Linear models

Mark Andrews

July 5, 2018

Introduction

In linear models, we model the expected value of an outcome variable as a linear function of one or more predictor variables. Even when we know that this is not a great modelling assumption, linear models can still be very informative, especially for exploratory work. In any case, it is hard to progress to more complex and realistic models without first understanding linear models.

For the following, we will use some data from the `psych` package. So, first load that, and a few other goodies:

```
library(MASS)
library(car)
library(psych)
library(dplyr)
library(ggplot2)
library(readr)
library(lme4)
library(pander) # for making nice tables

Df <- mutate(sat.act,
             gender = factor(gender))
```

We'll start by predicting ACT (a standardized academic test) scores on the basis of education level (measured on a five point scale):

```
M <- lm(ACT ~ education, data=Df)
pander(summary(M))
```

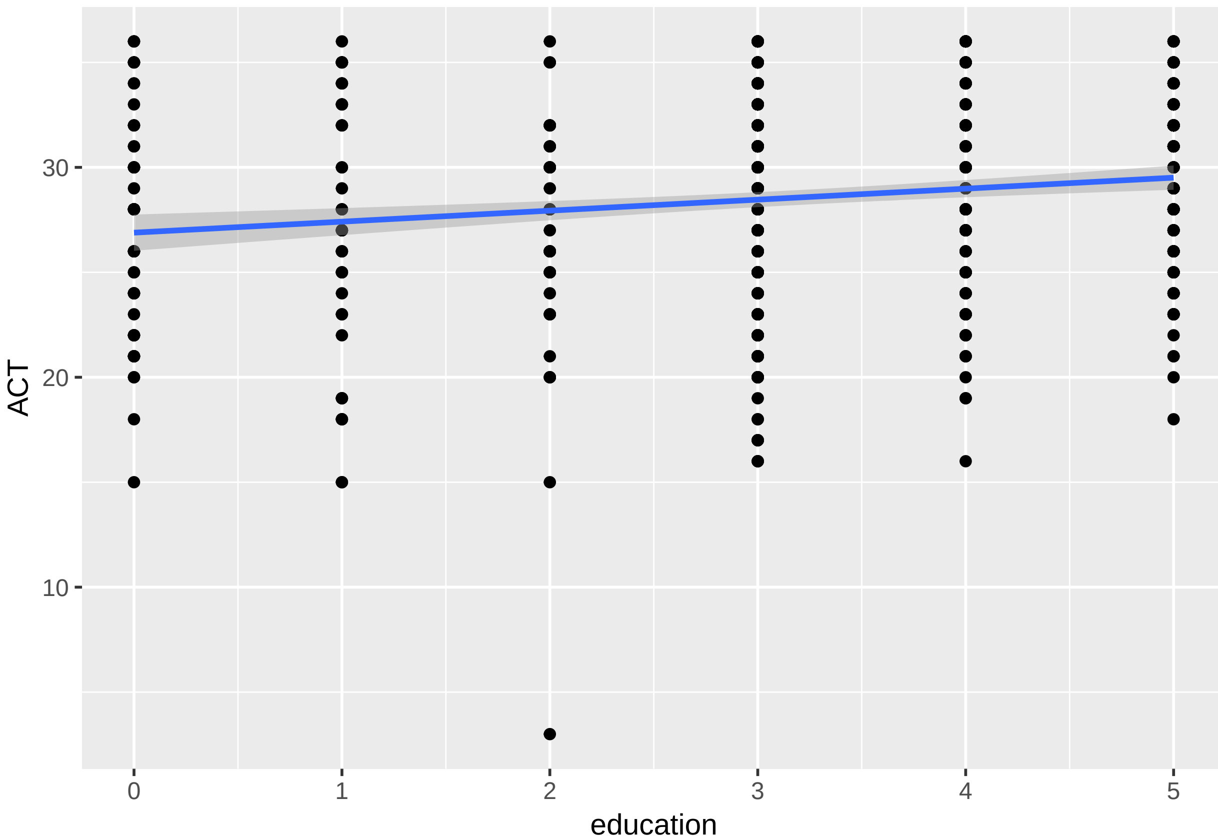
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	26.89	0.4391	61.23	6.733e-283
education	0.524	0.1265	4.14	3.89e-05

