Regression

Self-test answers

- How is the *t* in Output 7.1 calculated? Use the values in the output to see if you can get the same value as **R**.

It is calculated using this equation:

$$\begin{aligned} t &= \frac{b_{\text{observed}} - b_{\text{expected}}}{SE_b} \\ &= \frac{b_{\text{observed}}}{SE_b} \end{aligned}$$

Using the values from Output 7.1 to calculate *t* for the constant (*t* = 134.10/7.537 = 17.79), for the advertising budget we get 0.09612/0.009632 = 9.979.

- How many units would be sold if we spent £666,000 on advertising the latest album by black metal band Abgott?

A total of 198,080 CDs would be sold:

$$\begin{aligned} \text{album sales}_i &= 134.14 + (0.096 \times \text{advertising budget}_i) \\ &= 134.14 + (0.096 \times 666) \\ &= 198.08 \end{aligned}$$

Labcoat Leni's real research

Why do you like your lecturers?

Problem

Chamorro-Premuzic, T., et al. (2008). *Personality and Individual Differences*, *44,* 965–976.

In the previous chapter we encountered a study by Chamorro-Premuzic et al. in which they measured students' personality characteristics and asked them to rate how much they wanted these same characteristics in their lecturers. In that chapter we correlated these scores; however, we could go a step further and see whether students' personality characteristics predict the characteristics that they would like to see in their lecturers.

The data from this study are in the file **Chamorro-Premuzic.dat**. Labcoat Leni wants you to carry out five multiple regression analyses: the outcome variable in each of the five analyses is how much students want to see neuroticism, extroversion, openness to experience, agreeableness and conscientiousness. For each of these outcomes, force Age and Gender into the analysis in the first step of the hierarchy, then in the second block force in the five student personality traits (Neuroticism, Extroversion, Openness to experience, Agreeableness and Conscientiousness). For each analysis create a table of the results.

Solution

Lecturer neuroticism

First of all, we need to load the data:

```
PersonalityData<-read.delim("Chamorro-Premuzic.dat", header = TRUE)
```

Then we need to create dataframes containing variables for each analysis. We need to do this because there are missing values in the data set and we need to exclude them so that they do not affect the analyses. The lines of code below will tell **R** to drop variables that are not required in each of the individual five regressions. For example, when we are looking at student personality as a predictor of wanting a *neurotic* lecturer, we only need to include the variable 'neurotic lecturer' in the regression analysis – not 'extroverted lecturer' or 'conscientious lecturer', etc. Therefore, we can tell **R** to drop these scores from the analysis.

We can run all these lines of code at once:

```
dropVars<-names(PersonalityData) %in% c("lecturerE","lecturerO", "lecturerA",
"lecturerC")
neuroticLecturer<-PersonalityData[!dropVars]

dropVars<-names(PersonalityData) %in% c("lecturerN","lecturerO", "lecturerA",
"lecturerC")
extroLecturer<-PersonalityData[!dropVars]

dropVars<-names(PersonalityData) %in% c("lecturerE","lecturerN", "lecturerA",
"lecturerC")
openLecturer<-PersonalityData[!dropVars]

dropVars<-names(PersonalityData) %in% c("lecturerE","lecturerO", "lecturerN",
"lecturerC")
agreeLecturer<-PersonalityData[!dropVars]

dropVars<-names(PersonalityData) %in% c("lecturerE","lecturerO", "lecturerA",
"lecturerN")
concLecturer<-PersonalityData[!dropVars]
```

We also need to tell **R** to delete cases with any missing values on any variable:

```
neuroticLecturer <-neuroticLecturer[complete.cases(neuroticLecturer),]
extroLecturer <-extroLecturer[complete.cases(extroLecturer),]
openLecturer <-openLecturer[complete.cases(openLecturer),]
agreeLecturer <-agreeLecturer[complete.cases(agreeLecturer),]
concLecturer <-concLecturer[complete.cases(concLecturer),]
```

The first of the five regressions we'll do is whether students want lecturers to be neurotic. We will create two models: the first, *LecturerN.1*, will have **age** and **gender** as predictors. The second model, *LecturerN.2*, will have all of the five student personality traits (**Neuroticism, Extroversion, Openness to experience**, **Agreeableness** and **Conscientiousness**) as predictors. Remember that the data that we need to tell **R** to use is the *neuroticLecturer* data, not the *PersonalityData*. The *neuroticLecturer* data are a version of the *PersonaliltyData* but with all lecturer variables (except the neurotic lecturer variable) excluded from the analysis and missing cases within the neurotic lecturer variable excluded:

```
LecturerN.1 <- lm(lecturerN ~ Age + Gender, data= neuroticLecturer)
```

```
LecturerN.2 <- lm(lecturerN ~ Age + Gender + studentN + studentE + studentO + studentA
+ studentC, data= neuroticLecturer)
```

To view the output of the two regressions, we can use the *summary()* function:

```
summary(LecturerN.1)
```

```
Call:
lm(formula = lecturerN ~ Age + Gender, data = neuroticLecturer)

Residuals:
    Min      1Q  Median      3Q     Max 
-12.916  -6.627  -1.791   3.652  46.260 
```

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -28.2199     2.5860 -10.913   <2e-16 ***
Age              0.2784     0.1294   2.151   0.0321 *  
Gender[T.Male]   2.4188     1.0230   2.364   0.0186 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 8.774 on 370 degrees of freedom
Multiple R-squared: 0.02785,  Adjusted R-squared: 0.0226 
F-statistic:   5.3 on 2 and 370 DF, p-value: 0.005377
```

```
summary(LecturerN.2)
```

```
Call:
lm(formula = lecturerN ~ Age + Gender + studentN + studentE + 
    studentO + studentA + studentC, data = PersonalityData)

Residuals:
    Min      1Q  Median      3Q     Max 
-12.433  -5.914  -1.791   3.885  46.525 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)   
(Intercept) -16.77420    5.29631  -3.167  0.00167 **
Age           0.30132    0.12808   2.353  0.01917 * 
Gender        1.90321    1.08484   1.754  0.08021 . 
studentN     -0.06017    0.05885  -1.022  0.30723   
studentE     -0.10750    0.07528  -1.428  0.15414   
studentO     -0.17424    0.07286  -2.391  0.01730 * 
studentA      0.08721    0.07158   1.218  0.22391   
studentC     -0.20262    0.08162  -2.482  0.01350 * 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 8.669 on 365 degrees of freedom
  (57 observations deleted due to missingness)
Multiple R-squared: 0.06384,  Adjusted R-squared: 0.04588 
F-statistic: 3.556 on 7 and 365 DF, p-value: 0.001033
```

We can calculate the change in R^2 from the first model (*LecturerN.1*) to the second model (*LecturerN.2*) by subtracting R^2 in model 1 from R^2 in model 2:

0.06384 – 0.02785 = **0.04**

We can use the *anova()* command to compare the two models and obtain the significance for the change in R^2:

```
anova(LecturerN.1, LecturerN.2)
```

```
Analysis of Variance Table

Model 1: lecturerN ~ Age + Gender
Model 2: lecturerN ~ Age + Gender + studentN + studentE + studentO + studentA + 
studentC
  Res.Df   RSS Df   Sum of Sq    F    Pr(>F)  
1    370  28483                              
2    365  27429  5    1054.3   2.806 0.01677 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The significant *F*-statistic informs us that the change in R^2 is significant, i.e. model 2 is a better fit to the data than model 1.

