Lab04 - Multiple linear regression

0) Clarification on the "degrees of freedom" concept in the calculation of RSE

Errors vs Residuals
They are both deviations of the observed value of an element of a statistical sample from some other value:

- **error**: deviation from the *true* value of a quantity of interest (e.g. *population mean*)
- **residual**: deviation from the *estimated* value of a quantity of interest (e.g. *sample mean*)

Example with the sample mean:

Suppose we have 10 numbers drawn randomly from a normal distribution with mean 5 and variance 0.1:

```
set.seed(12345)
# 12345? That's amazing, I got the same combination on my luggage!
#
Prepare....Spaceball 1 for for immediate departure.....and change the combination on my luggag
e!

n = 100
mu = 5
# let us generate n points with mean mu and standard deviation .1
x = rnorm(n,mu,.1)

mean(x)
# [1] 5.02452
var(x)
# [1] 0.01242625
sd(x)
#[1] 0.1114731

#-------------------------------------------------------------------------------------
# note that the sum of the residuals (x-xm) is zero
xm = mean(x)
sum(x-xm)
# [1] -3.552714e-15

# this means that the residuals are not all independent
sum(mean(x)-x[1:n-1]) # this is the sum of all the residuals minus the last one
mean(x)-x[n] # ... and this is the last one: if you sum it to the others, you'll get zero

# typical formula for variance (compare it with var(x)
sum((x-xm)^2) / n
# [1] 0.01230199

sum((x-xm)^2) / (n-1)
# [1] 0.01242625 # this is exactly var(x)!

#-------------------------------------------------------------------------------------
```

```
# Let us try this in the regression context.
x = rnorm(n)
y = 2 * x + rnorm(n,0,.2)

plot(x,y)
cor(x,y)

fit = lm(y~x)

summary(fit)
coef(fit)

yhat = coef(fit)[1] + coef(fit)[2] * x
yreal = 2 * x
```

Note that the MSE (Mean Squared **Error**) is actually a mean of squared **Residuals**!!!

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

```
# let us calculate the residuals and the errors:
res = y - yhat
err = y - yreal

sum(res^2) / n    # this is the MSE
sum(res^2) / (n-2)    # this is a "corrected" version of the MSE
var(err)

# note how the "corrected" MSE better estimates the variance of the errors!
# Now let us come back to the RSE and RSS calculation we did last time...
RSS = sum(res^2)
MSE = RSS/n
RSEsq = RSS/(n-2)
RSE = sqrt(RSS/(n-2))
sd(err)
```

Useful links:
https://en.wikipedia.org/wiki/Degrees_of_freedom_(statistics)
https://en.wikipedia.org/wiki/Errors_and_residuals
http://mathworld.wolfram.com/Variance.html

1) Study the Advertising dataset and try to answer the questions starting at page 73 of the book.

```
# Load Advertising dataset (from http://www-bcf.usc.edu/~gareth/ISL/Advertising.csv) and show its main features
Advertising = read.csv("~/Downloads/Advertising.csv")
dim(Advertising)
n = dim(Advertising)[1] # the number of points, will be useful later
attach(Advertising)
summary(Advertising)
```

```
pairs(Advertising)

Advertising = Advertising[,2:5]
pairs(Advertising)
```

