8. Additional topics in linear modeling

Outline

- We now have practical skills to
  1. Write down linear models,
  2. Fit them in R,
  3. Interpret the output in terms of parameter estimates, confidence intervals and hypothesis tests,
  4. Check that R is fitting the model that we intend,
  5. Check that the model we intend is appropriate for the data.
- These skills provide a foundation for many extensions helpful for particuluar situations.

Topics

- The linear model formula notation in R, as a third model representation to join the subscript format and matrix format.
- Interactions between explanatory variables.
- The R^2 statistic to assess model fit.
- Fitting polynomial relationships using linear models.
- Multicollinearity: What happens when two or more explanatory variables are highly correlated. How to notice it, and what to do about it.
- Power: What is the probability of rejecting the null hypothesis when the alternative is true?

The R model formula notation

- A **formula** in `lm()` is something that looks like `y~x`.
- The R formula notation has various conventions that are designed to make it easy to specify useful models.
- `?formula` tells you everything you need to know, and more.
- The R formula for `lm()` is a way of constructing a design matrix.
- Inspect the resulting design matrix using `model.matrix()` and check you understand what R has produced. If you can do this, you can safely use the power of the formula notation.

Question 8.1. In a report, the model should be written in mathematical notation, not as an R formula. Why?

Experimenting with the R formula notation

- Consider the freshman GPA data

```
gpa <- read.table("gpa.txt",header=T); head(gpa,3)
```

```
##   ID  GPA High_School ACT Year
## 1  1 0.98          61  20 1996
## 2  2 1.13          84  20 1996
## 3  3 1.25          74  19 1996
```

- We can play the game of trying out various things in R formula notation, inspecting the resulting design matrix, and figuring out how to write the model efficiently in mathematical notation.
- You can also think about whether the different models give any new insights into the data.

```r
lm1 <- lm(GPA~ACT+High_School*Year,data=gpa)
coef(summary(lm1))[,1:2]
```

```
##                       Estimate   Std. Error
## (Intercept)      -4.722613e+01 1.350854e+02
## ACT               3.708961e-02 5.946966e-03
## High_School       3.460100e-01 1.702035e+00
## Year              2.428369e-02 6.760800e-02
## High_School:Year -1.681424e-04 8.518297e-04
```

- The * here denotes inclusion of an **interaction** between `High_School` and `Year`, written in the R output as `High_School:Year`.

Question 8.2. Conceptually, what do you think an interaction between two variables is, and why might it be needed?

- To find out exactly what R thinks an interaction is, we can check the design matrix.

```r
head(model.matrix(lm1))
```

```
##   (Intercept) ACT High_School Year High_School:Year
## 1           1  20          61 1996           121756
## 2           1  20          84 1996           167664
## 3           1  19          74 1996           147704
## 4           1  23          95 1996           189620
## 5           1  28          77 1996           153692
## 6           1  23          47 1996            93812
```

Question 8.3. Write out the sample model that R has computed in `lm1` using subscript notation.

Interactions and additivity

```
lm2 <- lm(GPA~ACT+High_School+Year+High_School:Year,data=gpa)
head(model.matrix(lm2),4)
```

```
##   (Intercept) ACT High_School Year High_School:Year
## 1           1  20          61 1996           121756
## 2           1  20          84 1996           167664
## 3           1  19          74 1996           147704
## 4           1  23          95 1996           189620
```

- `lm2` has the same design matrix as `lm1`.
- We see that, in R formula notation, `y~u*v` is the same as `y~u+v+u:v`.
- In the model `y~u+v` the effects of the variables are said to be **additive**.
- In a causal interpretation of an additive model, the result of changing `u` to `u2` and `v` to `v2` is the sum of the marginal effect of changing `u` to `u2` plus the marginal effect of changing `v` to `v2`.
- The interaction term `u:v` breaks additivity. In this case, we can't know the consequence for the fitted value of changing `u` to `u2` unless we know the value of `v`.

The interaction between ACT and high school percentile

- We have not (yet) found any interesting effect of year. Let's drop year out of the model and look for whether there is an interaction between ACT and high school percentile for predicting freshman GPA.

```
lm3 <- lm(GPA~ACT*High_School,data=gpa)
```

Question 8.4. Write out the fitted sample linear model in subscript form, letting y_i, a_i, h_i and e_i be the freshman GPA, ACT score, high school percentile and residual error respectively for the ith student.

