Regression Models week1

Reboot: ip <- installed.packages() pkgs.to.remove <- ip[!(ip[,“Priority”] %in% c(“base”, “recommended”)), 1] sapply(pkgs.to.remove, remove.packages)

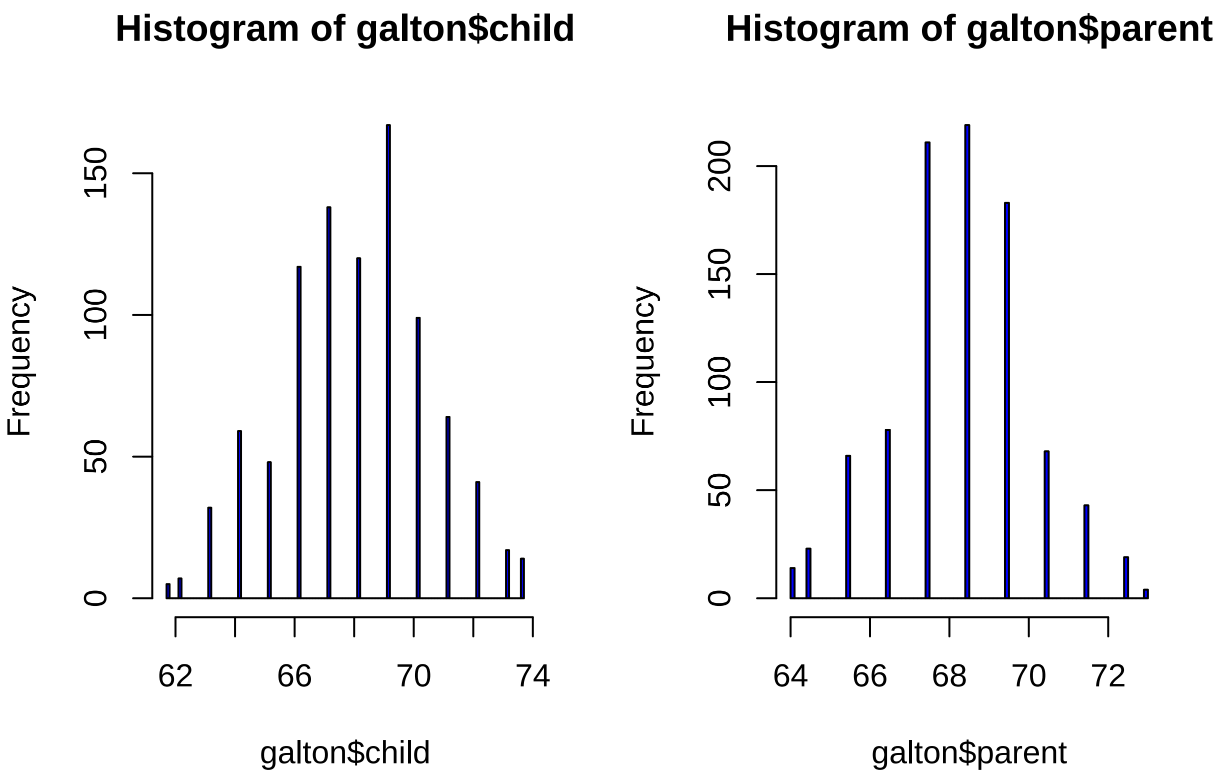
Regression

Galton, cousin of Darwin invented the idea of regression.

```r
library(UsingR)
```

```
## Loading required package: MASS
## Loading required package: HistData
## Loading required package: Hmisc
## Loading required package: grid
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
## Loading required package: ggplot2
##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
##
##
## Attaching package: 'UsingR'
##
## The following object is masked from 'package:ggplot2':
##
##     movies
##
## The following object is masked from 'package:survival':
##
##     cancer
```

```r
data(galton)
par(mfrow = c(1,2))
hist(galton$child, col = "blue", breaks = 100)
hist(galton$parent, col = "blue", breaks = 100)
```



Looks fairly similar, without the pairing.

