Multiple Linear Regression &

General Linear Model in *R*

Multiple linear regression is used to model the relationship between one numeric outcome or response or dependent variable (Y), and several (multiple) explanatory or independent or predictor or regressor variables (X). When some predictors are categorical variables, we call the subsequent regression model as the General Linear Model.

1. Import Data in .csv format

#Download data in www.math.uah.edu/stat/data/Galton.csv

(1) You can import data directly from the internet:

```
Data <-read.csv("http://www.math.uah.edu/stat/data/Galton.csv", header = T)
```

(2) You can save the data in your R working directory, and then import the data to R;

```
#getwd() returns an absolute filepath representing
#the current working directory of the R process;
#setwd(dir) is used to set the working directory to dir.
getwd()
```

```
## [1] "C:/Users/***/Documents"
```

```
#Put your Galton.csv in the directory above
#Then you can read the file directly!

data<-read.csv("Galton.csv")
#read.csv(file, header = TRUE, sep = ",", quote = "\"",
#         dec = ".", fill = TRUE, comment.char = "", ...)
#More information about reading data, use ?read.csv()
```

(3) Alternatively, you can save the data to any of your own directories, and then import to R

```
# for example, D:/ams394/galton/
```

```
# See the following website for more options to import data to R
```

```
# http://www.r-tutor.com/r-introduction/data-frame/data-import

#Then you can read the file into R directly!

data = read.csv("d:/ams394/galton/Galton.csv")

#Recall:
#Y = height of child
#x1 = height of father
#x2 = height of mother
#x3 = gender of children

y<-data$Height
x1<-data$Father
x2<-data$Mother
x3<-as.numeric(data$Gender)-1
#You can see as.numeric transfer M&F into numbers
#Check this function by ?as.numeric()
```

2. Multiple regression using the lm() function

```
#To perform Multiple Regression,
#we use the same functions as we use in Simple Linear Regression
#Notice that we use "+" between two variables in lm()
mod<-lm(y ~ x1+x2+x3)
summary(mod)

##
## Call:
## lm(formula = y ~ x1 + x2 + x3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.523 -1.440  0.117  1.473  9.114
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.34476    2.74696   5.586 3.08e-08 ***
## x1           0.40598    0.02921  13.900  < 2e-16 ***
## x2           0.32150    0.03128  10.277  < 2e-16 ***
## x3           5.22595    0.14401  36.289  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Residual standard error: 2.154 on 894 degrees of freedom
## Multiple R-squared:  0.6397, Adjusted R-squared:  0.6385
## F-statistic:   529 on 3 and 894 DF,  p-value: < 2.2e-16
```

```
#Notice that 4 p-values are very small,
#which means variables x1,x2,x3
#have strong linear relationship with y.
#We conclude that all β are significantly different from zero.
#Since F = 529>2.615, we reject Ho,
#and conclude that our model predicts height better than by chance.
#Equivalently, F-statistic's p-value: < 2.2e-16, hence we reject Ho.
```