Interpreting a discovered interaction

```
coef(summary(lm3))[,1:2]
```

```
##                      Estimate   Std. Error
## (Intercept)       3.157679842 0.4788067771
## ACT              -0.046067744 0.0213355076
## High_School      -0.014405030 0.0061479608
## ACT:High_School   0.001071326 0.0002638611
```

Question 8.5. Explain in words to the admissions director what you have found about the interaction under investigation here.

Marginal effects when there is an interaction

- Notice in ‘lm3‘ that the coefficients for ACT score and high school percentile are negative. That is surprising!

```
ACT_centered <- gpa$ACT-mean(gpa$ACT)
HS_centered <- gpa$Hi - mean(gpa$Hi)
lm3b <- lm(GPA~ACT_centered*HS_centered,data=gpa)
signif(coef(summary(lm3b))[,c(1,2,4)],3)
```

```
##                          Estimate Std. Error Pr(>|t|)
## (Intercept)               2.94000   0.022900 0.00e+00
## ACT_centered              0.03640   0.005880 1.04e-09
## HS_centered               0.01190   0.001350 8.23e-18
## ACT_centered:HS_centered  0.00107   0.000264 5.46e-05
```

Question 8.6. After centering the variables, the interaction effect stays the same, but the marginal effects change sign. What is happening? Why?

Quantifying the improvement in the model

```
s3 <- summary(lm3)$sigma
lm4 <- lm(GPA~ACT+High_School,data=gpa)
s4 <- summary(lm4)$sigma
lm5 <- lm(GPA~1,data=gpa)
s5 <- summary(lm5)$sigma
cat("s3 =",s3,"; s4 =",s4,"; s5 =",s5)
```

```
## s3 = 0.5610067 ; s4 = 0.5671605 ; s5 = 0.6345278
```

Question 8.7. Comment on both **statistical significance** and **practical significance** of the interaction between a prediction of freshman GPA.

An interaction involving a factor

- Let's go back to the football field goal data.

```
goals <- read.table("FieldGoals2003to2006.csv",header=T,sep=",")
goals[1,c("Name","Teamt","FGt","FGtM1")]


##              Name Teamt  FGt FGtM1
## 1 Adam Vinatieri    NE 73.5    90


lm6 <- lm(FGt~FGtM1*Name,data=goals)
```

Question 8.8. What model do you think is being fitted here? Write it in subscript form, where y_{ij} is the field goal average for the jth year of kicker i, with $i = 1, \ldots, 19$ and $j = 1, 2, 3, 4$. Let e_{ij} be the residual error, and let x_{ij} be the previous year's average. Check your answer against the design matrix shown on the next slide.

```r
X<-model.matrix(lm6) ; colnames(X)<-1:38 ; X[1:17,c(1:8,21:26)]
```

```
##    1    2 3 4 5 6 7 8   21   22   23 24 25 26
## 1  1 90.0 0 0 0 0 0 0  0.0  0.0  0.0  0  0  0
## 2  1 73.5 0 0 0 0 0 0  0.0  0.0  0.0  0  0  0
## 3  1 93.9 0 0 0 0 0 0  0.0  0.0  0.0  0  0  0
## 4  1 80.0 0 0 0 0 0 0  0.0  0.0  0.0  0  0  0
## 5  1 88.2 1 0 0 0 0 0 88.2  0.0  0.0  0  0  0
## 6  1 82.7 1 0 0 0 0 0 82.7  0.0  0.0  0  0  0
## 7  1 84.3 1 0 0 0 0 0 84.3  0.0  0.0  0  0  0
## 8  1 72.7 1 0 0 0 0 0 72.7  0.0  0.0  0  0  0
## 9  1 72.2 0 1 0 0 0 0  0.0 72.2  0.0  0  0  0
## 10 1 87.0 0 1 0 0 0 0  0.0 87.0  0.0  0  0  0
## 11 1 85.2 0 1 0 0 0 0  0.0 85.2  0.0  0  0  0
## 12 1 75.0 0 1 0 0 0 0  0.0 75.0  0.0  0  0  0
## 13 1 82.1 0 0 1 0 0 0  0.0  0.0 82.1  0  0  0
## 14 1 95.6 0 0 1 0 0 0  0.0  0.0 95.6  0  0  0
## 15 1 85.7 0 0 1 0 0 0  0.0  0.0 85.7  0  0  0
## 16 1 79.1 0 0 1 0 0 0  0.0  0.0 79.1  0  0  0
## 17 1 80.0 0 0 0 1 0 0  0.0  0.0  0.0 80  0  0
```