Least square

The least square is defined as

$$\mu \rightarrow min \sum_{i=1}^{n} (Y_i - \mu)^2$$

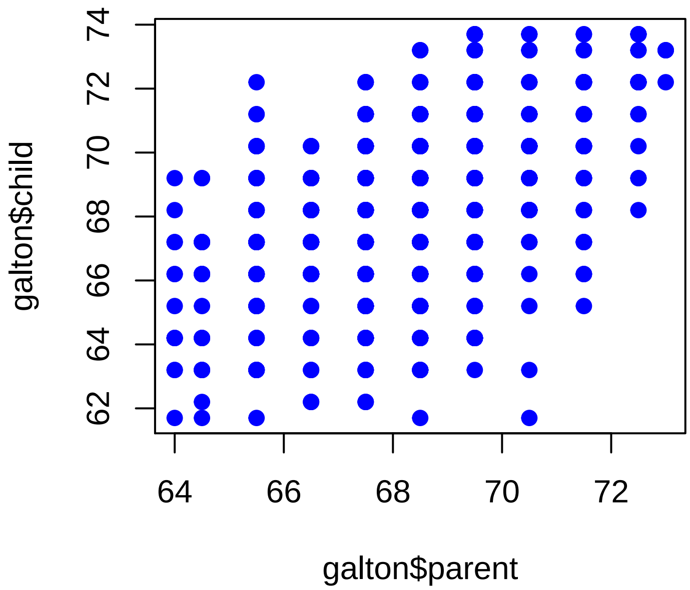
which is obviously

$$\overline{X} = \mu$$

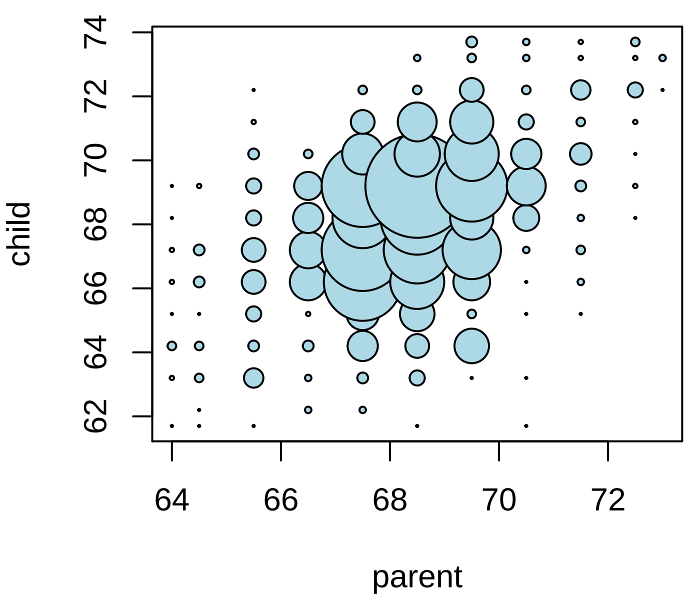
```
par(mfrow = c(1,1))
library(manipulate)
myHist <- function(mu){
  hist(galton$child, col = "blue", breaks = 100)
  lines(c(mu,mu), c(0,150), col = "red", lwd = "5")
  mse <- mean((galton$child - mu)^2)
  text(63, 150, paste("mu = ", mu))
  text(63, 140, paste("MSE = ", round(mse, 2)))
}
manipulate(myHist(mu), mu = slider(62, 74, step = 0.5))
```

The least squares estimate is the empiracal mean

```
plot(galton$parent, galton$child, pch =19, col = "blue")
```




```
freqData <- as.data.frame(table(galton$child, galton$parent))
names(freqData) <- c("child", "parent", "freq")
plot(as.numeric(as.vector(freqData$parent)),
     as.numeric(as.vector(freqData$child)),
     pch = 21, col = "black", bg = "lightblue",
     cex = .15 * freqData$freq,
     xlab = "parent", ylab = "child")
```



Regression through the origin

```
myPlot <- function(beta){
  y <- galton$child - mean(galton$child)
  x <- galton$parent - mean(galton$parent)
```

```r
  freqData <- as.data.frame(table(x, y))
  names(freqData) <- c("child", "parent", "freq")
  plot(
    as.numeric(as.vector(freqData$parent)),
    as.numeric(as.vector(freqData$child)),
    pch = 21, col = "black", bg = "lightblue",
    cex = .15 * freqData$freq,
    xlab = "parent",
    ylab = "child"
    )
  abline(0, beta, lwd = 3)
  points(0, 0, cex = 2, pch = 19)
  mse <- mean( (y - beta * x)^2 )
  title(paste("beta = ", beta, "mse = ", round(mse, 3)))
}
manipulate(myPlot(beta), beta = slider(0.6, 1.2, step = 0.02))
```

