Statistical modelling: practical 2 solutions

Dr Colin Gillespie

1 *Simple linear regression*

1. Consider the data in table 1 for ice cream sales at Luigi Minchella's ice cream parlour.

   (a) Perform a linear regression of y on x. Should temperature be included in the model?

```
x = c(4, 4, 7, 8, 12, 15, 16, 17, 14, 11, 7, 5)
y = c(73, 57, 81, 94, 110, 124, 134, 139, 124, 103,
    81, 80)
m = lm(y ~ x)
summary(m)

##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.312  -2.656  -0.016   2.880   7.240
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   45.520      3.503    13.0  1.4e-07
## x              5.448      0.319    17.1  9.9e-09
##
## Residual standard error: 5.04 on 10 degrees of freedom
## Multiple R-squared:  0.967,Adjusted R-squared:  0.964
## F-statistic:  292 on 1 and 10 DF,  p-value: 9.88e-09

## The p-value for the gradient is 9.9e-09 This
## suggests temperature is useful
```

   (b) Calculate the sample correlation coefficient r. Perform a hypothesis test, where $H_0 : \rho = 0$ and $H_1 : \rho \neq 0$.[1] Compare this p-value to the p-value for testing if $\beta_1 = 0$.

[1] Where ρ is the population correlation coefficient.

```
## The p-value for the correlation is also 9.9e-09
cor.test(x, y)

##
##  Pearson's product-moment correlation
##
## data:  x and y
## t = 17.1, df = 10, p-value = 9.879e-09
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
##  0.9398 0.9955
## sample estimates:
##    cor
## 0.9833
```

(c) Construct a graph of the data. Add a dashed red line indicating the line of best fit.

```
plot(x, y, xlab = "Temp", ylab = "Sales")
abline(m, col = 2, lty = 2)
```

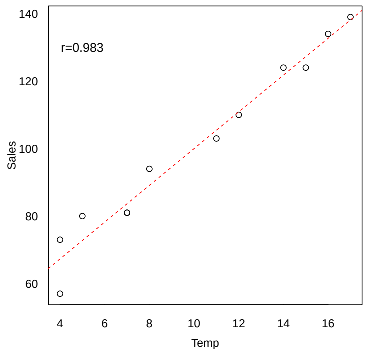

Figure 1: Scatterplot with the earnings data. Also shows the line of best fit (question 1c).

(d) Using the `text` function, add the text $r = 0.983$ to your plot.

```
text(5, 130, "r=0.983")
```

(e) Plot the standardised residuals against the fitted values. Does the graph look random?

```
## Model diagnostics look good
plot(fitted.values(m), rstandard(m))
```

(f) Construct a q-q plot of the standardised residuals.

```
qqnorm(rstandard(m))
## Model diagnostics look good
```

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
x, °C	4	4	7	8	12	15	16	17	14	11	7	5
y, £000's	73	57	81	94	110	124	134	139	124	103	81	80

Table 1: Monthly sales data from Luigi Minchella's ice cream parlour.

2. In a study of the effect of temperature x on yield y of a chemical process, the data in table 2 was obtained.

   (a) Perform a linear regression of y on x.

   (b) Calculate the sample correlation coefficient r.

   (c) Plot the data and add the line of best fit to your plot.

   (d) Plot the Studentized residuals against the fitted values.

   (e) Construct a q-q plot of the Studentized residuals.

x	25	26	27	28	29	30	31	32	33	34	35	36
y	10	16	13	17	20	18	19	23	25	22	29	26

Table 2: Twelve measurements from a study of temperature on yield.

2 Multiple linear regression

1. The data (below) are from 101 consecutive patients attending a combined thyroid-eye clinic. The patients have an endocrine disorder, Graves' Ophthalmopathy, which affects various aspects of their eyesight. The ophthalmologist measures various aspects of their eyesight and constructs an overall index of how the disease affects their eyesight. This is the Ophthalmic Index (OI) given in the dataset. The age of the patient and their sex are also recorded. In practice, and as this is a chronic condition which can be ameliorated but not cured, the OI would be monitored at successive clinic visits to check on the patient's progress. However, these data are obtained at presentation. We are interested in how OI changes with age and gender. The data can be obtained from

```
library("nclRmodelling")
data(graves)
```

   (a) Fit the multivariate regression model predicting OI from age and gender.

   (b) Examine the Studentized residual plots. Is there anything that would suggest you have a problem with your model? What do you do?

2. Dr Phil comes to see you with his data. He believes that IQ can be predicted by the number of years education. Dr Phil does not differentiate between primary, secondary and tertiary education. He has four variables:

   - `IQ` - the estimated IQ of the person (the response variable);
   - `AgeBegin` - the age of the person when they commenced education;
   - `AgeEnd` - the age of the person when they finished education;

- `TotalYears` - the total number of years a person spent in education.