Question 8.9. Interpret the ANOVA table below.

```
anova(lm6)
```

```
## Analysis of Variance Table
##
## Response: FGt
##            Df  Sum Sq Mean Sq F value   Pr(>F)
## FGtM1       1   87.20  87.199  1.9008 0.176047
## Name       18 2252.47 125.137  2.7279 0.004565 **
## FGtM1:Name 18  417.75  23.209  0.5059 0.938592
## Residuals  38 1743.20  45.874
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Collinear explanatory variables in a linear model

- Let $\mathbb{X} = [x_{ij}]_{n\times p}$ be an $n \times p$ design matrix.
- If there is a nonzero vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)$ such that $\mathbb{X}\boldsymbol{\alpha} = \mathbf{0}$ then the columns of $\mathbb{X}$ are **collinear**.
- Here, $\mathbf{0}$ is the zero vector, $(0, 0, \ldots, 0)$.
- We can write $\mathbf{x}_j = (x_{1j}, x_{2j}, \ldots, x_{nj})$ for the jth column of $\mathbb{X}$. Then,

$$\mathbb{X}\boldsymbol{\alpha} = \alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2 + \cdots + \alpha_j\mathbf{x}_j.$$

  We see that $\mathbb{X}\boldsymbol{\alpha}$ can be thought of as a **linear combination of the columns of** $\mathbb{X}$.
- Collinearity of explanatory variables has important consequences for fitting a linear model to data.
- It can also be useful to notice whether the variables are close to collinear, meaning that $\mathbb{X}\boldsymbol{\alpha}$ is small but nonzero.

Example: an intercept with a coefficient for each factor

- Recall the mouse weight dataset. Consider a sample linear model,

$$y_{ij} = \mu + \mu_j + e_{ij}.$$

- Suppose that we don't set the $\mu_1 = 0$ so we try to estimate both μ_1 and μ_2 at the same time as the intercept, μ.
- Let's work with just 3 mice in each treatment group, so $i = 1, 2, 3$ and $j = 1, 2$. The design matrix is therefore

```
X <- cbind(rep(1,6),rep(c(1,0),each=3),rep(c(0,1),each=3)) ; X

##      [,1] [,2] [,3]
## [1,]    1    1    0
## [2,]    1    1    0
## [3,]    1    1    0
## [4,]    1    0    1
## [5,]    1    0    1
## [6,]    1    0    1
```

- For $\boldsymbol{\alpha} = (1, -1, -1)$, we have $\mathbb{X}\boldsymbol{\alpha} = 0$

The least squares fit with collinear predictors

- Suppose that $\mathbf{b}$ is a least squares coefficient vector, so that the fitted value vector $\hat{\mathbf{y}} = \mathbb{X}\mathbf{b}$ minimizes $\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2$.
- Suppose that $\mathbb{X}$ is collinear, with $\mathbb{X}\boldsymbol{\alpha} = \mathbf{0}$.
- Since

$$\mathbb{X}(\mathbf{b} + \boldsymbol{\alpha}) = \mathbb{X}\mathbf{b} + \mathbb{X}\boldsymbol{\alpha} = \mathbb{X}\mathbf{b} + \mathbf{0} = \mathbb{X}\mathbf{b},$$

  we see that $\mathbf{b} + \boldsymbol{\alpha}$ is also a least squares coefficient vector.
- **When $\mathbb{X}$ is collinear, a least squares coefficent still exists, but it is not unique.**

Question 8.10. Let c be any number. Recall multiplication of a vector by a scalar: $c\boldsymbol{\alpha} = (c\alpha_1, \ldots, c\alpha_p)$. Show that $\mathbf{b} + c\boldsymbol{\alpha}$ is also a least squares fit.

Standard errors for collinear variables

Question 8.11. Any variable that is part of a collinear combination of variables has infinite standard error. Why?

What does R do if give it collinear variables?

