Minilab 5b Worksheet

Linear Regression: Fitting a Linear Model

Simple linear regression is the method of finding the "line of best fit" to data involving one quantitative *predictor* variable X and one quantitative *response* variable Y. We have seen (in the previous minilab) that the best line is the one that minimises the *sum of squared residuals*. We can apply formulas to calculate the coefficients of the line from the data values.

We think of this process as fitting a *linear model* (in this case a line) to the *data*. In this minilab, we use R to carry out the model fitting process using simple function calls that take care of all the calculations.

1. Fitting a Linear Model

The R function `lm()` is used to fit a *linear model*.

(1) Install the following R package (run this command only once).

```
install.packages("datarium")
```

(2) We will look at some marketing data that shows the impact of three different advertising media (youtube, facebook, and newspaper) on sales. The data are advertising budget in $US1000 along with the sales.

```
library(tidyverse)
library(datarium)
marketing = as_tibble(marketing)
marketing
```

We can see the resulting tibble has four columns (youtube, facebook, newspaper and sales).

(3) Suppose we want to investigate how “sales” depends on spending on “youtube” advertising alone, i.e., can we predict sales knowing only the spending on youtube advertising.
The response variable Y is *sales* and the predictor variable X is *youtube*.
Exploratory data analysis should start with a scatterplot.

```
ggplot(data=marketing,aes(x=youtube,y=sales)) +
  geom_point()
```

It looks plausible that there is a positive relationship between youtube and sales.

(4) To fit a *linear model* we use the `lm()` function in R.
The *form* of the linear model is provided as a formula e.g., here we use “`y~x`” to mean that the variable y is a linear function of the variable x alone, i.e., the model we wish to fit is the line $y = a + bx$.

```
x = marketing$youtube
y = marketing$sales
# Model is y = a + b*x
model = lm(y~x)
summary(model)
a = model$coefficients[1]
b = model$coefficients[2]
a
b
```

This gives $a = 8.439112$ and $b = 0.04753664$, i.e., the fitted linear model is

$$y = (8.439112) + (0.04753664) \times x$$

Notice that we can also see the fitted coefficients (values of a and b) in the summary output.

Exercise. Calculate the correlation coefficient between sales and youtube.

(5) We usually bypass the explicit use of x and y and use the data columns directly from the data tibble.
We use the linear model “sales~youtube” to mean that the variable *sales* is a linear function of the variable *youtube* alone, i.e., the model we wish to fit is the line

$$sales = a + b \times youtube$$

We also pick out the values of the fitted coefficients directly from the summary output.
It is also easy to add the line of best fit to the scatterplot using ggplot.

```
# Model is sales = a + b*youtube
ggplot(marketing,aes(x=youtube,y=sales)) +
  geom_point() +
  geom_smooth(method=lm, se=FALSE)
model = lm(sales~youtube, data=marketing)
summary(model)
```

The summary gives information about the two coefficients (a and b).

- The *intercept* is a so the summary is saying that $a = 8.439112$ is the best estimate.

- The coefficient of $youtube$ is b so the summary is saying that $b = 0.047537$ is the best estimate.

So the fitted linear model is:

$$sales = (8.439112) + (0.047537) \times youtube$$

(6) The summary function gives a summary of the fitted model. The R tidyverse package “broom” can be used to nicely tidy up (sweep up) the output of the `lm()` function as a tibble that can be further used within the tidyverse set of functions and packages.

```
library(broom)
tidy(model)
```

(7) We can then calculate the *predicted* values (called “y hat”) $\widehat{y_i} = a + bx_i$ and the *residuals* $e_i = y_i - \widehat{y_i}$ using R. Residual is always *actual* minus *predicted*.
The R tidyverse package “broom” is able to tidy house and to do all of this for us.

```
fitted = augment(model)
fitted
```

Notice that the fitted tibble contains the original data using in the model (sales and youtube) together with a bunch of other values for each row/observation (including “.fitted” and “.resid”).

