Class 5: Regression I.

Practice Exercises[*]

Diego Perez Ruiz <diego.perezruiz@manchester.ac.uk>

Teacher Assistants: Wei Zhuang <wei.zhuang-2@postgrad.manchester.ac.uk> and Shing Yan Kwong<shingyan.kwong@postgrad.manchester.ac.uk>

1 Linear regression: Baby Weight

Here we use data to check if we can determine which variables impact birth weight. Download the data called 'babies.data' from BlackBoard. Load the data in R (make sure you put in the correct directory for where you saved the file):

```
babies.data <- read.table("[insert data file location]", header = TRUE)
attach(babies.data)
```

The `attach` function makes it easier to call all variables available in our data set by their name. To see how many observations and variables are available in the data set, you can use the following command:

```
> dim(babies.data)
[1] 1174    7
```

You can also check the beginning or end of the data using commands

```
head(babies.data)
tail(babies.data)
```

We fit two very simple models: one with only the intercept term (commonly referred to as a 'null model') and one with variable `mom.weight` as predictor. We thus want to see if the weight of the mother can be used as a predictor for babies weight.

```
fit0 <- lm(birth.weight ~ 1)
fit1 <- lm(birth.weight ~ mom.weight)
```

Look at the results from the first (non-null) model. What do you note?

```
> summary(fit1)
Call:
lm(formula = birth.weight ~ mom.weight)

Residuals:
Min      1Q  Median      3Q     Max
-66.051 -10.916   0.328  11.026  56.084
```

[*] Please do these exercises before attending the online labs.

```
Coefficients:
Estimate   Std. Error   t value   Pr(>|t|)
(Intercept) 101.75393    3.31927     30.655    < 2e-16 ***
mom.weight    0.13783    0.02551     5.404     7.89e-08 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 18.11 on 1172 degrees of freedom
Multiple R-squared:  0.02431,Adjusted R-squared:  0.02348
F-statistic:  29.2 on 1 and 1172 DF,  p-value: 7.887e-08
```

You can also visualise the fitted model by using (see lecture notes for this example):

```
plot(mom.weight,birth.weight,  type = ’p’, col = ’black’, pch=16,
main = "birth weight predicted by mom’s weight",
xlab = "mom’s weight", ylab = "birth weight")
abline(fit1, col = ’red’)
```

Although the coefficient estimate seems to be highly significant, the r^2 indicates a very week linear relationship between the two variables. Recall that for simple linear regression, the r^2 value is simply the square of the correlation coefficients. Let’s look at how all variables correlate with ’birth.weight’:

```
> cor(babies.data, birth.weight)
[,1]
birth.weight    1.00000000
gestation       0.40754279
not.first.born -0.04390817
mom.age         0.02698291
mom.height      0.20370418
mom.weight      0.15592327
mom.smokes     -0.24679951
```

As noted here, the correlation between `birth.weight` and `mom.weight` is only 0.156 (and therefore we got $r^2 = 0.156^2 \approx 0.0256$ as shown in model output.) From the correlations we note that variable `gestation` seems to have the strongest linear relationship with `birth.weight`. Fit a second model by yourself now repeating the above codes but with predictor `gestation` instead. Do we have a better fit?

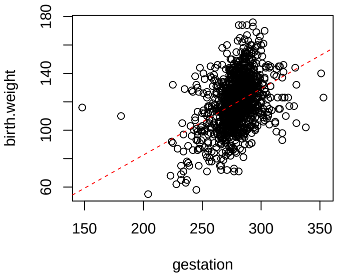
```
fit2 <- lm(birth.weight ~ gestation)
```

The answer is no. For this specific data set, we cannot find another single predictor that correlates stronger with `birth.weight`. You can see the fitted regression line using following commands:

```
plot(gestation, birth.weight)
abline(fit2, lty = 2, col = "red")
```

Can you detect outliers in this plot? The following command lines calculates Cook’s distance which is a way to find outcomes that may distort results because they are outliers.

```
case.numbers <- 1:length(gestation)
influence <- cooks.distance(fit2)
plot(case.numbers, influence)
```



A very useful way to identify observation numbers in scatter plots is the interactive tool called `identify`:

`identify(gestation, birth.weight)`

Once you run the above code line, you can click on points in the scatter plot to identify them. Once you have clicked the ones you think are outliers (the two points to the left), press `esc` to end. You should get the following output:

```
> identify(gestation, birth.weight)
[1] 239 820
```