Table 2: Fitting linear model: ACT ~ education

Observations	Residual Std. Error	R^2	Adjusted R^2
700	4.769	0.02397	0.02257

We can visualize this as follows:

```
ggplot(Df,
       aes(x= education, y=ACT)) +
  geom_point() +
  stat_smooth(method='lm')
```



Confidence intervals

We can get confidence intervals as follows:

```
confint(M)
```

```
##                  2.5 %     97.5 %
## (Intercept) 26.0270213 27.7513496
## education    0.2755026  0.7724163
```

Predictions

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:

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

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	26.93	0.5919	45.5	1.025e-210
education	0.4789	0.1523	3.143	0.00174
age	0.01623	0.02278	0.7123	0.4765
gender2	-0.4861	0.3798	-1.28	0.2011

Table 4: Fitting linear model: ACT ~ education + age + gender

Observations	Residual Std. Error	R^2	Adjusted R^2
700	4.768	0.0272	0.02301

Collinearity

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

```r
vif(M)
```

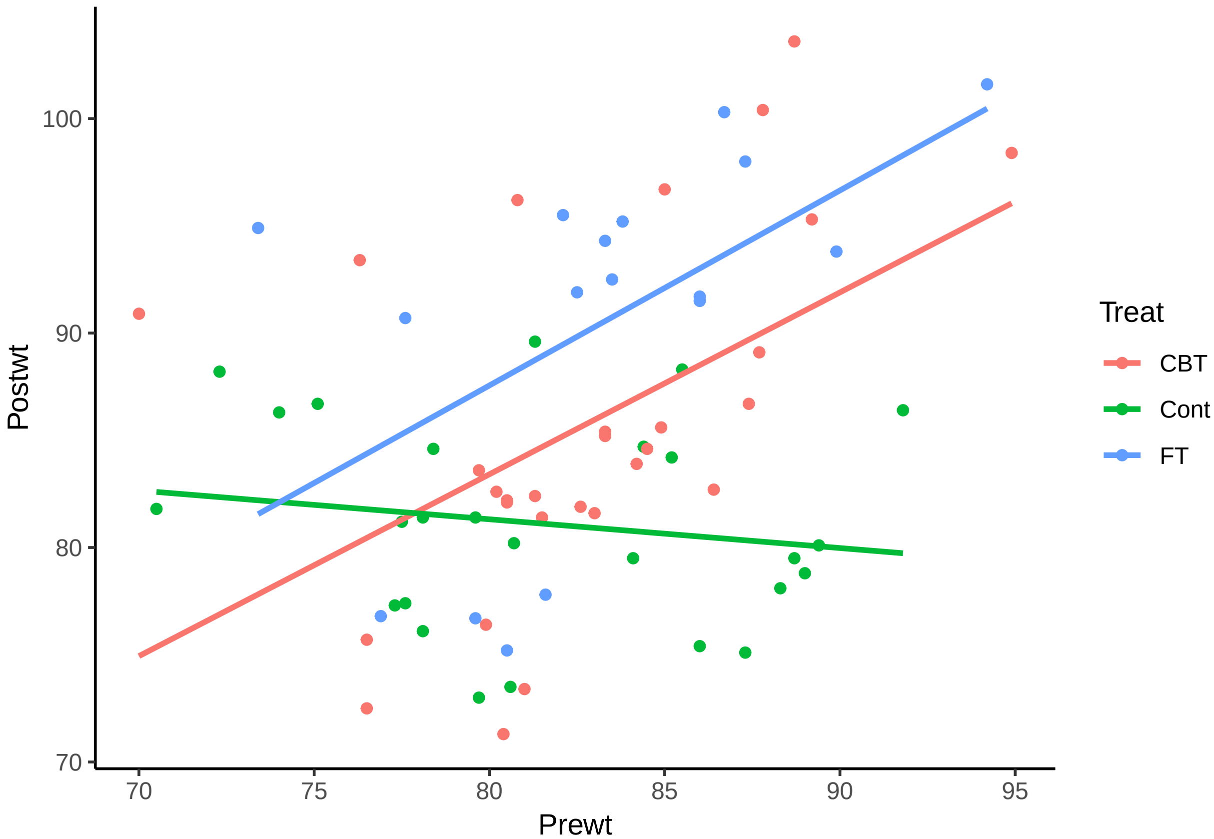
```
## education       age    gender
##  1.450002  1.439585  1.014574
```