To obtain the standardized beta estimates (sometimes called beta, β_i) we need to use a function called *lm.beta()*. This is found in the *QuantPsyc* package, and so you need to install (if you haven't already) and load this package:

```
install.packages("QuantPsyc")
Library(QuantPsyc)
```

Then we can run it to obtain the standardized beta estimates:

```
lm.beta(LecturerN.1)
```

```
Warning in var(as.vector(x), na.rm = na.rm) :
  NAs introduced by coercion
       Age           Gender[T.Male] 
     0.1103429             NA
```

```
lm.beta(LecturerN.2)

Warning in var(as.vector(x), na.rm = na.rm) :
  NAs introduced by coercion
    Age        Gender[T.Male]    studentN       studentE       studentO       studentA
 0.11942944          NA         -0.05933126    -0.07829448    -0.12270633     0.07276446

 studentC
-0.15651844
```

We can also obtain some statistics, such as the VIF by using the vif() function and Durbin Watson's test using the *dwt()* function:

```
 vif(LecturerN.2)

     Age   Gender studentN studentE studentO studentA studentC
1.004731 1.153441 1.312781 1.171989 1.026678 1.390897 1.549902
```

```
dwt(LecturerN.2)

 lag Autocorrelation   D-W Statistic p-value
   1      0.01733968       1.963358    0.68
 Alternative hypothesis: rho != 0
```

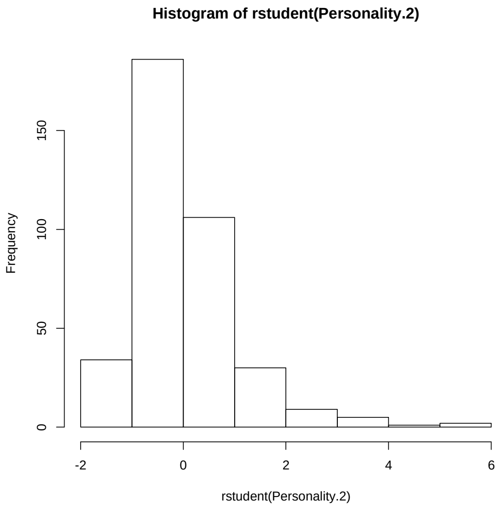
We can obtain the confidence intervals by using the confint() command:

```
confint(LecturerN.2)

                    2.5 %       97.5 %
(Intercept) -27.18931719 -6.35908268
Age           0.04945960  0.55318990
Gender       -0.23011258  4.03652339
studentN     -0.17588793  0.05555007
studentE     -0.25553236  0.04053303
studentO     -0.31752453 -0.03094877
studentA     -0.05355921  0.22797221
studentC     -0.36313181 -0.04211567
```

One useful plot that you should always obtain is the histogram of the residuals (or the standardized or studentized residuals):

```
hist(rstudent(Lecturer.2))
```



You could report these results as follows:

	B	*SE B*	*β*
Step 1			
Constant	−28.22	2.59	
Age	0.28	0.13	0.11*
Gender	2.42	1.02	0.12*

Step 2			
Constant	−16.77	5.30	
Age	0.30	0.13	0.12*
Gender	1.90	1.08	0.10
Neuroticism	−0.06	0.06	−0.06
Extroversion	−0.11	0.08	−0.08
Openness	−0.17	0.07	−0.12*
Agreeableness	0.09	0.07	0.07
Conscientiousness	−0.20	0.08	−0.16*

Note: R^2 = .03 for step 1: ΔR^2 = .04 for step 2 (p < .05). * p < .05.

So basically, age, openness and conscientiousness were significant predictors of wanting a neurotic lecturer (note that for openness and conscientiousness the relationship is negative, i.e. the more a student scored on these characteristics, the *less* they wanted a neurotic lecturer).

Lecturer extroversion

The second variable we want to predict is lecturer extroversion. I will run through the code to run the main regression analysis and report the results, but I will not include the output this time.

First we need to create two models, I have called them *LecturerE.1* and *LecturerE.2.* Remember that the data that we need to tell **R** to use is the *extroLecturer* data, not the *PersonalityData*. The *extroLecturer* data are a version of the *PersonaliltyData* but with all lecturer variables (except the extroverted lecturer variable) excluded from the analysis and missing cases within the extroverted lecturer variable excluded:

```
LecturerE.1 <- lm(lecturerE ~ Age + Gender, data= extroLecturer)

LecturerE.2 <- lm(lecturerE ~ Age + Gender + studentN + studentE + studentO + studentA
+ studentC, data= extroLecturer)
```

Obtain your output:

```
summary(LecturerE.1)

summary(LecturerE.2)
```

Run an ANOVA to compare the two models:

```
anova(LecturerE.1, LecturerE.2)
```

Standardized beta estimates:

```
lm.beta(LecturerE.1)

lm.beta(LecturerE.2)
```

You could report these results as follows:

	B	*SE B*	β
Step 1			
Constant	11.82	2.28	
Age	0.04	0.11	0.02
Gender	1.12	0.94	0.07
Step 2			
Constant	2.03	4.77	
Age	0.01	0.11	0.00
Gender	1.58	1.01	0.10
Neuroticism	0.02	0.06	0.02
Extroversion	0.16	0.07	0.16*
Openness	0.05	0.07	0.04
Agreeableness	0.01	0.06	0.01
Conscientiousness	0.11	0.08	0.11

Note. R^2 = .01 for step 1: ΔR^2 = .04 for step 2 (p > .05). * p < .05.

So basically, student extroversion was the only significant predictor of wanting an extrovert lecturer; the model overall did not explain a significant amount of the variance in wanting an extroverted lecturer.

Lecturer openness to experience

The third variable we want to predict is lecturer openness to experience. As before, I will run through the code to run the main regression analysis and report the results, but I will not include the output.

First we need to create two models, I have called them *LecturerO.1* and *LecturerO.2.* Remember that the data that we need to tell **R** to use are the *openLecturer* data, not the *PersonalityData*. The *openLecturer* data are a version of the *PersonaliltyData* but with all lecturer variables (except the openness to experience lecturer variable) excluded from the analysis and missing cases within the openness to experience lecturer variable excluded:

```
LecturerO.1 <- lm(lecturerO ~ Age + Gender, data= openLecturer)

LecturerO.2 <- lm(lecturerO ~ Age + Gender + studentN + studentE + studentO + studentA
+ studentC, data= openLecturer)
```

Obtain your output:

```
summary(LecturerO.1)

summary(LecturerO.2)
```

Run an ANOVA to compare the two models:

```
anova(LecturerO.1, LecturerO.2)
```

Standardized beta estimates:

```
lm.beta(LecturerO.1)

lm.beta(LecturerO.2)
```

You could report these results as:

	B	*SE B*	*β*
Step 1			
Constant	9.35	2.37	
Age	−0.03	0.12	−0.01
Gender	0.11	0.94	0.01
Step 2			
Constant	−5.51	4.83	
Age	−0.04	0.12	−0.02
Gender	−0.22	0.99	−0.01
Neuroticism	0.01	0.05	0.01
Extroversion	0.07	0.07	0.05
Openness	0.28	0.07	0.22***
Agreeableness	0.15	0.07	0.13*
Conscientiousness	−0.06	0.07	−0.05

Note: R^2 = .00 for step 1 (*ns*): ΔR^2 = .06 for step 2 (p < .001). * p < .05, *** p < .001.

So basically, student openness to experience was the most significant predictor of wanting a lecturer who is open to experiences, but student agreeableness predicted this also.

Lecturer agreeableness

The fourth variable we want to predict is lecturer agreeableness. As before, I will run through the code to run the main regression analysis and report the results, but I will not include the output.

