinearity

- We'll evaluate multicollinearity using Variance Inflation Factor (VIF):

```
library(car)
vif(M)
```

```
## education       age    gender
##  1.450002  1.439585  1.014574
```

General linear models

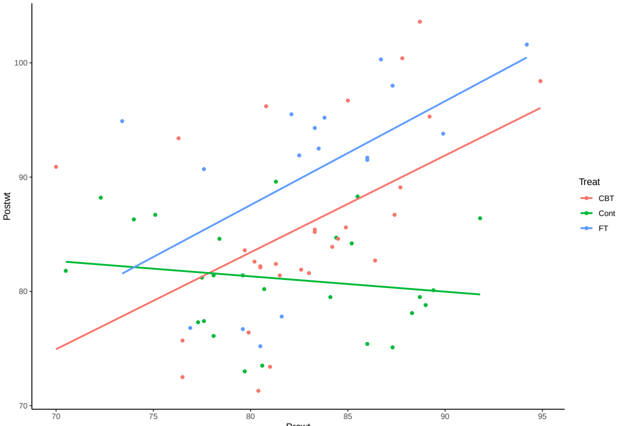
- We can use predictors that categorical as well as continuous in our model.
- Here, we investigate how the post treatment weight of a patient differs from their pre treatment weight, for three different types of therapy (control, CBT, family therapy).

```
anorexia <- read_csv('../data/anorexia.csv')
```

General linear models (continued)

- First, we'll visualize the data.

```
ggplot(anorexia,
       aes(x = Prewt, y = Postwt, col=Treat)) +
  geom_point() +
  stat_smooth(method='lm', se=F) +
  theme_classic()
```



General linear models (continued)

- Here, we do a *varying intercept*, which is also known as an *ANCOVA*:

```
M <- lm(Postwt ~ Prewt + Treat, data=anorexia)
summary(M)
```

```
##
## Call:
## lm(formula = Postwt ~ Prewt + Treat, data = anorexia)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.1083  -4.2773  -0.5484   5.4838  15.2922
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  49.7711    13.3910   3.717  0.00041 ***
## Prewt         0.4345     0.1612   2.695  0.00885 **
## TreatCont    -4.0971     1.8935  -2.164  0.03400 *
## TreatFT       4.5631     2.1333   2.139  0.03604 *
##
```

General linear models (continued)

- We cam also do a *varying slopes and varying intercepts* model. This is a type of interaction model:

```
M_interaction <- lm(Postwt ~ Prewt * Treat, data=anorexia)
summary(M_interaction)
```

```
##
## Call:
## lm(formula = Postwt ~ Prewt * Treat, data = anorexia)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.8125  -3.8501  -0.9153   4.0010  15.9640
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      15.57724   21.20828   0.734  0.46525
## Prewt             0.84798    0.25606   3.312  0.00151 **
## TreatCont        76.47423   28.34700   2.698  0.00885 **
## TreatFT          -0.75749   34.55162  -0.022  0.98258
## Prewt:TreatCont  -0.98217    0.34424  -2.853  0.00578 **
```

Model evaluation

- We can compare any two linear models using the generic `anova` function.
- Here, we'll use this to test whether the varying slopes and intercepts model is a better fit to the data than the just varying intercepts model:

```
anova(M, M_interaction)
```

```
## Analysis of Variance Table
##
## Model 1: Postwt ~ Prewt + Treat
## Model 2: Postwt ~ Prewt * Treat
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     68 3311.3
## 2     66 2844.8  2    466.48 5.4112 0.006666 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

One-way Anova

- We can use `aov` for one-way (and other) Anova.