```r
mice <- read.table("femaleMiceWeights.csv",header=T,sep=",")
chow=rep(c(1,0),each=12)
hf=rep(c(0,1),each=12)
lm1 <- lm(Bodyweight~chow+hf,data=mice)
coef(summary(lm1))


##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 26.834167   1.039353 25.818139 6.045435e-18
## chow        -3.020833   1.469867 -2.055174 5.192480e-02
```

- R noticed that the three explanatory variables are collinear, and refused to fit the third

```r
model.matrix(lm1)
```

```
##    (Intercept) chow hf
## 1            1    1  0
## 2            1    1  0
## 3            1    1  0
## 4            1    1  0
## 5            1    1  0
## 6            1    1  0
## 7            1    1  0
## 8            1    1  0
## 9            1    1  0
## 10           1    1  0
## 11           1    1  0
## 12           1    1  0
## 13           1    0  1
## 14           1    0  1
## 15           1    0  1
## 16           1    0  1
## 17           1    0  1
## 18           1    0  1
## 19           1    0  1
```

Collinear variables and the determinant of $\mathbb{X}^{\mathrm{T}}\mathbb{X}$

- Recall that the variance of $\hat{\beta}$ in the usual linear model is $\sigma^2 \big(\mathbb{X}^{\mathrm{T}}\mathbb{X}\big)^{-1}$.
- Collinearity means the variance is infinite, a matrix version of dividing by zero.
- Recall that a square matrix is invertible if its determinant is nonzero.
- We can check that collinearity means $\det\big(\mathbb{X}^{\mathrm{T}}\mathbb{X}\big) = 0$.

```
X <- model.matrix(lm1)
t(X)%*%X
```

```
##             (Intercept) chow hf
## (Intercept)          24   12 12
## chow                 12   12  0
## hf                   12    0 12
```

```
det(t(X)%*%X)
```

```
## [1] 0
```

Linearly independent vectors and matrix rank

- Columns of a matrix that are not collinear are said to be **linearly independent**.
- The **rank** of $\mathbb{X}$ is the number of linearly independent columns.
- $\mathbb{X}$ has **full rank** if all the columns are linearly independent. In this case, we expect the least squares coefficient to be uniquely defined and so $\mathbb{X}^T\mathbb{X}$ has non-zero determinant and is invertible.
- If $\mathbb{X}$ does not have full rank, we can drop **linearly dependent** columns until the remaining columns are linearly independent. This is a practical approach to handling collinearity.

Example: reducing a design matrix to full rank

```
X <- model.matrix(lm1)
det(t(X)%*%X)

## [1] 0

X2 <- X[,1:2]
det(t(X2)%*%X2)

## [1] 144
```

- Dropping the third column of `X` has given us a full-rank design matrix.

Question 8.12. The least squares fitted values are the same using the predictor matrix `X2` as `X`. Why does dropping the last column not change the fitted values?

Almost collinear variables

- If the determinant of $\mathbb{X}^{\mathrm{T}}\mathbb{X}$ is close to zero, the variance of the model-generated least squares coefficient vector becomes large.
- This can happen when multiple explanatory variables are included in a model which all model similar things.

Question 8.13. Recall our data analysis using unemployment to explain life expectancy. What would happen if we added total employment as an additional explanatory variable? (Being unemployed is not the only alternative to being employed, since only adults currently looking for work are counted as unemployed.)

Using linear models to fit polynomial relationships

- Recall the basic linear trend model from Chapter 1 for data $y_1, \ldots, y_n$ with y_i measured at time t_i, $[\mathrm{M1}] \quad y_i = b_0 + b_1 t_i + e_i, \quad i = 1, \ldots, n$
- What if the data have a trend that is not linear?
- The next thing we might consider is a quadratic trend model, $[\mathrm{M2}] \quad y_i = b_0 + b_1 t_i + b_2 t_i^2 + e_i, \quad i = 1, \ldots, n$

M1and [M2] are both linear models, with respectiv design matrices

$$\mathbb{X}^{[1]} = \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_n \end{bmatrix} \qquad \mathbb{X}^{[1]} = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ \vdots & \vdots & \\ 1 & t_n & t_n^2 \end{bmatrix}$$

The order p polynomial smoothing model

- When the explanatory variable for y_i is the time of measurement, t_i, then we call the linear model a trend.
- When we fit y_i using a function of an arbitrary explanatory variable x_i we say we are **smoothing**.
- We can choose any p in the general order p polynomial smoothing model,

$$[\text{M3}] \quad y_i = b_0 + b_1 x_i + b_2 x_i^2 + b_3 x_i^3 + \cdots + b_p x_i^p + e_i, \quad i = 1, \ldots, n$$

- This is a linear model with design matrix

$$\mathbb{X}^{[3]} = \begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^p \\ 1 & x_2 & x_2^2 & \ldots & x_2^p \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^p \end{bmatrix}$$

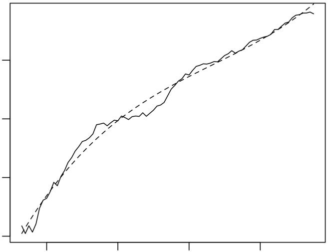
Question 8.14. How would you decide what order p to use for the polynomial smoothing?

Cubic polynomial smoothing of life expectancy