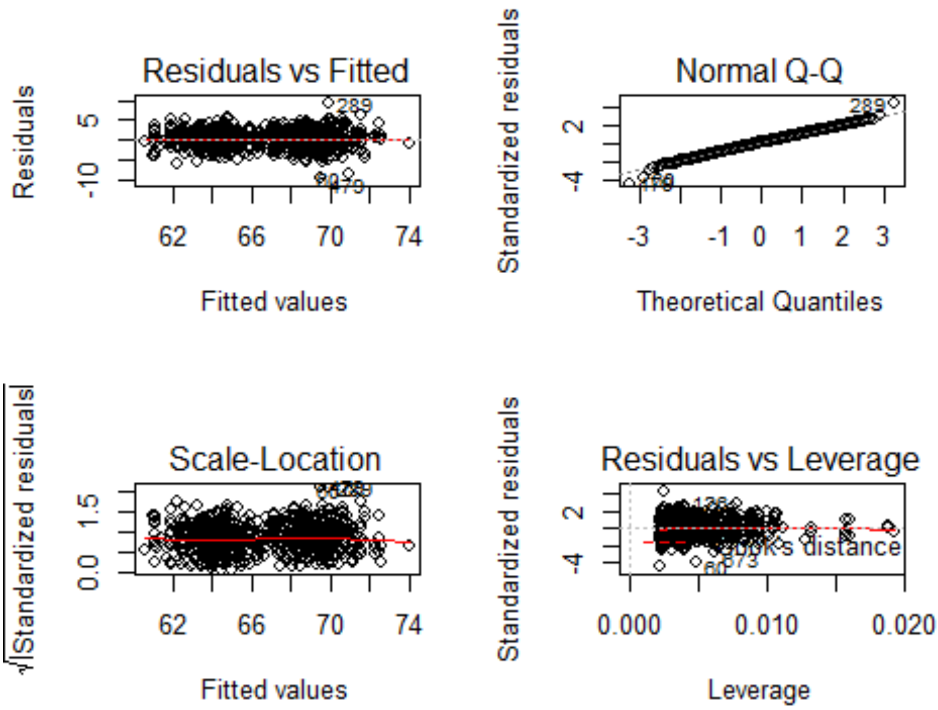
3. Obtain confidence intervals for model parameters

```
#The following function is used to get CI of your "Beta"
confint(mod,level=0.95)   #confint(mod,conf.level=0.95)
```

```
##                 2.5 %     97.5 %
## (Intercept) 9.9535161 20.7360040
## x1          0.3486558  0.4633002
## x2          0.2601008  0.3828894
## x3          4.9433183  5.5085843
```

4. Check model goodness-of-fit

```
par(mfrow=c(2,2))
plot(mod)
```



5. General Linear Model using the glm() function

```
#We can use glm() as well - this is especially convenient when we ha
ve categorical variables in our data set. Instead of creating dummy
variables by ourselves, R can directly work with the categorical var
iables. This is in the same spirit as the Proc GLM procedure in SAS.

#glm {stats}
#Fitting Generalized Linear Models
#Description:
#glm is used to fit generalized linear models, specified by giving a
 symbolic description of the #linear predictor and a description of
the error distribution.
#Usage:
#glm(formula, family = gaussian, data, weights, subset,
#    na.action, start = NULL, etastart, mustart, offset,
#    control = list(...), model = TRUE, method = "glm.fit",
#    x = FALSE, y = TRUE, contrasts = NULL, ...)
#You can see the details by help(glm)


x3<- data$Gender
```

```r
mod1<-glm(y~x1+x2+factor(x3))
#Use factor(x3) to let R knows x3 is a categorical variable

#check by yourself by help(factor)

summary(mod1)

##
## Call:
## glm(formula = y ~ x1 + x2 + factor(x3))
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -9.523  -1.440   0.117   1.473   9.114
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.34476    2.74696   5.586 3.08e-08 ***
## x1           0.40598    0.02921  13.900  < 2e-16 ***
## x2           0.32150    0.03128  10.277  < 2e-16 ***
## factor(x3)M  5.22595    0.14401  36.289  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 4.641121)
##
##     Null deviance: 11515.1  on 897  degrees of freedom
## Residual deviance:  4149.2  on 894  degrees of freedom
## AIC: 3932.8
##
## Number of Fisher Scoring iterations: 2

#The result is similar to summary(mod)
#As we see, R uses “F” as x3's reference level because “F” comes bef
ore “M” in the alphabetic order.

#Now we change it into “M”
x3<-relevel(factor(x3),ref="M")

mod1<-glm(y~x1+x2+factor(x3))

summary(mod1)

##
## Call:
```

```
## glm(formula = y ~ x1 + x2 + factor(x3))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -9.523  -1.440   0.117   1.473   9.114
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.34476    2.74696   5.586 3.08e-08 ***
## x1           0.40598    0.02921  13.900  < 2e-16 ***
## x2           0.32150    0.03128  10.277  < 2e-16 ***
## factor(x3)F -5.22595    0.14401 -36.289  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 4.641121)
##
##     Null deviance: 11515.1  on 897  degrees of freedom
## Residual deviance:  4149.2  on 894  degrees of freedom
## AIC: 3932.8
##
## Number of Fisher Scoring iterations: 2
```