First we need to create two models, I have called them *LecturerA.1* and *LecturerA.2.* Remember that the data that we need to tell **R** to use are the *agreeLecturer* data, not the *PersonalityData*. The *oagreeLecturer* data are a version of the *PersonaliltyData* but with all lecturer variables (except the agreeableness lecturer variable) excluded from the analysis and missing cases within the agreeableness lecturer variable excluded:

```
LecturerA.1 <- lm(lecturerA ~ Age + Gender, data= agreeLecturer)

LecturerA.2 <- lm(lecturerA ~ Age + Gender + studentN + studentE + studentO + studentA
+ studentC, data= agreeLecturer)
```

Obtain your output:

```
summary(LecturerA.1)

summary(LecturerA.2)
```

Run an ANOVA to compare the two models:

```
anova(LecturerA.1, LecturerA.2)
```

Standardized beta estimates:

```
lm.beta(LecturerA.1)

lm.beta(LecturerA.2)
```

You could report these results as follows:

	B	*SE B*	*β*
Step 1			
Constant	18.19	2.79	
Age	−0.47	0.14	−0.17***
Gender	−0.76	1.09	−0.04
Step 2			
Constant	7.04	5.61	
Age	−0.48	0.14	−0.17***
Gender	1.00	1.15	0.05
Neuroticism	0.17	0.06	0.16**
Extroversion	0.06	0.08	0.04
Openness	−0.21	0.08	−0.14**
Agreeableness	0.17	0.08	0.13*
Conscientiousness	0.10	0.09	0.07

Note: R^2 = .03 for step 1 (p < .01): ΔR^2 = .97 for step 2 (p < .001). * p < .05, ** p < .01, *** p < .001

Age, student openness to experience, student neuroticism and student agreeableness significantly predicted wanting a lecturer who is agreeable. Age and openness to experience had negative relationships (the older and more open to experienced you are, the less you want an agreeable lecturer), whereas as student neuroticism increases so does the desire for an agreeable lecturer (not surprisingly, because neurotics will lack confidence and probably feel more able to ask an agreeable lecturer questions).

Lecturer conscientiousness

The final variable we want to predict is lecturer conscientiousness. As before, I will run through the code to run the main regression analysis and report the results, but I will not include the output.

First we need to create two models, I have called them *LecturerC.1* and *LecturerC.2.* Remember that the data that we need to tell **R** to use are the *concLecturer* data, not the *PersonalityData*. The *concLecturer* data are a version of the *PersonaliltyData* but with all lecturer variables (except the lecturer conscientiousness variable) excluded from the analysis and missing cases within the lecturer conscientiousness variable excluded:

```
LecturerC.1 <- lm(lecturerC ~ Age + Gender, data= concLecturer)

LecturerC.2 <- lm(lecturerC ~ Age + Gender + studentN + studentE + studentO + studentA
+ studentC, data= concLecturer)
```

Obtain your output:

```
summary(LecturerC.1)

summary(LecturerC.2)
```

Run an ANOVA to compare the two models:

```
anova(LecturerC.1, LecturerC.2)
```

Standardized beta estimates:

```
lm.beta(LecturerC.1)

lm.beta(LecturerC.2)
```

You could report these results as follows:

	B	*SE B*	*β*
Step 1			
Constant	14.97	2.18	
Age	0.12	0.11	0.06
Gender	−2.28	0.87	−0.14**
Step 2			
Constant	6.36	4.43	
Age	0.10	0.11	0.05
Gender	−1.56	0.91	−0.09
Neuroticism	−0.01	0.05	−0.01
Extroversion	−0.07	0.06	−0.06
Openness	−0.01	0.06	−0.01
Agreeableness	0.15	0.06	0.14*
Conscientiousness	0.14	0.07	0.13*

Note: R^2 = .02 for step 1 (p < .05): ΔR^2 = .05 for step 2 (p < .01). * p < .05, ** p < .01.

Student agreeableness and conscientiousness both predicted wanting a lecturer who is conscientious. Note also that gender predicted this in the first step, but its *b* became non-significant when the student personality variables were forced in as well. However, gender is probably a variable that should be explored further within this context.

Compare your results to Table 4 in the actual article. I've highlighted the area of the table relating to our analyses (our five analyses are represented by the columns labelled N, E, O, A and C).

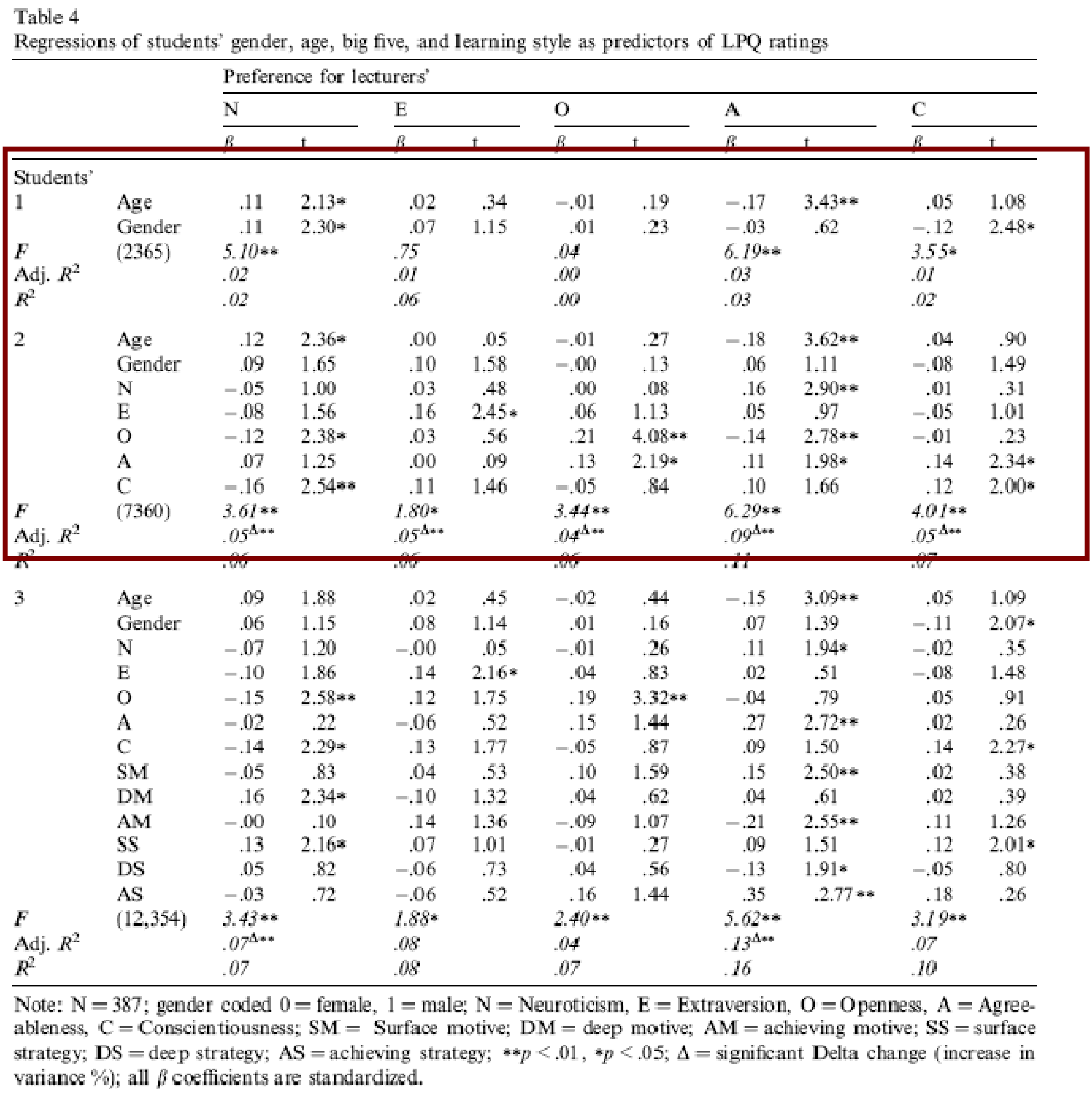


Table 4
Regressions of students' gender, age, big five, and learning style as predictors of LPQ ratings