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).

First, we'll visualize the data (we'll turn off standard error shading to allow the lines to be seen more easily):

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



Now, we'll do a *varying intercept*, which is also known as an *ANCOVA*:

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

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	49.77	13.39	3.717	0.0004101
Prewt	0.4345	0.1612	2.695	0.00885
TreatCont	-4.097	1.893	-2.164	0.034
TreatFT	4.563	2.133	2.139	0.03604

Table 6: Fitting linear model: Postwt ~ Prewt + Treat

Observations	Residual Std. Error	R^2	Adjusted R^2
72	6.978	0.2777	0.2458

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

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

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	15.58	21.21	0.7345	0.4653
Prewt	0.848	0.2561	3.312	0.001507

	Estimate	Std. Error	t value	Pr(>|t|)
TreatCont	76.47	28.35	2.698	0.008852
TreatFT	-0.7575	34.55	-0.02192	0.9826
Prewt:TreatCont	-0.9822	0.3442	-2.853	0.005776
Prewt:TreatFT	0.06124	0.4155	0.1474	0.8833

Table 8: Fitting linear model: Postwt ~ Prewt * Treat

Observations	Residual Std. Error	R^2	Adjusted R^2
72	6.565	0.3794	0.3324

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:

```
model_comparison <- anova(M, M_interaction)
pander(model_comparison, missing='')
```

Table 9: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
68	3311				
66	2845	2	466.5	5.411	0.006666

Anova

An Anova is just a general linear model. I'd love if we just left it like that, but some people in some fields treat Anova like is a some different and special. They're wrong, but let's give them what they want just to keep the peace.

One-way Anova

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

Table 10: Analysis of Variance Model

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
group	2	3.766	1.883	4.846	0.01591
Residuals	27	10.49	0.3886	NA	NA

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

```r
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
```