6. Import Data in .xlsx format

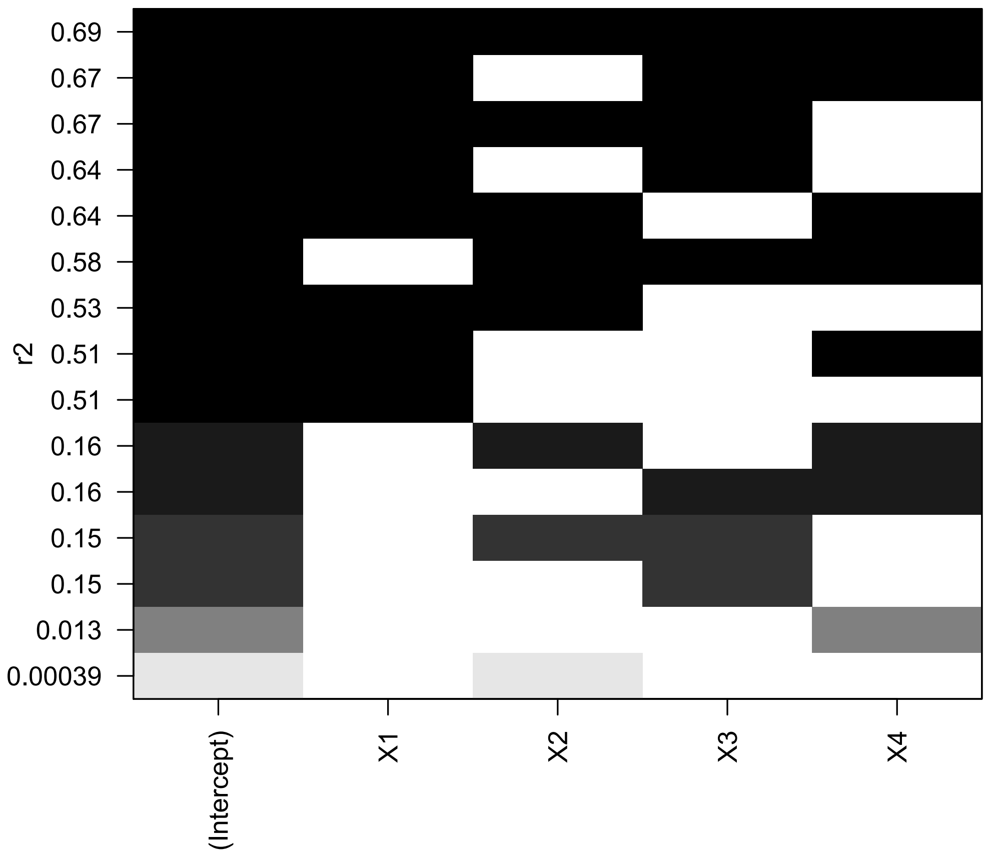
```
#Now we are about to study the variable selection procedures in R. First, we shall analyze the heat data as shown on slide 53 of review material.

#I have saved the data as an excel file in my galton directory.

# Now we need to install the package 'xlsx' to read excel files.

library(xlsx)

data1<-read.xlsx("~/Desktop/heat.xlsx", 1)

#data1<-read.xlsx("d:/ams394/galton/heat.xlsx", 1)

data1
```

7. Best subset variable selection

```
# Now we need to first install the library 'leaps', and then
we call it:

install.packages("leaps")

library(leaps)

attach(data1)

#The attach command above will enable R to use the variables
in the dataset directly.

leaps1<-regsubsets(Y~X1+X2+X3+X4,data=data1,nbest=10)

summary(leaps1)

Subset selection object

Call: regsubsets.formula(Y ~ X1 + X2 + X3 + X4, data = data1, nbest
= 10)

4 Variables  (and intercept)

10 subsets of each size up to 4

Selection Algorithm: exhaustive

         X1  X2  X3  X4

1  ( 1 ) " " " " " " "*"

1  ( 2 ) " " "*" " " " "

1  ( 3 ) "*" " " " " " "

1  ( 4 ) " " " " "*" " "

2  ( 1 ) "*" "*" " " " "

2  ( 2 ) "*" " " " " "*"

2  ( 3 ) " " " " "*" "*"

2  ( 4 ) " " "*" "*" " "

2  ( 5 ) " " "*" " " "*"

2  ( 6 ) "*" " " "*" " "

3  ( 1 ) "*" "*" " " "*"
```

```
3  ( 2 ) "*" "*" "*" " "
3  ( 3 ) "*" " " "*" "*"
3  ( 4 ) " " "*" "*" "*"
4  ( 1 ) "*" "*" "*" "*"
```

plot(leaps1, scale="r2")



plot(leaps1, scale="adjr2")

plot(leaps1, scale="bic")

plot(leaps1, scale="Cp")