R can do this

```r
lm(I(child - mean(child)) ~ I(parent - mean(parent)) -1, data = galton)
```

```
##
## Call:
## lm(formula = I(child - mean(child)) ~ I(parent - mean(parent)) -
##     1, data = galton)
##
## Coefficients:
## I(parent - mean(parent))
##                   0.6463
```

Basic notations

Sample variance, Sample covariance, sample corelation. also called emprical.

Linear least squares

$$\sum_{i=1}^{n}\{Y_i - (\beta_0 + \beta_1 X_i)\}^2$$

It turns out to be

$$\hat{\beta}_0 = \overline{Y} - \hat{\beta}_1 \overline{X}$$

$$\hat{\beta}_1 = Cor(X, Y)\frac{Sd(X)}{Sd(Y)}$$

If you normalize the data, then

$$\hat{\beta}_1 = Cor(X, Y)$$

Revisiting Galton's data

```
y <- galton$child
x <- galton$parent
beta1 <- cor(y,x)*sd(y)/sd(x)
beta0 <- mean(y) - beta1*mean(x)
rbind(c(beta0, beta1), coef(lm(y~x)))
```

```
##      (Intercept)         x
## [1,]    23.94153 0.6462906
## [2,]    23.94153 0.6462906
```

Center the origin

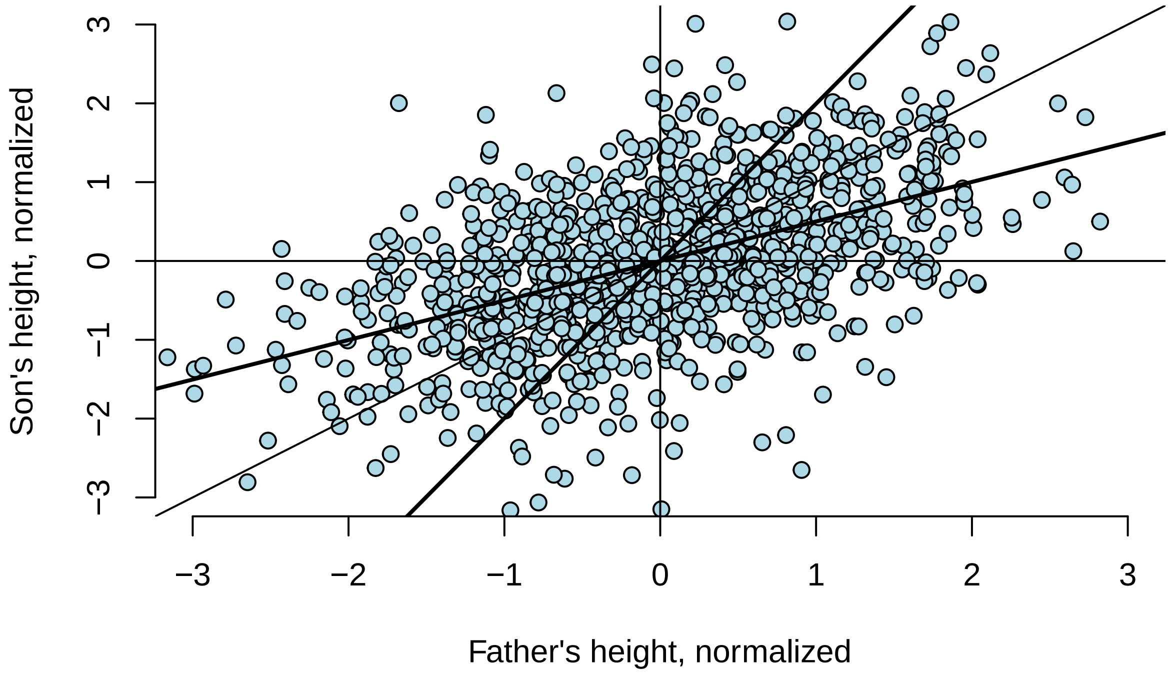
```
yc <- y - mean(y)
xc <- x - mean(x)
beta1 <- sum(yc*xc)/sum(xc*xc)
c(beta1, coef(lm(y~x))[2])
```

```
##                   x
## 0.6462906 0.6462906
```