2) Run simple regression on sales/TV, sales/Radio, and sales/Newspapers, and take advantage of this to do a recap

```
fit = lm(Sales~TV)
plot(TV,Sales)
abline(fit,col='green')
summary(fit)

fit = lm(Sales~Radio)
plot(Radio,Sales)
abline(fit,col='green')
summary(fit)

fit = lm(Sales~Newspaper)
plot(Newspaper,Sales)
abline(fit,col='green')
summary(fit)

# let us recap on some concepts:
```

```
> summary(fit)

Call:
lm(formula = Sales ~ Newspaper)

Residuals:
     Min       1Q   Median       3Q      Max
-11.2272  -3.3873  -0.8392   3.5059  12.7751

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 12.35141    0.62142   19.88  < 2e-16 ***
Newspaper    0.05469    0.01658    3.30  0.00115 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.092 on 198 degrees of freedom
Multiple R-squared:  0.05212,   Adjusted R-squared:  0.04733
F-statistic: 10.89 on 1 and 198 DF,  p-value: 0.001148
```

```
# RSS
SalesHat = coef(fit)[1] + coef(fit)[2]*Newspaper
res = Sales - SalesHat
RSS=sum(res^2)
# if you do "min(res)" you get the Min Residual shown above (-11.23)

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

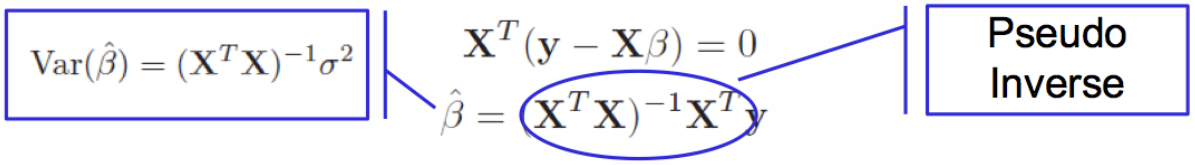
```
# calculate SEb0 and SEb1
SEb0 = sqrt(RSE^2 * (1/n + mean(Newspaper)^2/sum((Newspaper-mean(Newspaper))^2)))
SEb1 = sqrt(RSE^2 /sum((Newspaper-mean(Newspaper))^2))

# show confidence intervals and compare them with
b0 = coef(fit)[1]
b1 = coef(fit)[2]
confint(fit)
c(b0-2*SEb0, b0+2*SEb0)
c(b1-2*SEb1, b1+2*SEb1)

# remember that 2 is an approximation! Go and check the best value in the t-statistics table
# (try e.g. 1.984, 1.98, 1.97, etc)

# compute t-statistics and look for them on the t-distribution table
t0 = (b0-0) / (SEb0)
t1 = (b1-0) / (SEb1)
```

3) Comment the multiple linear regression approach, and show how parameters are calculated in this case (some matrix algebra here)

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p.$$

$$\begin{aligned} \text{RSS} &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - \cdots - \hat{\beta}_p x_{ip})^2. \end{aligned}$$

Beta is found by solving the following equation:



where

- X is an N x (p+1) data matrix
- y is an N x 1 vector of the desired output
- beta is a (p+1) x 1 vector of the model coefficients

In R:

```
# slice Advertising to contain only the first three columns
X = as.matrix(Advertising[,1:3])

# add a column of ones to take into account the intercept
X = cbind(1,X)

# implement the least squares solution
beta = solve( t(X) %*% X) %*% t(X) %*% Sales

# or:
install.packages('pracma')
# install and then load the package "pracma"
library('pracma')

beta = pinv(X) %*% Sales

# automatically, with R:
fit = lm(Sales ~ TV + Radio + Newspaper)
summary(fit)

# comment results of multiple linear regression with correlation (see e.g. cor(Radio,Newspaper))
cor(Advertising)
```

4) Compute the F-statistic (answer to: "is there a relationship between the response and the predictors?")

$$F = \frac{(\mathrm{TSS} - \mathrm{RSS})/p}{\mathrm{RSS}/(n-p-1)},$$

```
# TSS = total sum of squares (similar to RSS but wrt the mean and not the yi)
TSS = sum((Sales-mean(Sales))^2)

p = length(beta)-1
SalesHat = beta[1] + beta[2]*TV + beta[3]*Radio + beta[4]*Newspaper
RSS = sum((Sales-SalesHat)^2)

F = ((TSS-RSS)/p) / (RSS/(n-p-1))
```

5) Subset selection

```
detach(Advertising)
rm(list = ls())
Credit = read.csv("~/Downloads/Credit.csv")
attach(Credit)
pairs(Credit)

# show a manually calculated selection (first step)
fit = lm(Balance ~ Income)
```

```
BalHat = coef(fit)[1] + coef(fit)[2] * Income
sum((Balance-BalHat)^2)

fit = lm(Balance ~ Limit)
BalHat = coef(fit)[1] + coef(fit)[2] * Limit
sum((Balance-BalHat)^2)

fit = lm(Balance ~ Rating)
BalHat = coef(fit)[1] + coef(fit)[2] * Rating
sum((Balance-BalHat)^2)

fit = lm(Balance ~ Cards)
BalHat = coef(fit)[1] + coef(fit)[2] * Cards
sum((Balance-BalHat)^2)

fit = lm(Balance ~ Age)
BalHat = coef(fit)[1] + coef(fit)[2] * Age
sum((Balance-BalHat)^2)

fit = lm(Balance ~ Education)
BalHat = coef(fit)[1] + coef(fit)[2] * Education
sum((Balance-BalHat)^2)
...