```
L_poly3 <- lm(Total~Year+I(Year^2)+I(Year^3),data=L)
```

```
plot(L$Year,L$Total,
  type="line",
  xlab="Year",
  ylab="Life expectancy")

lines(L$Year,fitted(L_poly3),
  lty="dashed")
```



Question 8.15. Why do we need to write `I(Year^2)` not just `Year^2` to fit a polynomial smoothing model in the R formula notation?

Checking the cubic smoothing calculation

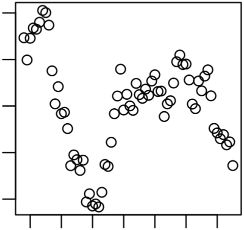
Question 8.16. How would you check that the R model formula we wrote is correct for the cubic polynomial we intend to fit?

Question 8.17. If we have done a good job of modeling the trend, we might hope that the residuals look like independent measurement errors. How would you check if this is the case?

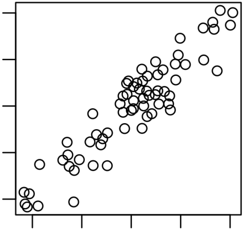
Repeating diagnostic tests for life expectancy vs unemployment using cubic detrending

```r
L_detrended <- L_poly3$residuals
U_annual <- apply(U[,2:13],1,mean)
U_detrended <- lm(U_annual~Year+I(Year^2)+I(Year^3),
  data=U)$residuals
L_detrended <- subset(L_detrended,L$Year %in% U$Year)
lm_poly3 <- lm(L_detrended~U_detrended)
n <- length(resid(lm_poly3))
e <- resid(lm_poly3)[2:n] ; lag_e <- resid(lm_poly3)[1:(n-1)]
```

```r
plot(U$Year,resid(lm_poly3))
```



```r
plot(lag_e,e)
```

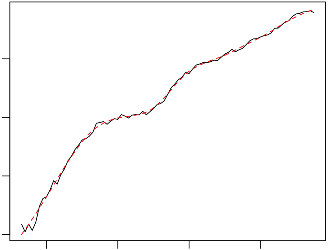


Local linear smoothing of life expectancy

```
L_loess <- loess(Total~Year,data=L,span=0.3)
```

```
plot(L$Year,L$Total,
  type="line",
  xlab="Year",
  ylab="Life expectancy")

lines(L$Year,fitted(L_loess),
  lty="dashed",col="red")
```

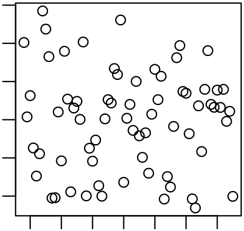


- `loess()` is a **smoother** that fits a local linear model. This means that, at each point x_j, the smoother predicts y_i fitting a linear model that ignores all the data except for points close to x_i.
- Setting `span=0.3` means that the closest 30% of the points are used.

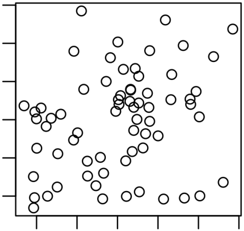
Repeating diagnostic tests for life expectancy vs unemployment using a smoother

```
L_detrended <- resid(L_loess)
U_annual <- apply(U[,2:13],1,mean)
U_detrended <- resid(loess(U_annual~Year,data=U,span=0.3))
L_detrended <- subset(L_detrended,L$Year %in% U$Year)
lm_loess <- lm(L_detrended~U_detrended)
n <- length(resid(lm_loess))
e <- resid(lm_loess)[2:n] ; lag_e <- resid(lm_loess)[1:(n-1)]
```

```
plot(U$Year,resid(lm_loess))
```



```
plot(lag_e,e)
```



Revisiting the evidence for pro-cyclical mortality

```
coef(summary(lm_loess))


##                Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 0.007138079 0.01613621 0.4423641 0.6596720450
## U_detrended 0.067235405 0.01628394 4.1289394 0.0001045733
```

- Recall that linear detrending gave a signficant association between life expectancy and unemployment.
- This suggested that mortality is **pro-cyclical**, meaning it increases when the business cycles is in economic expansion and unemployment is low.
- In Chapter 7, we found the residuals in this regression had a strong pattern, casting doubt on the validity of our linear model and its unintuitive conclusion.

Question 8.18. Re-assess the evidence based on this new analysis.