The data can be obtained from:

```
data(drphil)
```

Read the data into R and fit the linear regression model:

$$IQ = \beta_0 + \beta_1 AgeBegin + \beta_2 AgeEnd + \beta_3 TotalYears + \epsilon$$

Explain what is wrong with this model? Suggest a possible remedy.

```
(m = lm(IQ ~ AgeBegin + AgeEnd + TotalYears, data = drphil))

##
## Call:
## lm(formula = IQ ~ AgeBegin + AgeEnd + TotalYears, data = drphil)
##
## Coefficients:
## (Intercept)     AgeBegin       AgeEnd
##    101.8662      -0.2012       0.0738
##  TotalYears
##          NA

# The problem is TotalYears = AgeEnd - AgeBegin
# Solution: remove TotalYears
```

3 *One way ANOVA tables*

1. A PILOT STUDY was developed to investigate whether music influenced exam scores. Three groups of students listened to 10 minutes of Mozart, silence or heavy metal before an IQ test. The results of the IQ test are as follows

Mozart	109	114	108	123	115	108	114
Silence	113	114	113	108	119	112	110
Heavy Metal	103	94	114	107	107	113	107

Table 3: Results from the study on how music affects examination performance.

(a) Construct a one-way ANOVA table. Are there differences between treatment groups?

```
x1 = c(109, 114, 108, 123, 115, 108, 114)
x2 = c(113, 114, 113, 108, 119, 112, 110)
x3 = c(103, 94, 114, 107, 107, 113, 107)
dd = data.frame(values = c(x1, x2, x3), type = rep(c("M",
    "S", "H"), each = 7))
m = aov(values ~ type, dd)
summary(m)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## type          2    193    96.6     3.4  0.056
## Residuals    18    511    28.4

## The p value is around 0.056. This suggests a
## difference may exist.
```

(b) Check the standardised residuals of your model.

```
plot(fitted.values(m), rstandard(m))
## Residual plot looks OK
```

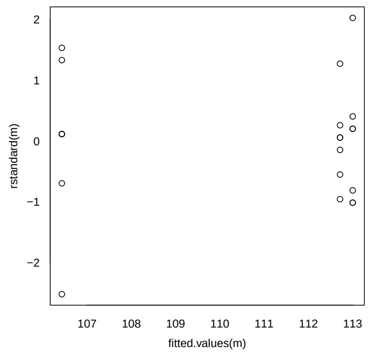

Figure 2: Model diagnostics for the music data.

(c) Perform a multiple comparison test to determine where the difference lies.

```
TukeyHSD(m)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = values ~ type, data = dd)
##
## $type
##        diff     lwr    upr  p adj
## M-H  6.5714 -0.6982 13.841 0.0804
## S-H  6.2857 -0.9839 13.555 0.0971
## S-M -0.2857 -7.5553  6.984 0.9945
```

THE FOLLOWING SECTIONS use the results of the Olympic heptathlon competition, Seoul, 1988. To enter the data into R, use the following commands

```
data(hep)
## Remove the athletes names and final scores.
hep_s = hep[, 2:8]
```

4 *Hierarchical clustering*

USING THE HEPTATHLON data set, carry out a clustering analysis. Try different distance methods and clustering functions.

```
plot(hclust(dist(hep_s)), labels = hep[, 1])
```

5 *Principal components analysis*

1. Calculate the correlation matrix of the hep data set.

```
## Round to 2dp
signif(cor(hep_s), 2)

##          hurdles highjump  shot run200m longjump
## hurdles   1.0000  -0.8100 -0.65    0.77   -0.910
## highjump -0.8100   1.0000  0.44   -0.49    0.780
## shot     -0.6500   0.4400  1.00   -0.68    0.740
## run200m   0.7700  -0.4900 -0.68    1.00   -0.820
## longjump -0.9100   0.7800  0.74   -0.82    1.000
## javelin  -0.0078   0.0022  0.27   -0.33    0.067
## run800m   0.7800  -0.5900 -0.42    0.62   -0.700
##          javelin run800m
## hurdles  -0.0078    0.78
## highjump  0.0022   -0.59
## shot      0.2700   -0.42
## run200m  -0.3300    0.62
## longjump  0.0670   -0.70
## javelin   1.0000    0.02
## run800m   0.0200    1.00
```

