Lab03 - Linear Regression basics

1) Linear regression: simple exercise with only 8 points.

The exercise has ben solved step by step, using R only to help with calculations. First of all, let us estimate the parameters beta0 (b0) and beta1 (b1), i.e. the intercept and slope of the linear model.

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

```
# set up the predictor variables xi and the responses yi
# note that x and y have been generated as follows:
# x = rnorm(8)
# y = 2 * x + rnorm(8,5,.5)
# then they have been rounded to ease the calculations
x = c(0.75,-0.64,1.43,-0.61,0.23,0.43,-1.48,2.06)
y = c(6.60,4.31,7.51,3.48,5.21,5.74,1.65,9.76)
n = length(x)
```

First, calculate b1 (slope) and b0

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

```
mean(x)
# [1] 0.27125
mean(y)
# [1] 5.5325

x - 0.27
# [1]  0.48 -0.91  1.16 -0.88 -0.04  0.16 -1.75  1.79
y - 5.53
# [1]  1.07 -1.22  1.98 -2.05 -0.32  0.21 -3.88  4.23

(x-0.27) * (y - 5.53)
# [1] 0.51 1.11 2.30 1.80 0.01 0.03 6.79 7.57
## sum((x-0.27) * (y - 5.53)) = 20.13

(x-0.27)^2
# [1] 0.23 0.83 1.35 0.77 0.00 0.03 3.06 3.20
## sum((x-0.27)^2) = 9.47

## b1 = slope coefficient = sum((x-0.27) * (y - 5.53))/sum((x-0.27)^2)
b1 = 20.13/9.47
# b1 = 2.12

## b0 = intercept
```

```
# b0 = mean(y) - b1 * mean(x)
b0 = 5.53 - 2.12 * 0.27
# b0 = 4.96

# given the parameters we calculated, the estimated yhat = 4.96 + 2.12 * x
# plot the points
plot(x,y)
# draw the estimated function
abline(b0,b1);
# draw the original function
abline(5,2,col="red")
```

---

2) Calculate the residuals

```
yhat = b0 + b1 * x
RSS=sum((y-yhat)^2)
# RSS = 1.059709
```

Note that this value of RSS is a minimum: changing values of b0 and b1 RSS will always be bigger

---

3) Calculate the standard error:

$$\mathrm{SE}(\hat\beta_0)^2 = \sigma^2\left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i-\bar{x})^2}\right], \quad \mathrm{SE}(\hat\beta_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i-\bar{x})^2}$$

We already have mean(x) = 0.27 and the SSE = 9.47. However, in practice we don't know sigma (we are usually not given the original distribution), so we need to estimate that. RSE (the Residual Standard Error) is a good estimate for it:

$$\mathrm{RSE} = \sqrt{\mathrm{RSS}/(n-2)}$$

```
RSE = sqrt(RSS/(n-2))
# RSE = 0.42

SEb0 = sqrt(.42^2 * (1/8 + (0.27^2 / 9.47)))
# SEb0 = 0.1529965
SEb1 = sqrt(.42^2 / 9.47)
# SEb1 = 0.1364817
```

---

4) Compute 95% confidence intervals

sample of data. For linear regression, the 95 % confidence interval for β_1 approximately takes the form

$$\hat\beta_1 \pm 2 \cdot \mathrm{SE}(\hat\beta_1). \tag{3.9}$$

```
c(b1-2*SEb1, b1+2*SEb1)
# [1] 1.852697 2.398623

c(b0-2*SEb0, b0+2*SEb0)
# [1] 4.651607 5.263593
```

NOTE that 2 is just an approximation (see note 3 at page 66). The next step will show how to calculate the proper interval to have 95% confidence.

---

5) Compute the t-statistic

t = (b1-0) / (SEb1) = 15.56769

For simple linear regression we use a t-distribution with n – 2 degrees of freedom: the sample size minus the number of estimated parameters.

Look up the table with the pre-computed probabilities for different degrees of freedom and values of t:

PDF File

(original source: http://bcs.whfreeman.com/ips6e/content/cat_050/ips6e_table-d.pdf)

---

6) Recall RSE and compute R^2

RSE is an estimate of the *lack of fit:*

$$\mathrm{RSE} = \sqrt{\mathrm{RSS}/(n-2)}$$

```
% TSS = total sum of squares (similar to RSS but wrt the mean and not the yi)
TSS = sum((y-mean(y))^2)
% [1] 43.84995

% compute R^2
Rs = (TSS-RSS)/TSS
% [1] 0.9758408

% show relationship between R^2 and correlation in the univariate case
cor(x,y)^2
```

---

7) Show semi-automatic solution

The experiment above can be conducted in a faster way, just by making R do more calculations (instead of moving actual numbers from one formula to another - that was just to give a step-by-step introduction to linear regression). Here is the code:

```
# initialize variables
x = c(0.75,-0.64,1.43,-0.61,0.23,0.43,-1.48,2.06)
y = c(6.60,4.31,7.51,3.48,5.21,5.74,1.65,9.76)
n = length(x)
```

```
# find parameters
b1 = sum((x-mean(x)) * (y-mean(y))) / sum((x-mean(x))^2)
b0 = mean(y) - b1 * mean(x)

# calculate RSS and RSE
yhat = b0 + b1 * x
RSS=sum((y-yhat)^2)
RSE = sqrt(RSS/(n-2))

# calculate SEb0 and SEb1
SEb0 = sqrt(RSE^2 * (1/length(x) + mean(x)^2/sum((x-mean(x))^2)))
SEb1 = sqrt(RSE^2 /sum((x-mean(x))^2))

# compute t-statistics
t0 = (b0-0) / (SEb0)
t1 = (b1-0) / (SEb1)

# compute R^2
TSS = sum((y-mean(y))^2)
Rs = (TSS-RSS)/TSS
```

---

8) Redo everything automagically with R

```
help(lm)

lm.fit = lm(y~x)
plot(x,y); abline(lm.fit); abline(5,2,col="red")

# show that the values we find are consistent with the ones we calculated previously
summary(lm.fit)
coef(lm.fit)
confint(lm.fit)

# show that predictions can also be done
predict(lm.fit,data.frame(x = 4), interval="confidence")
```

---

9) Finally, show how estimates change with (1) number of points and (2) variance

```
# more points
x = rnorm(100)
y = 2 * x + rnorm(100, 5, .5)

lm.fit = lm(y~x)
plot(x,y); abline(lm.fit); abline(5,2,col="red")
summary(lm.fit)

# same points as in simple experiment, much more variance
x = rnorm(8)
y = 2 * x + rnorm(8, 5, 5)
```