		Preference for lecturers'									
		N		E		O		A		C	
		β	t	β	t	β	t	β	t	β	t
Students'											
1	Age	.11	2.13*	.02	.34	−.01	.19	−.17	3.43**	.05	1.08
	Gender	.11	2.30*	.07	1.15	.01	.23	−.03	.62	−.12	2.48*
F	(2365)	*5.10***		*.75*		*.04*		*6.19***		*3.55**	
Adj. R^2		*.02*		*.01*		*.00*		*.03*		*.01*	
R^2		*.02*		*.06*		*.00*		*.03*		*.02*	
2	Age	.12	2.36*	.00	.05	−.01	.27	−.18	3.62**	.04	.90
	Gender	.09	1.65	.10	1.58	−.00	.13	.06	1.11	−.08	1.49
	N	−.05	1.00	.03	.48	.00	.08	.16	2.90**	.01	.31
	E	−.08	1.56	.16	2.45*	.06	1.13	.05	.97	−.05	1.01
	O	−.12	2.38*	.03	.56	.21	4.08**	−.14	2.78**	−.01	.23
	A	.07	1.25	.00	.09	.13	2.19*	.11	1.98*	.14	2.34*
	C	−.16	2.54**	.11	1.46	−.05	.84	.10	1.66	.12	2.00*
F	(7360)	*3.61***		*1.80**		*3.44***		*6.29***		*4.01***	
Adj. R^2		*.05*$^{\Delta**}$		*.05*$^{\Delta**}$		*.04*$^{\Delta**}$		*.09*$^{\Delta**}$		*.05*$^{\Delta**}$	
R^2		*.06*		*.06*		*.06*		*.11*		*.07*	
3	Age	.09	1.88	.02	.45	−.02	.44	−.15	3.09**	.05	1.09
	Gender	.06	1.15	.08	1.14	.01	.16	.07	1.39	−.11	2.07*
	N	−.07	1.20	−.00	.05	−.01	.26	.11	1.94*	−.02	.35
	E	−.10	1.86	.14	2.16*	.04	.83	.02	.51	−.08	1.48
	O	−.15	2.58**	.12	1.75	.19	3.32**	−.04	.79	.05	.91
	A	−.02	.22	−.06	.52	.15	1.44	.27	2.72**	.02	.26
	C	−.14	2.29*	.13	1.77	−.05	.87	.09	1.50	.14	2.27*
	SM	−.05	.83	.04	.53	.10	1.59	.15	2.50**	.02	.38
	DM	.16	2.34*	−.10	1.32	.04	.62	.04	.61	.02	.39
	AM	−.00	.10	.14	1.36	−.09	1.07	−.21	2.55**	.11	1.26
	SS	.13	2.16*	.07	1.01	−.01	.27	.09	1.51	.12	2.01*
	DS	.05	.82	−.06	.73	.04	.56	−.13	1.91*	−.05	.80
	AS	−.03	.72	−.06	.52	.16	1.44	.35	2.77 **	.18	.26
F	(12,354)	*3.43***		*1.88**		*2.40***		*5.62***		*3.19***	
Adj. R^2		*.07*$^{\Delta**}$		*.08*		*.04*		*.13*$^{\Delta**}$		*.07*	
R^2		*.07*		*.08*		*.07*		*.16*		*.10*	

Note: N = 387; gender coded 0 = female, 1 = male; N = Neuroticism, E = Extraversion, O = Openness, A = Agreeableness, C = Conscientiousness; SM = Surface motive; DM = deep motive; AM = achieving motive; SS = surface strategy; DS = deep strategy; AS = achieving strategy; $**p < .01$, $*p < .05$; Δ = significant Delta change (increase in variance %); all β coefficients are standardized.

Smart Alex's solutions

Task 1

- Run a simple regression for the **pubs.dat** data in Jane Superbrain Box 7.1, predicting **mortality** from number of **pubs.** Try repeating the analysis but bootstrapping the regression parameters.

First load the data file by setting your working directory to the location of the file (see section 3.4.4) and executing:

```
pubs<-read.delim("pubs.dat", header = TRUE)
```

Then, by executing

```
pubs
```

we can view the data set:

```
       pubs  mortality
1       10   1000
2       20   2000
3       30   3000
4       40   4000
5       50   5000
6       60   6000
7       70   7000
8       500  10000
```

We run a regression analysis using the *lm()* function:

```
pubs.1 <- lm(mortality ~ pubs, data = pubs)
```

We have created an object called *pubs.1* that contains the results of our analysis. We can show the object by executing:

```
summary(pubs.1)
```

This displays the information in the output below:

```
Call:
lm(formula = mortality ~ pubs, data = pubs)

Residuals:
    Min      1Q  Median      3Q     Max 
-2495.3  -996.3  -223.5  1145.2  2644.3 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept) 3351.955    781.236   4.291  0.00515 **
pubs          14.339      4.301   3.334  0.01572 * 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1864 on 6 degrees of freedom
Multiple R-squared: 0.6495,   Adjusted R-squared: 0.591 
F-statistic: 11.12 on 1 and 6 DF, p-value: 0.01572
```

For these data, *F* is 11.12, which is significant at *p* < .05 (because the *p*-value given is less than .05). Therefore, we can conclude that the number of pubs significantly predicts mortality.

Next we want to repeat the analysis but bootstrapping the regression parameters. To do this, first make sure you have executed the *bootReg()* function from the book chapter. We can then use the function to obtain the bootstrap samples:

```
bootResults<-boot(statistic = bootReg, formula = mortality ~ pubs, data = pubs, R = 
2000)
```

Executing this command creates an object called *bootResults* that contains the bootstrap samples. We use the *boot()* function to get these. Instead of one statistic, we need to obtain bootstrap confidence intervals for the intercept, and the slope for pubs. We can do this with the *boot.ci()* function that we encountered in Chapter 6. However, **R** doesn't know the names of the statistics in *bootResults,* so we instead have to use their location in the *bootResults* object (because **R** does know this information). The intercept is the first thing in *bootResults,* so to obtain the bootstrapped confidence interval for the intercept we use index = 1:

```
boot.ci(bootResults, type = "bca", index = 1)
```

The location of the coefficient for **pubs** is given by index = 2, so we can get the bootstrap confidence intervals for this predictor by executing:

```
boot.ci(bootResults, type = "bca", index = 2)
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 2000 bootstrap replicates

CALL :
```

```
boot.ci(boot.out = bootResults, type = "bca", index = 1)

Intervals :
Level       BCa
95%   (0, 6238)
Calculations and Intervals on Original Scale

> boot.ci(bootResults, type = "bca", index = 2)

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 2000 bootstrap replicates

CALL :
boot.ci(boot.out = bootResults, type = "bca", index = 2)

Intervals :
Level       BCa
95%   (7.53, 100.00)
```

The output above shows the confidence limitss for the **intercept** are 0, 6238 (remember that because of how bootstrapping works, you won't get exactly the same result as me, but it should be very close). The bootstrap confidence limits for the predictor **pubs** are 7.53, 100.00. These values do not cross zero, indicating that number of pubs still has a significant effect on mortality when using a robust method.