(8) We can use the fitted values to demonstrate the link between the fitted values and the original values on our scatterplot if we wish.

```
ggplot(fitted,aes(x=youtube,y=sales)) +
  geom_point() +
  geom_smooth(method=lm, se=FALSE) +
  geom_segment(aes(xend=youtube,yend=.fitted), colour="red", size=0.3)
```

Summary. To summarise, suppose we wish to predict "sales" from "youtube" advertising. In R, the steps involved are simply:

```
library(tidyverse)
library(datarium)
marketing = as_tibble(marketing)
# Model is sales = a + b*youtube
ggplot(marketing,aes(x=youtube,y=sales)) +
  geom_point() +
  geom_smooth(method=lm, se=FALSE)
model = lm(sales~youtube, data=marketing)
summary(model)
```

Conclusion: the fitted model is:

$$sales = (8.439112) + (0.047537) \times youtube$$

Note that ggplot takes care of plotting the scatterplot and plotting the line of best fit, lm() takes care of fitting the linear model (hence the "lm") and summary() prints information about the fitted model (including the fitted model coefficients).

Exercise. Now consider the two linear models "sales~facebook" (with *facebook* advertising as the only predictor variable) and "sales~newspaper" (with *newspaper* advertising as the only predictor variable). Investigate these linear models by scatterplot and fitting the linear model. Write down the equation of the fitted model in each case.

Exercise. A dataset on using wetland systems to treat wastewater uses biochemical oxygen demand (BOD) mass *loading* to predict BOD mass *removal*.

```
library(tidyverse)
loading = c(3,8,10,11,13,16,27,30,35,37,38,44,103,142)
removal = c(4,7, 8, 8,10,11,16,26,21, 9,31,30, 75, 90)
wetland = tibble(loading,removal)
```

(a) Write R code to construct a scatterplot of the dataset above (using ggplot), including the line of best fit.

(b) Use lm() to fit the linear model "removal~loading" to the dataset. What is the equation of the line of best fit?

(c) Use lm() to fit the linear model "removal~loading-1" (no intercept term) to the dataset. What is the equation of the line of best fit?

2. A first taste of Multiple Regression

Suppose we wish to use <u>two</u> predictor variables (*youtube* and *facebook*) to predict *sales*.

We wish to fit the 3D "plane of best fit" of the form

$$z = a + bx + cy$$

where z is sales, x is *youtube* and y is *facebook*.

Again the criteria for the <u>best plane</u> is to minimise the sum of the squared residuals.

The linear model is "sales~youtube+facebook" and we can use `lm()` to fit this model to the data.

```
# Model is sales = a + b*youtube + c*facebook
model = lm(sales~youtube+facebook,data=marketing)
summary(model)
```

The summary output shows that the fitted model is:

$$sales = 3.50532 + 0.04575 \times youtube + 0.18799 \times facebook$$

Exercise.

(a) Consider the linear model “sales~youtube+facebook+newspaper” (with three predictor variables). Use lm() to fit this linear model to the data and write down the fitted model.

(b) How is the scatterplot (produced by the R code below) related to the fitted model from part (a)?

```
ggplot(marketing,aes(x=youtube+facebook+newspaper,y=sales)) +
  geom_point() +
  geom_smooth(method=lm, se=FALSE)
```

Summary

In this minilab, we have used R to fit a *simple linear model* (one quantitative predictor variable and one quantitative response variable) to a dataset in R using the `lm()` function. We have also had a very brief look at *multiple regression* (two or more predictor variables and one response variable). The *form* of the model is specified by an expression like “sales~youtube” or “sales~youtube+facebook”. In each case, the fitted model coefficients can be read directly off the output from the summary function and we can then write down the equation of the fitted model. *There is a lot more output here that we will learn about in the coming minilabs.*

The function `lm()` is the main way that R handles linear models. The R tidyverse package *broom* includes some useful functions for bringing the results of `lm()` into a tibble and for calculating the *fitted* values and *residuals*. We will see more of fitted and residual values when we look at diagnostic plots in a future minilab.