2. Carry out a PCA on this data set.[2]

   - Keep `score` in your PCA analysis. What happens and why? Do you think you should remove `score`?

```
## Remove: 1st column: athletes name Last column:
## It's a combination of the other columns
dd = hep[, 2:8]

## Run principle components
prcomp(dd)

## Standard deviations:
## [1] 8.36464 3.59098 1.38570 0.58571 0.32382
## [6] 0.14712 0.03325
##
## Rotation:
##                PC1        PC2      PC3      PC4
## hurdles   0.069509 -0.0094891  0.22181 -0.32738
## highjump -0.005570  0.0005647 -0.01451  0.02124
## shot     -0.077906  0.1359282 -0.88374 -0.42501
## run200m   0.072968 -0.1012004  0.31006 -0.81585
## longjump -0.040369  0.0148845 -0.18494  0.20420
## javelin   0.006686  0.9852955  0.16021 -0.03217
## run800m   0.990994  0.0127653 -0.11656  0.05828
##              PC5       PC6       PC7
## hurdles  -0.80703  0.424851 -0.083123
## highjump  0.14014  0.098374 -0.984881
## shot     -0.10442 -0.051745 -0.015650
## run200m   0.46179  0.082486  0.051313
```

[2] Remember to remove the athletes names.

```
## longjump  0.31899  0.894593  0.142110
## javelin   0.04880  0.006170  0.005033
## run800m   0.02785 -0.002987  0.001041
```

- Do you think you need to scale the data?

```
## Yes!. run800m dominates the loading since the
## scales differ
```

- Construct a biplot of the data.

```
prcomp(dd, scale = TRUE)

## Standard deviations:
## [1] 2.1119 1.0928 0.7218 0.6761 0.4952 0.2701
## [7] 0.2214
##
## Rotation:
##               PC1      PC2      PC3      PC4
## hurdles   0.45287 -0.15792 -0.04515  0.02654
## highjump -0.37720  0.24807  0.36778 -0.67999
## shot     -0.36307 -0.28941 -0.67619 -0.12432
## run200m   0.40790  0.26039  0.08359 -0.36107
## longjump -0.45623  0.05587 -0.13932 -0.11129
## javelin  -0.07541 -0.84169  0.47156 -0.12080
## run800m   0.37496 -0.22449 -0.39586 -0.60341
##               PC5      PC6      PC7
## hurdles  -0.09495 -0.78334 -0.38025
## highjump -0.01880 -0.09940 -0.43393
## shot     -0.51165  0.05086 -0.21762
## run200m  -0.64983  0.02496  0.45338
## longjump  0.18430 -0.59021  0.61206
## javelin  -0.13511  0.02724  0.17295
## run800m   0.50432  0.15556  0.09831

biplot(prcomp(dd, scale = TRUE))
```

6 *Survival analysis*

This final section, is meant to give you a taste at other statistical techniques available. R has many packages[3] available. To install a package, we use the command

[3] A package is just an "add-on" that provides new functionality.

```
install.packages("survival")
```

Once the package is installed, we load it using `library` function

```
library(survival)
```

The data set we will use is the `lung` data set[4] - this comes with the `survival` package. To load this data set, we use the command

[4] Survival in patients with advanced lung cancer from the North Central Cancer Treatment Group. Performance scores rate how well the patient can perform usual daily activities.

```r
data(lung)
```

We the use the `dim` function to extract the number of rows and columns

```r
dim(lung)

## [1] 228  10
```

This creates a data frame with the following columns

- `inst`: Institution code
- `time`: Survival time in days
- `status`: censoring status 1=censored, 2=dead
- `age`: Age in years
- `sex`: Male=1 Female=2
- `ph.ecog`: ECOG performance score (0=good 5=dead)
- `ph.karno`: Karnofsky performance score (bad=0-good=100) rated by physician
- `pat.karno`: Karnofsky performance score as rated by patient
- `meal.cal`: Calories consumed at meals
- `wt.loss`: Weight loss in last six months

To begin, we create a `Surv` object:[5]

[5] Look at `?Surv` for further details.

```r
Surv(lung$time, lung$status)
```

We can fit a Kaplein-Meier curve using the `surfit` function:

```r
survfit(Surv(lung$time, lung$status) ~ 1)
```

and also plot survival curve

```r
plot(survfit(Surv(lung$time, lung$status) ~ 1))
```

To fit the model using an additional covariate, we just alter the formula

```r
plot(survfit(Surv(lung$time, lung$status) ~ lung$sex))
```

Task: Load the `heart` data set:[6]

[6] Look at the help page: `?heart`

```r
data(heart)
```

and make a `Surv` object

```r
Surv(heart$start, heart$stop, heart$event)
```

Why are there three arguments in the above function? Construct a Kaplein-Meier plot. Look at the `coxph` function. Try fitting a cox proportional hazard function.

Solutions

Solutions are contained within this package:

```r
library("nclRmodelling")
vignette("solutions2", package = "nclRmodelling")
```