Task 2

- A fashion student was interested in factors that predicted the salaries of catwalk models. She collected data from 231 models. For each model she asked them their salary per day on days when they were working (**salary**), their age (**age**), how many years they had worked as a model (**years**), and then got a panel of experts from modelling agencies to rate the attractiveness of each model as a percentage, with 100% being perfectly attractive (**beauty**). The data are in the file **Supermodel.dat**. Unfortunately, this fashion student bought some substandard statistics text and so doesn't know how to analyse her data.☺ Can you help her out by conducting a multiple regression to see which factor predict a model's salary? How valid is the regression model?

First, load in the **Supermodel.dat** data:

```
Supermodel<-read.delim("Supermodel.dat", header = TRUE)
```

Then create a regression model (I have called the model *Supermodel.1*) to predict salary from age, number of years being a supermodel and beauty:

```
Supermodel.1 <- lm(salary~age + beauty + years, data= Supermodel)
```

Obtain the output of the regression model:

```
summary(Supermodel.1)

Call:
lm(formula = salary ~ age + beauty + years, data = Supermodel)

Residuals:
    Min      1Q  Median      3Q     Max 
-24.853  -7.950  -4.197   4.605  68.085 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -60.8897    16.4966  -3.691 0.000280 ***
age           6.2344     1.4112   4.418 1.54e-05 ***
beauty       -0.1964     0.1524  -1.289 0.198711    
years        -5.5612     2.1222  -2.621 0.009372 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 14.57 on 227 degrees of freedom
Multiple R-squared: 0.184,    Adjusted R-squared: 0.1733
F-statistic: 17.07 on 3 and 227 DF, p-value: 4.973e-10
```

To begin with, a sample size of 231 with three predictors seems reasonable because this would easily detect medium to large effects (see the diagram in the chapter).

Overall, the model accounts for 18.4% of the variance in salaries and is a significant fit to the data (*F*(3, 227) = 17.07, *p* < .001). The adjusted R^2 (.17) shows some shrinkage from the unadjusted value (.184), indicating that the model may not generalize well. We can also use Stein's formula:

$$\begin{aligned}\text{adjusted } R^2 &= 1-\left[\left(\frac{231-1}{231-3-1}\right)\left(\frac{231-2}{231-3-2}\right)\left(\frac{231+1}{231}\right)\right](1-0.184)\\ &= 1-[1.031](0.816)\\ &= 1-0.841\\ &= 0.159\end{aligned}$$

This also shows that the model may not cross-generalize well.

Next we can obtain the standardized beta estimates:

```
lm.beta(Supermodel.1)

     age       beauty       years
 0.94214234 -0.08299604 -0.54779846
```

In terms of the individual predictors we could report the following:

	B	SE B	β
Constant	–60.89	16.50	
Age	6.23	1.41	0.94**
Years as a model	–5.56	2.12	–0.55*
Attractiveness	–0.20	0.15	–0.08

Note: R^2 = .18 (*p* < .001). * *p* < .01, ** *p* < .001.

It seems as though salaries are significantly predicted by the age of the model. This is a positive relationship (look at the sign of the beta), indicating that as age increases, salaries increase too. The number of years spent as a model also seems to significantly predict salaries, but this is a negative relationship indicating that the more years you've spent as a model, the lower your salary. This finding seems very counter-intuitive, but we'll come back to it later. Finally, the attractiveness of the model doesn't seem to predict salaries.

If we wanted to write the regression model, we could write it as:

$$\begin{aligned}\text{Salary} &= \beta_0 + \beta_1\text{Age}_i + \beta_2\text{Experience}_i + \beta_3\text{Attractiveness}_i\\ &= -60.89 + (6.23\text{Age}_i) - (5.56\text{Experience}_i) - (0.02\text{Attractiveness}_i)\end{aligned}$$

The next part of the question asks whether this model is valid.

```
 dwt(Supermodel.1)
lag Autocorrelation D-W Statistic p-value
   1     -0.03061432      2.057416   0.724
 Alternative hypothesis: rho != 0

vif(Supermodel.1)
age          beauty     years
12.652841  1.153364   12.156757

1/vif(Supermodel.1)
  age        beauty      years
0.07903364 0.86702902 0.08225878
```

We can obtain the casewise diagnostics. However, if we just obtain them, **R** will print a long list for us. This won't be very useful. Instead, we'll store them, and that way we can look at them more easily. To make our life a little easier, we'll add them to the *Supermodel.1* data set:

```
Supermodel$cooks.distance<-cooks.distance(Supermodel.1)
Supermodel$residuals<-resid(Supermodel.1)
```

```
Supermodel$standardized.residuals <- rstandard(Supermodel.1)
Supermodel$studentized.residuals <- rstudent(Supermodel.1)
Supermodel$dfbeta <- dfbeta(Supermodel.1)
Supermodel$dffit <- dffits(Supermodel.1)
Supermodel$leverage <- hatvalues(Supermodel.1)
Supermodel$covariance.ratios <- covratio(Supermodel.1)
```

List of standardized residuals greater than 2:

```
Supermodel$standardized.residuals>2| Supermodel$standardized.residuals < -2
```

Create a variable called *large.residual*, which is TRUE (or 1) if the residual is greater than 2 or less than –2:

```
Supermodel$large.residual <- Supermodel$standardized.residuals > 2|
Supermodel$standardized.residuals < -2
```

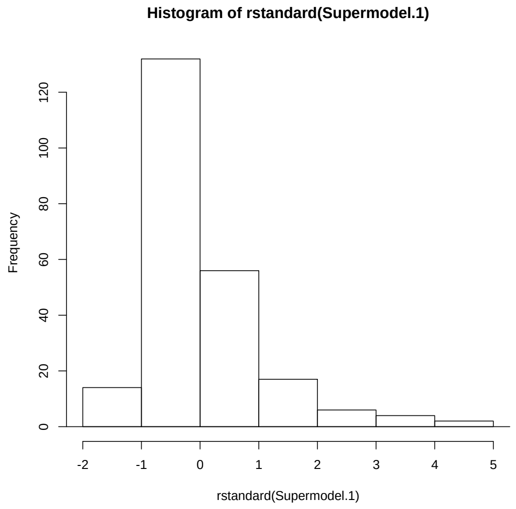
Now we have a variable that we can use. To use it, it is useful to remember that R stores 'TRUE' as 1, and 'FALSE' as 0. Because of that, we can get the sum of the variable *large.residual*, and this will be the number of cases with a large residual:

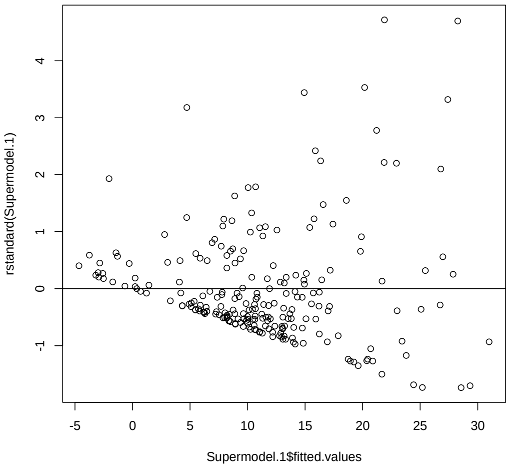
```
sum(Supermodel$large.residual)
[1] 12
```