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

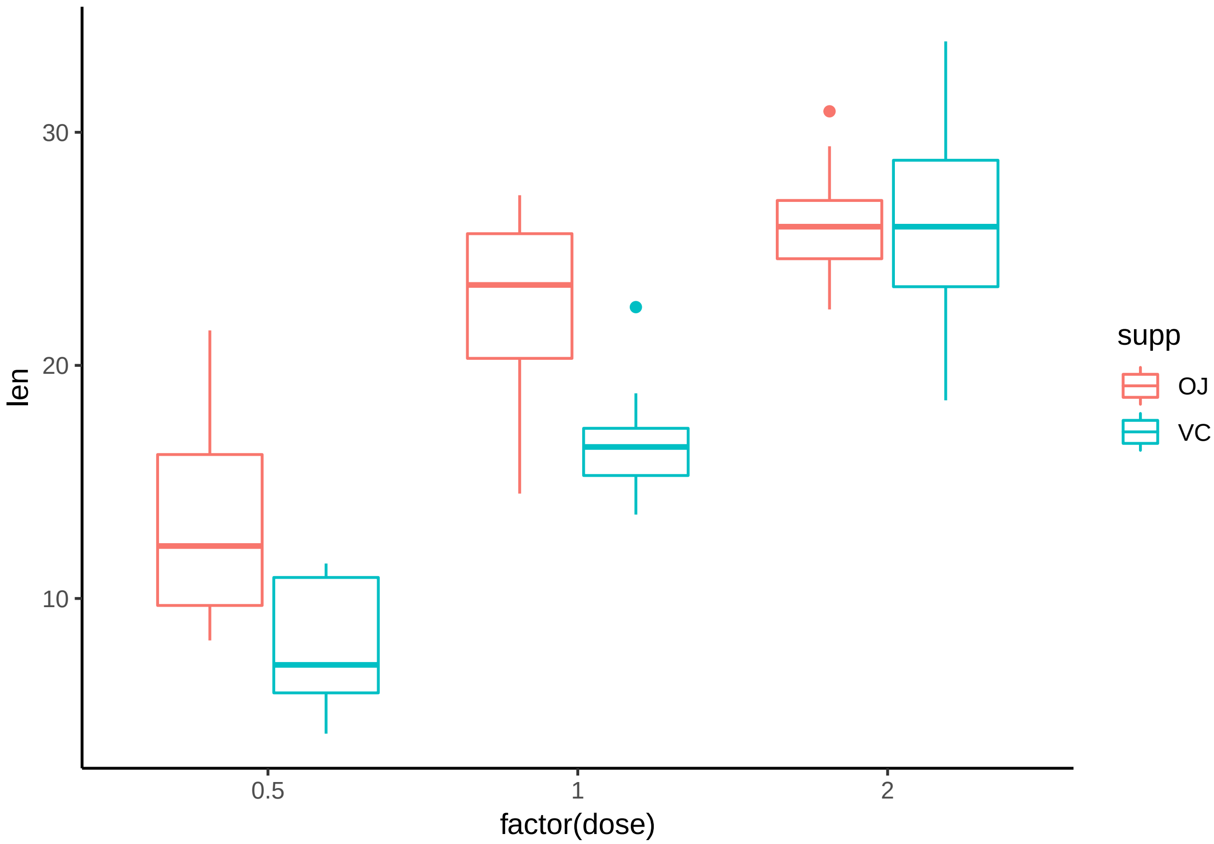
```r
M <- lm(weight ~ group, data=PlantGrowth)
anova(M)
```

```
## 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 ' ' 1
```

Two-way anova

```r
data("ToothGrowth")

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




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

Repeated measures Anova

Oneway

```
Df <- read_table('../data/recall_data.txt')
```

```
## Parsed with column specification:
## cols(
##   Observation = col_integer(),
##   Subject = col_character(),
##   Valence = col_character(),
##   Recall = col_integer()
## )
```

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

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Residuals	4	105.1	26.27	NA	NA
Valence	2	2030	1015	189.1	1.841e-07
Residuals1	8	42.93	5.367	NA	NA

Multiple comparisons, with Bonferroni correction

```r
with(Df,
     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 1.9e-05 -
## Pos 0.00014 7.1e-08
##
## P value adjustment method: holm
```

Twoway

```r
Df <- read_table('../data/recall_data2.txt')
```

```
## Parsed with column specification:
## cols(
##   Observation = col_integer(),
##   Subject = col_character(),
##   Task = col_character(),
##   Valence = col_character(),
##   Recall = col_integer()
## )
```

```r
M <- aov(Recall ~ Valence*Task + Error(Subject/(Task*Valence)), data=Df)
pander(M)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Residuals	4	349.1	87.28	NA	NA
Task	1	30	30	7.347	0.05351
Residuals1	4	16.33	4.083	NA	NA
Valence	2	9.8	4.9	1.459	0.2883
Residuals2	8	26.87	3.358	NA	NA
Valence:Task	2	1.4	0.7	0.2907	0.7553
Residuals	8	19.27	2.408	NA	NA

Multilevel models

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

```r
M <- lmer(Recall ~ Valence*Task + (1|Subject),
          data=Df)
```