```
data(PlantGrowth)
M <- aov(weight ~ group, data=PlantGrowth)
summary(M)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## group        2  3.766  1.8832   4.846 0.0159 *
## Residuals   27 10.492  0.3886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

Multiple comparisons

- We can do Tukey's range test to perform multiple comparisons:

```
TukeyHSD(M)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = weight ~ group, data = PlantGrowth)
##
## $group
##             diff        lwr       upr     p adj
## trt1-ctrl -0.371 -1.0622161 0.3202161 0.3908711
## trt2-ctrl  0.494 -0.1972161 1.1852161 0.1979960
## trt2-trt1  0.865  0.1737839 1.5562161 0.0120064
```

One-way Anova (alternative)

- Note that we can also we can do Anova using `lm()`:

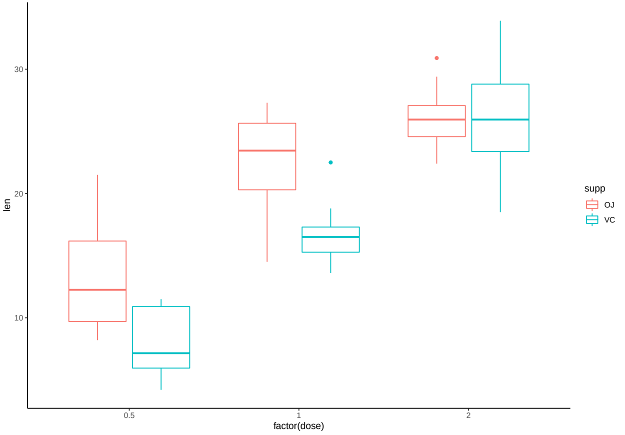
```
M <- lm(weight ~ group, data=PlantGrowth)
anova(M)
```

```
## Analysis of Variance Table
##
## Response: weight
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## group      2  3.7663  1.8832  4.8461 0.01591 *
## Residuals 27 10.4921  0.3886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

Two-way anova

```
data("ToothGrowth")

ggplot(ToothGrowth,
       aes(x = factor(dose), y = len, col = supp)) +
  geom_boxplot() +
  theme_classic()
```



Two-way (factorial) anova

```r
M <- aov(len ~ supp*dose, data=ToothGrowth)
summary(M)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## supp         1  205.4   205.4  12.317 0.000894 ***
## dose         1 2224.3  2224.3 133.415  < 2e-16 ***
## supp:dose    1   88.9    88.9   5.333 0.024631 *  
## Residuals   56  933.6    16.7                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

One-way repeated measures Anova

```
recall_data <- read_csv('../data/recall_data.csv')

M <- aov(Recall ~ Valence + Error(Subject/Valence), data=recall_
summary(M)
```

```
##
## Error: Subject
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  4  105.1   26.27
##
## Error: Subject:Valence
##           Df Sum Sq Mean Sq F value      Pr(>F)
## Valence    2 2029.7  1014.9   189.1 0.000000184 ***
## Residuals  8   42.9     5.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

One-way repeated measures Anova (continued)

- Multiple comparisons, with Bonferroni correction

```
with(recall_data,
     pairwise.t.test(x=Recall, g=Valence),
     p.adjust.methods='bonferroni',
     paired=T)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  Recall and Valence
##
##     Neg         Neu
## Neu 0.000019118 -
## Pos 0.00014     0.000000071
##
## P value adjustment method: holm
```

Twoway repeated measures Anova

```
recall_data2 <- read_csv('../data/recall_data2.csv')
M <- aov(Recall ~ Valence*Task + Error(Subject/(Task*Valence)),
         data=recall_data2)
summary(M)
```

```
##
## Error: Subject
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  4  349.1   87.28
##
## Error: Subject:Task
##           Df Sum Sq Mean Sq F value Pr(>F)
## Task       1  30.00  30.000   7.347 0.0535 .
## Residuals  4  16.33   4.083
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
##
## Error: Subject:Valence
##           Df Sum Sq Mean Sq F value Pr(>F)
## Valence    2   9.80   4.900   1.459  0.288
```

Multilevel models

- The repeated measures anova above can be done, and I think *should* be done, using multilevel models too.

```
library(lme4)
M <- lmer(Recall ~ Valence*Task + (1|Subject),
          data=recall_data2)
```