# do it automatically

install.packages('leaps')
library(leaps)
fit = regsubsets(Balance~., Credit)
summary(fit)
```

---

6) If time allows it… the icecreams and sharks example!!!

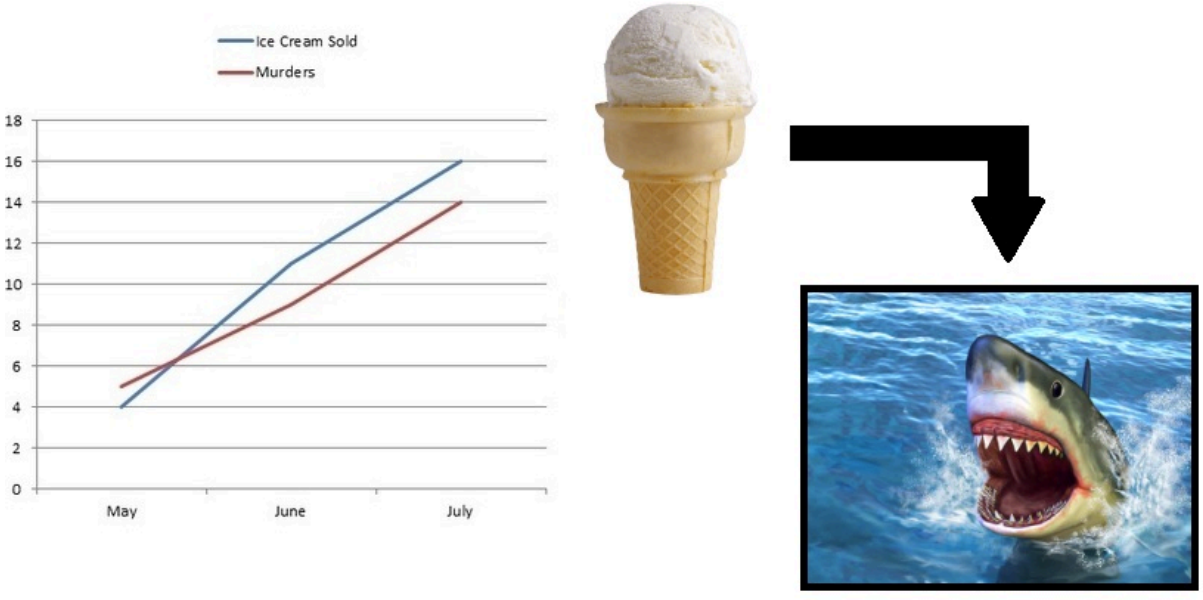


Image courtesy of http://www.vaccinenation.org/

(full url: http://www.vaccinenation.org/2014/04/10/ice-cream-causes-shark-attacks-vaccines-cause-autism/)

```
# let us create our dataset: first people...
people = rnorm(100,50,20)
# then icecreams (not everyone buys them: roughly 2 every 3 people)
icecreams = .6 * people + rnorm(100, 0, 5)
# then sharks (one attack every 100 people - I know that's rather high ;-))
sharks = .01 * people + rnorm(100, 0, .1)

# let us fit icecreams sales wrt people
lm.fit = lm(icecreams~people)
plot(people,icecreams)
abline(lm.fit,col="gray");

# read the summary and comment the null hypothesis over the intercept
summary(lm.fit)

# now show how the plot would look like if we accept the fact that intercept is 0
abline(0,coef(lm.fit)[2],col="green");

# … and now show it w.r.t. the ground truth
abline(0,.6,col="red");

# repeat for sharks
lm.fit = lm(sharks~people)
plot(people,sharks)
abline(lm.fit,col="gray");
summary(lm.fit)

# same as above - accept the null hypothesis
abline(0,coef(lm.fit)[2],col="blue");

# now see what happens about sharks and ice creams...
lm.fit = lm(sharks~icecreams)
plot(icecreams,sharks)
abline(lm.fit,col="red");
summary(lm.fit)

### WHAT??? IS THERE CORRELATION BETWEEN THEM?
cor(sharks, icecreams)

# that's pretty high… now let us try multiple linear regression
lm.fit = lm(sharks~people+icecreams)
```