We can now trim the data by removing these two points (observation 239 and 820) and attach the trimmed data set:

```
trim.data <- babies.data[c(-239, -820), ]
detach(babies.data)
attach(trim.data)
```

As seen the data now has two observations less:

```
> dim(trim.data)
[1] 1172    7
```

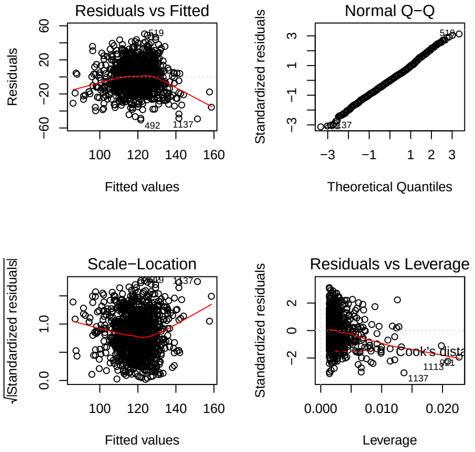
Repeat the above codes to run the regression `fit2` model using the trimmed data set and see if the results got better.

Spend some time now fitting multiple regression models. Choose two predictors from the variable set, e.g. a model with `mom.smokes` and `gestation` as predictors is run by the following codes (you can continue working with the trimmed data set):

```
fit3 <- lm(birth.weight ~ gestation + mom.smokes)
```

We can run some diagnostics and check if model assumptions are satisfied:

```
par(mfrow = c(2,2))
plot(fit3)
dev.off()
```



Interpret the different plots (refer to lecture notes if unsure). The first plot (residuals vs. fitted values) is a simple scatterplot between residuals and predicted values. It should look more or less random (*why?*). The second plot (normal Q-Q) is a normal probability plot. It will give a straight line if the errors are distributed normally. The third plot (Scale-Location), like the the first, should not show any pattern. The last plot (Cook's distance) tells us which points have the greatest influence on the regression. These are also indicators of outliers but we have already removed them here.

2 Logistic regression: Spam Filter

We will in this example use emails from one gmail account over three months in 2012 to see if we can build a model that successfully can predict if an email is spam based on different characteristics of the email. Start by loading the data set which is already in R format and can be downloaded from BlackBoard. Then check the number of observations and variables available and what the first observations look like.

```
load("email.Rdata")
attach(email)
dim(email)
head(email)
```

Here are the 19 variables available to us:

1. **spam**: Indicator for whether the email was spam
2. **to multiple**: Indicator for whether the email was addressed to more than one recipient.
3. **from**: Whether the message was listed as from anyone (this is usually set by default for regular outgoing email).
4. **cc**: Indicator for whether anyone was CCed.
5. **sent email**: Indicator for whether the sender had been sent an email in the last 30 days.
6. **image**: Indicates whether any images were attached.
7. **attach**: Indicates whether any files were attached.
8. **dollar**: Indicates whether a dollar sign or the word 'dollar' appeared in the email.
9. **winner**: Indicates whether "winner" appeared in the email.
10. **inherit**: Indicates whether "inherit" (or an extension, such as inheritance) appeared in the email.
11. **password**: Indicates whether "password" appeared in the email.

12. **num char**: The number of characters in the email, in thousands.
13. **line breaks**: The number of line breaks in the email (does not count text wrapping).
14. **format**: Indicates whether the email was written using HTML (e.g. may have included bolding or active links) or plaintext.
15. **re subj**: Indicates whether the subject started with "Re:", "RE:", "re:", or "rE":
16. **exclaim subj**: Indicates whether there was an exclamation point in the subject.
17. **urgent subj**: Indicates whether the word "urgent" was in the email subject.
18. **exclaim mess**: The number of exclamation points in the email message.
19. **number**: Factor variable saying whether there was no number, a small number (under 1 million), or a big number.

Steps and questions:

1. Fit a logistic regression model (model1) between spam and ‘to multiple’ using the glm function (‘to multiple’ as the independent variable).
2. If an email is randomly selected and it has just one address in the ‘To’ field, what are the odds and probability that it is spam?
3. What if more than one address is listed in the ‘To’ field?
4. Use all variables available to predict if an email is a spam or not (model2).
5. Which variables appear to be meaningful for identifying spam? Which predictor appears to have the largest effect?
6. Run some diagnostics for your email classifier (hint: plot predicted probability vs spam/non-spam.)
7. Test different models using different set of predictors from available variables in the data set. Use the significance of coefficients and the AIC to guide in model selection and assessment. (hint: for comparison, use ‘AIC’ function in R.).