It might be better not just to know how many cases there are, but also which cases they are. We can look at the values of the residuals by selecting only those cases where the residual is outside of the range from –2 to +2. And we can see the values of some of the variables, using the same approach. We only want to show certain cases – those in which *large.residual* is equal to TRUE. So we use:

```
Supermodel[,c("salary", "age", "beauty", "years", "standardized.residuals")]
```

```
     salary    age      beauty   years           standardized.residuals
2   53.72479 20.34707 68.56999 5.506886                 2.214829
5   95.33807 24.17183 71.77039 8.532050                 4.696607
24  48.86766 19.11451 73.32626 4.951027                 2.241876
41  51.02516 19.46200 80.00141 5.187275                 2.420635
91  56.83151 24.41146 80.65103 8.753041                 2.099147
116 64.79129 18.46839 78.91763 4.284322                 3.440027
127 61.31880 22.25275 78.92917 7.397138                 2.778123
135 89.98003 22.28899 75.93018 7.419825                 4.717284
155 74.86075 24.40682 86.09212 8.444767                 3.319137
170 54.56552 22.31422 88.01470 6.833367                 2.200115
191 50.65578 15.27406 66.38544 2.981697                 3.177863
198 71.32073 20.65061 77.57684 5.834559                 3.531357
```

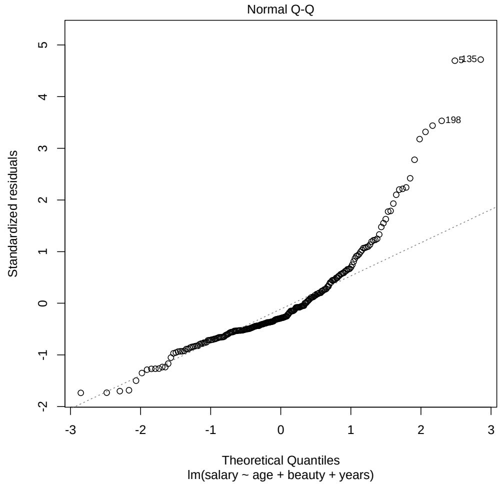

Histogram of rstandard(Supermodel.1)

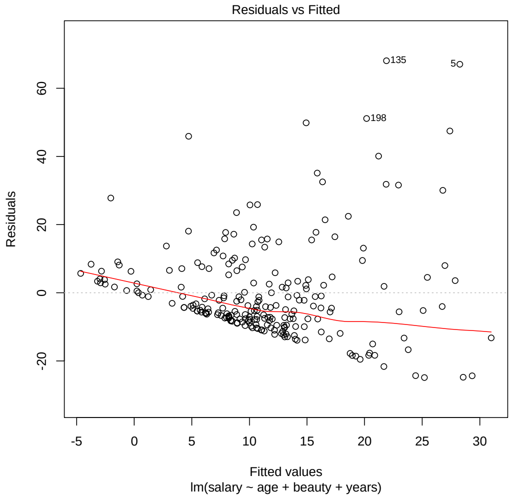


We can get some other plots by using *plot():*

```
plot(Supermodel.1)
```

A useful plot is a normal Q-Q plot:





Residuals: There are six cases that have a standardized residual greater than 3, and two of these are fairly substantial (case 5 and 135). We have 5.19% of cases with standardized residuals above 2, so that's as we expect, but 3% of cases with residuals above 2.5 (we'd expect only 1%), which indicates possible outliers.

Normality of errors: The histogram reveals a skewed distribution, indicating that the normality of errors assumption has been broken. The normal Q–Q plot verifies this because there is a large amount of deviation from the straight line.

Homoscedasticity and independence of errors: The scatterplot of the standardized residuals does not show a random pattern. There is a distinct funnelling, indicating heteroscedasticity. However, the Durbin–Watson statistic does fall within Field's recommended boundaries of 1–3, which suggests that errors are reasonably independent.

Multicollinearity: For the age and experience variables in the model, VIF values are above 10 (or alternatively, tolerance values are all well below 0.1), indicating multicollinearity in the data. It is possible that these two variables are highly correlated and therefore are measuring very similar things. Of course, this makes perfect sense because the older a model is, the more years she would've spent modelling! So, it was fairly stupid to measure both of these things! This also explains the weird result that the number of years spent modelling negatively predicted salary (i.e. more experience = less salary!): in fact if you do a simple regression with experience as the only predictor of salary you'll find it has the expected positive relationship. This hopefully demonstrates why multicollinearity can bias the regression model.

All in all, several assumptions have not been met and so this model is probably fairly unreliable.

Task 3

- Using the Glastonbury data from this chapter (with the dummy coding in **GlastonburyDummy.dat**), which you should've already analysed, comment on whether you think the model is reliable and generalizable.

This question asks whether this model is valid. The model would be:

```
gfr.1 <- lm(gfr$change ~ gfr$crusty + gfr$metaller + gfr$indie.kid, data=gfr)
```

However, the problem is that when you create the *gfr.1* model, people with missing data are excluded. Only 123 cases go into the model (due to a large amount of missing data). This is only really a problem when trying to get the residuals; the residuals is a vector that's 123 items long, and when you try to make that into a variable, it doesn't match. Therefore, we need to include:

```
na.action = na.exclude
```

in the model, so the model becomes:

```
gfr.1 <- lm(gfr$change ~ gfr$crusty + gfr$metaller + gfr$indie.kid, data=gfr,
na.action=na.exclude)
```

To view the output:

```
summary(gfr.1)

Call:
lm(formula = gfr$change ~ gfr$crusty + gfr$metaller + gfr$indie.kid,
    data = gfr, na.action = na.exclude)

Residuals:
     Min       1Q   Median       3Q      Max
-1.82569 -0.50489  0.05593  0.42430  1.59431

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       -0.55431    0.09036  -6.134 1.15e-08 ***
gfr$crustyTRUE    -0.41152    0.16703  -2.464   0.0152 *
gfr$metallerTRUE   0.02838    0.16033   0.177   0.8598
gfr$indie.kidTRUE -0.40998    0.20492  -2.001   0.0477 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6882 on 119 degrees of freedom
  (687 observations deleted due to missingness)
Multiple R-squared: 0.07617,  Adjusted R-squared: 0.05288
F-statistic:  3.27 on 3 and 119 DF,  p-value: 0.02369
```

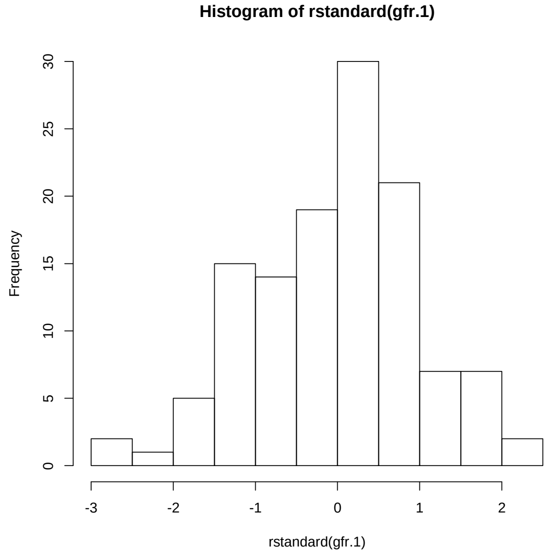
```
vif(gfr.1)

gfr$crusty  gfr$metaller gfr$indie.kid
     1.137931      1.143818      1.100084
```

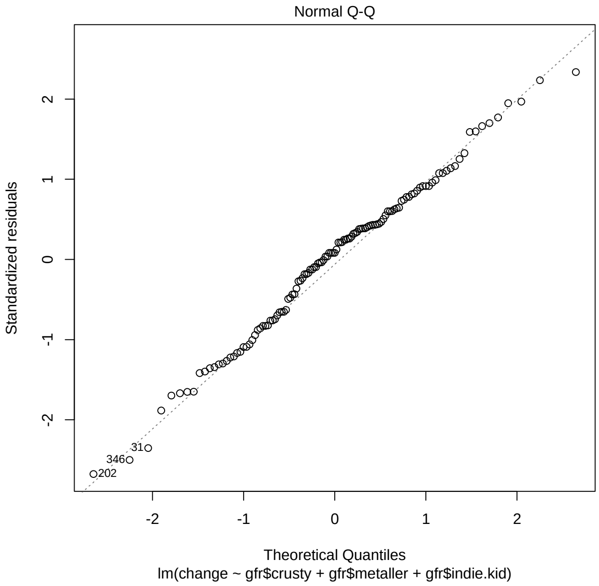
```
1/vif(gfr.1)
```

```
gfr$crusty  gfr$metaller gfr$indie.kid
    0.8787879     0.8742647     0.9090214
```