8. Stepwise variable selection

#Next, We use step() to perform Stepwise Regression

step selects the model by AIC

step is a slightly simplified version of stepAIC in package MASS

```
step(lm(Y~X1+X2+X3+X4), data=data1)

Start:  AIC=141.71

Y ~ X1 + X2 + X3 + X4


       Df Sum of Sq    RSS    AIC

- X2    1     14764 341378 140.28

- X4    1     17881 344496 140.40

- X3    1     52338 378952 141.64

<none>              326614 141.71

- X1    1    117910 444524 143.72


Step:  AIC=140.29

Y ~ X1 + X3 + X4


       Df Sum of Sq    RSS    AIC

- X4    1     37189 378567 139.63

<none>              341378 140.28

- X3    1    169447 510825 143.53

- X1    1    543234 884612 150.66


Step:  AIC=139.63
```

```
Y ~ X1 + X3

       Df Sum of Sq    RSS    AIC
<none>              378567 139.63
- X3    1    136710 515278 141.64
- X1    1    516693 895261 148.82

Call:
lm(formula = Y ~ X1 + X3)

Coefficients:
(Intercept)           X1           X3
    -635.31        62.28        29.42
summary(step(lm(Y~X1+X2+X3+X4), data=data1))
Start:  AIC=141.71
Y ~ X1 + X2 + X3 + X4

       Df Sum of Sq    RSS    AIC
- X2    1     14764 341378 140.28
- X4    1     17881 344496 140.40
- X3    1     52338 378952 141.64
<none>              326614 141.71
- X1    1    117910 444524 143.72

Step:  AIC=140.29
Y ~ X1 + X3 + X4
```

```
       Df Sum of Sq    RSS    AIC
- X4    1     37189 378567 139.63
<none>              341378 140.28
- X3    1    169447 510825 143.53
- X1    1    543234 884612 150.66

Step:  AIC=139.63
Y ~ X1 + X3

       Df Sum of Sq    RSS    AIC
<none>              378567 139.63
- X3    1    136710 515278 141.64
- X1    1    516693 895261 148.82

Call:
lm(formula = Y ~ X1 + X3)

Residuals:
    Min      1Q  Median      3Q     Max
-205.36 -180.84   -1.74  101.32  368.74

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -635.31     299.52  -2.121  0.05992 .
X1             62.28      16.86   3.694  0.00415 **
X3             29.42      15.48   1.900  0.08658 .
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 194.6 on 10 degrees of freedom

Multiple R-squared:  0.6394,    Adjusted R-squared:  0.5673

F-statistic: 8.865 on 2 and 10 DF,  p-value: 0.006098
```

 # The final model is: $Y = \beta_0 + \beta_1 X1 + \beta_3 X3 + \varepsilon$

9. Different Variable Selection Criteria

 # Please also note that SAS and R may give you different results in variable selection because different selection criteria maybe used. **For example, in SAS, for stepwise variable selection, we use the F-test/Partial correlation. However, in R, we use the AIC criterion. For those of you who are highly interested in this topic, you can study on your own, for example, the following papers:**

https://stat.ethz.ch/R-manual/R-devel/library/stats/html/step.html

http://analytics.ncsu.edu/sesug/2007/SA05.pdf

The AIC (Akaike information criterion) was proposed by Dr. Hirotugu Akaike.

Let L be the maximum value of the likelihood function for the given model, and let k be the number of estimated parameters in the model. The AIC value of the model is: AIC = 2k - 2ln(L)

Given a set of candidate models for the given data, the preferred model is the one with the minimum AIC value. AIC rewards goodness of fit (as assessed by the likelihood function), while including a penalty against overfitting (big k).