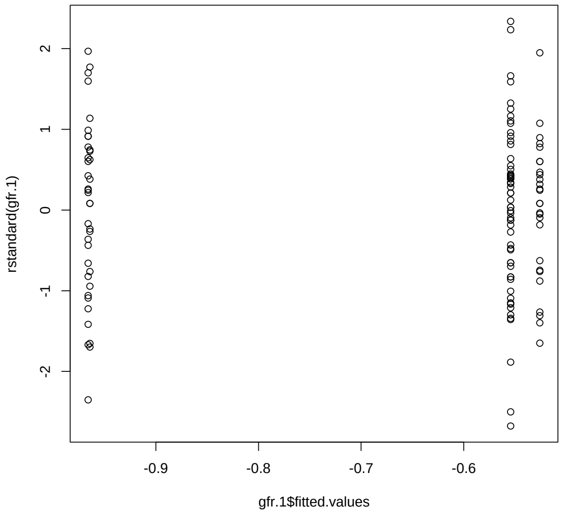
```
hist(rstandard(gfr.1))
```



```
plot(gfr.1)
```



```
plot(gfr.1$fitted.values,rstandard(gfr.1))
```



I will include the code for obtaining the residuals, but I will not put in the output as it is rather large. Obtain casewise diagnostics and add them to the original data:

```
gfr$cooks.distance<-cooks.distance(gfr.1)
gfr$residuals<-resid(gfr.1)
gfr$standardized.residuals<-rstandard(gfr.1)
gfr$studentized.residuals<-rstudent(gfr.1)
gfr$dfbeta<-dfbeta(gfr.1)
gfr$dffit<-dffits(gfr.1)
gfr$leverage<-hatvalues(gfr.1)
gfr$covariance.ratios<-covratio(gfr.1)
```

List of standardized residuals greater than 2:

```
gfr$standardized.residuals>2| gfr$standardized.residuals < -2
```

Create a variable called *large.residual*, which is TRUE (or 1) if the residual is greater than 2 or less than –2:

```
gfr$large.residual <- gfr$standardized.residuals > 2| gfr$standardized.residuals < -2
```

Count the number of large residuals:

```
sum(gfr$large.residual)
```

The Durbin–Watson statistic cannot be calculated if there are missing values in the data set and so you will need to use the model:

```
gfr.1 <- lm(gfr$change ~ gfr$crusty + gfr$metaller + gfr$indie.kid, data=gfr)


dwt(gfr.1)
```

```
lag Autocorrelation D-W Statistic p-value
   1      0.04948997      1.893407   0.558
 Alternative hypothesis: rho != 0
```

Residuals: There are no cases that have a standardized residual greater than 3. We have 4.07% of cases with standardized residuals above 2, and 0.81% of cases with residuals above 2.5 (and we'd expect 1%), so the data are consistent with what we'd expect.

Normality of errors: The histogram looks reasonably normally distributed. indicating that the normality of errors assumption has probably been met. The normal Q-Q plot verifies this because the dashed line doesn't deviate much from the plots (which indicates what you'd get from normally distributed errors).

Homoscedasticity and independence of errors: The scatterplot of ZPRED vs. ZRESID does look a bit odd with categorical predictors, but essentially we're looking for the height of the lines to be about the same (indicating the variability at each of the three levels is the same). This is true, indicating homoscedasticity. The Durbin–Watson statistic also falls within Field's recommended boundaries of 1–3, which suggests that errors are reasonably independent.

Multicollinearity: For all variables in the model, VIF values are below 10 (or alternatively, tolerance values are all well above 0.1) indicating no multicollinearity in the data.

All in all, the model looks fairly reliable (but you should check for influential cases!).

Task 4

- A study was carried out to explore the relationship between aggression and several potential predicting factors in 666 children who had an older sibling. Variables measured were **Parenting_Style** (high score = bad parenting practices), **Computer_Games** (high score = more time spent playing computer games), **Television** (high score = more time spent watching television), **Diet** (high score = the child has a good diet low in additives), and **Sibling_Aggression** (high score = more aggression seen in their older sibling). Past research indicated that parenting style and sibling aggression were good predictors of the level of aggression in the younger child. All other variables were treated in an exploratory fashion. The data are in the file **Child Aggression.dat**. Analyse them with multiple regression.

We need to conduct this analysis hierarchically, entering parenting style and sibling aggression in the first step (forced entry) and the remaining variables in a second step.

First load the **ChildAggression.dat** file:

```
ChildAggression<-read.delim("ChildAggression.dat", header = TRUE)
```

Create two regression models:

```
ChildAggression.1<-lm(Aggression ~ Sibling_Aggression + Parenting_Style, data =
ChildAggression)


ChildAggression.2<-lm(Aggression ~ Sibling_Aggression+Parenting_Style+ Diet +
Computer_Games + Television, data=ChildAggression)
```

Then view the output of the two models:

```
summary(ChildAggression.1)

Call:
lm(formula = Aggression ~ Sibling_Aggression + Parenting_Style,
    data = ChildAggression)

Residuals:
     Min       1Q   Median       3Q      Max
-1.09755 -0.17180  0.00092  0.15405  1.23037

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        -0.005784   0.012065  -0.479    0.632
Sibling_Aggression  0.093409   0.037505   2.491    0.013 *
Parenting_Style     0.061984   0.012257   5.057 5.51e-07 ***
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3113 on 663 degrees of freedom
Multiple R-squared: 0.05325,  Adjusted R-squared: 0.05039
F-statistic: 18.64 on 2 and 663 DF,  p-value: 1.325e-08
```

```
summary(ChildAggression.2)
```

```
Call:
lm(formula = Aggression ~ Sibling_Aggression + Parenting_Style +
    Diet + Computer_Games + Television, data = ChildAggression)

Residuals:
     Min       1Q   Median       3Q      Max
-1.12629 -0.15253 -0.00421  0.15222  1.17669

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        -0.004988   0.011983  -0.416 0.677350
Sibling_Aggression  0.081684   0.038780   2.106 0.035550 *
Parenting_Style     0.056648   0.014557   3.891 0.000110 ***
Diet               -0.109054   0.038076  -2.864 0.004315 **
Computer_Games      0.142161   0.036920   3.851 0.000129 ***
Television          0.032916   0.046057   0.715 0.475059
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.3071 on 660 degrees of freedom
Multiple R-squared: 0.08258,  Adjusted R-squared: 0.07563
F-statistic: 11.88 on 5 and 660 DF,  p-value: 5.025e-11
```

Obtain the standardized parameter estimates with the *lm.beta()* function:

```
lm.beta(ChildAggression.1)

Sibling_Aggression    Parenting_Style
        0.09557412         0.19406149

lm.beta(ChildAggression.2)

Sibling_Aggression    Parenting_Style      Diet         Computer_Games
   0.08357717            0.17735588      -0.11503080     0.15211518

Television
0.03192490
```

Compare the values of R^2 in two models using the ANOVA command:

```
anova(ChildAggression.1, ChildAggression.2)

Analysis of Variance Table

Model 1: Aggression ~ Sibling_Aggression + Parenting_Style
Model 2: Aggression ~ Sibling_Aggression + Parenting_Style + Diet + Computer_Games +
    Television
  Res.Df   RSS Df Sum of Sq      F    Pr(>F)
1    663 64.23
2    660 62.24  3    1.9900 7.0339 0.0001166 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Statistics:

```
vif(ChildAggression.1)

Sibling_Aggression    Parenting_Style
          1.031210           1.031210

1/vif(ChildAggression.1)

Sibling_Aggression    Parenting_Style
         0.9697344          0.9697344

vif(ChildAggression.2)

Sibling_Aggression    Parenting_Style               Diet     Computer_Games
```

```
          1.132618           1.494296           1.160466           1.122719

 Television
 1.435525
```

```
1/vif(ChildAggression.2)
```

```
Sibling_Aggression    Parenting_Style               Diet     Computer_Games
         0.8829104          0.6692115          0.8617231          0.8906946
        Television
         0.6966095
```

```
durbinWatsonTest(ChildAggression.1)
```

```
lag Autocorrelation D-W Statistic p-value
   1      0.05300815      1.890118   0.168
 Alternative hypothesis: rho != 0
```

```
dwt(ChildAggression.2)
```

```
lag Autocorrelation D-W Statistic p-value
   1      0.04218005      1.912808    0.24
 Alternative hypothesis: rho != 0
```

Obtain casewise diagnostics and add them to the original data:

```
ChildAggression$cooks.distance<-cooks.distance(ChildAggression.2)
ChildAggression$residuals<-resid(ChildAggression.2)
ChildAggression$standardized.residuals <- rstandard(ChildAggression.2)
ChildAggression$studentized.residuals <- rstudent(ChildAggression.2)
ChildAggression$dfbeta <- dfbeta(ChildAggression.2)
ChildAggression$dffit <- dffits(ChildAggression.2)
ChildAggression$leverage <- hatvalues(ChildAggression.2)
ChildAggression$covariance.ratios <- covratio(ChildAggression.2)
```

Get a list of standardized residuals greater than 2:

```
ChildAggression$standardized.residuals>2| ChildAggression$standardized.residuals < -2
```

I won't put the output of this list in here as it is a long list!

Create a variable called *large.residual*, which is TRUE (or 1) if the residual is greater than 2 or less than –2:

```
ChildAggression$large.residual <- ChildAggression$standardized.residuals > 2|
ChildAggression$standardized.residuals < -2
```

Count the number of large residuals:

```
sum(ChildAggression$large.residual)
```

```
[1] 37
```

If we want to display only some of the variables we can use:

```
ChildAggression[,c("Aggression",
"Sibling_Aggression","Parenting_Style","Diet","Computer_Games", "Television",
"standardized.residuals")]
```

Display the value of **Aggression**, **Parenting_Style**, **Diet**, **Computer_Games** and **Television** and the standardized residual, for those cases which have a residual greater than 2 or less than –2:

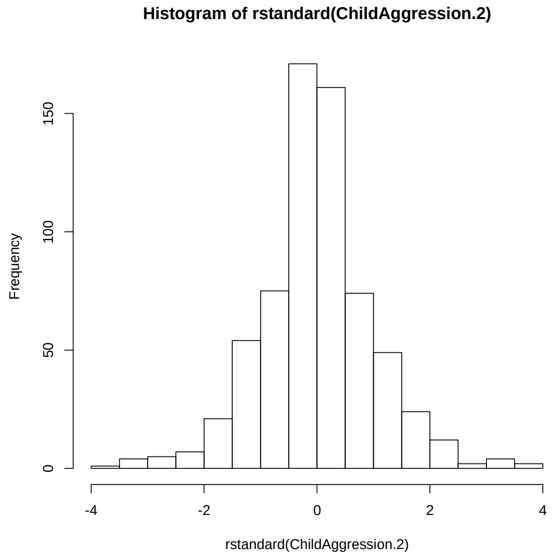
```
ChildAggression[ChildAggression$large.residual,c("Aggression",
"Sibling_Aggression","Parenting_Style","Diet","Computer_Games", "Television",
"standardized.residuals")]
```

```
standardized.residuals
2                  2.289855
45                -3.081966
47                 2.484122
71                -2.483667
75                 2.152417
150                2.006389
```

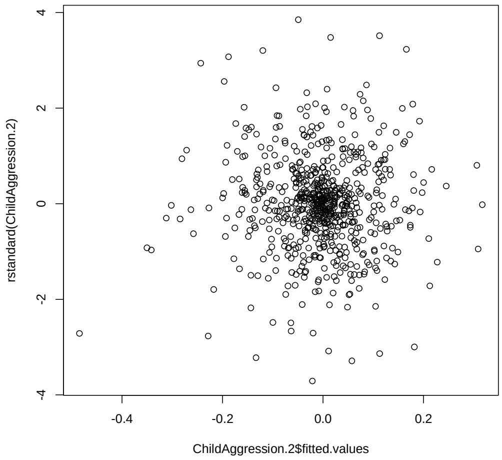
```
157              3.849871
163             -2.108737
169              3.206542
182              2.083747
199              2.558020
200              3.074039
204              2.090290
217             -2.712824
221              3.231368
266              2.028527
270             -2.996664
316              2.021033
351              2.396689
374              2.939848
375              2.322632
379             -2.766500
386              2.426592
407             -2.163684
411             -2.179070
421             -2.146689
431             -2.492695
439             -3.134520
440             -3.286946
463             -3.707637
482              3.477799
505             -3.220184
539              3.515591
589              2.020330
630             -2.114454
635             -2.662962
639             -2.706759
```

Histogram:

```
hist(rstandard(ChildAggression.2))
```



```
plot(ChildAggression.2$fitted.values,rstandard(ChildAggression.2))
```



Based on the final model (which is actually all we're interested in) the following variables predict aggression:

- ✓ Parenting style (*b* = 0.06, *β* = 0.18, *t* = 3.89, *p* < .001) significantly predicted aggression. The beta value indicates that as parenting increases (i.e. as bad practices increase), aggression increases also.
- ✓ Sibling aggression (*b* = 0.08, *β* = 0.08, *t* = 2.11, *p* < .05) significantly predicted aggression. The beta value indicates that as sibling aggression increases, aggression increases also.
- ✓ Computer games (*b* = 0.14, *β* = 0.15, *t* = 3.85, *p* < .001) significantly predicted aggression. The beta value indicates that as the time spent playing computer games increases, aggression increases also.
- ✓ E-numbers (*b* = –.11, *β* = –0.12, *t* = –2.86, *p* < .01) significantly predicted aggression. The beta value indicates that as the diet improved, aggression decreased.

The only factor not to predict aggression was:

- ✗ Television (*b* if entered = .03, *t* = 0.72, *p* > .05) did not significantly predict aggression.

Based on the standardized beta values, the most substantive predictor of aggression was actually parenting style, followed by computer games, diet and then sibling aggression.

R^2 is the squared correlation between the observed values of aggression and the values of aggression predicted by the model. The values in this output tell us that sibling aggression and parenting style in combination explain 5.3% of the variance in aggression. When computer game use and diet are factored in as well, 8.3% of the variance in aggression is explained (an additional 1.2%).

The Durbin–Watson statistic tests the assumption of 'independence of errors', which means that, for any two observations (cases) in the regression, their residuals should be uncorrelated (or independent). In this output the Durbin–Watson statistic falls within the recommended boundaries of 1–3, which suggests that errors are reasonably independent.

The scatterplot helps us to assess both *homoscedasticity* and *independence of errors*. The scatterplot does show a random pattern and so indicates no violation of the independence of errors assumption. Also, the errors on the scatterplot do not funnel out, indicating homoscedasticity of errors, thus no violations of these assumptions.