Regression to the mean

- normalize x and y (child and parent height)
- slope is Cor(Y,X)

```
library(UsingR)
data(father.son)
y <- (father.son$sheight - mean(father.son$sheight)) / sd(father.son$sheight)
x <- (father.son$fheight - mean(father.son$fheight)) / sd(father.son$fheight)
rho <- cor(x,y)
myPlot <- function(x,y){
  plot(x,y,
       xlab = "Father's height, normalized",
       ylab = "Son's height, normalized",
       xlim = c(-3,3),
       ylim = c(-3,3),
       bg = "lightblue", col = "black", cex = 1.1, pch = 21,
       frame = FALSE)
}
myPlot(x,y)
abline(0,1) # perfect correlation
abline(0, rho, lwd = 2) # father predicts son
abline(0, 1/rho, lwd = 2) # son predicts father
abline(h =0); abline(v = 0) # no relationship
```




```
?abline
par(mfrow = c(1,1))
```

Quiz

```
x <- c(0.18, -1.54, 0.42, 0.95)
w <- c(2, 1, 3, 1)
mu <- sum(w*x)/sum(w)
mu
```

```
## [1] 0.1471429
```

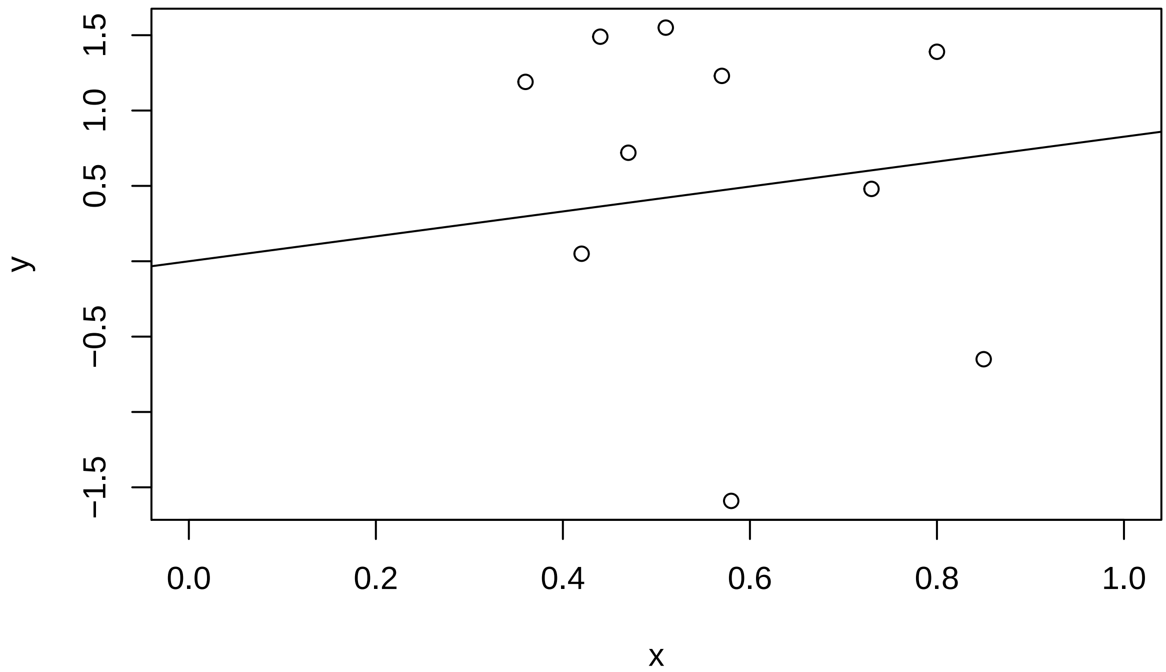
2

fit the regression throught the origin

```
x <- c(0.8, 0.47, 0.51, 0.73, 0.36, 0.58, 0.57, 0.85, 0.44, 0.42)
y <- c(1.39, 0.72, 1.55, 0.48, 1.19, -1.59, 1.23, -0.65, 1.49, 0.05)
fit <- lm( y ~ x -1)
fit
```

```
##
## Call:
## lm(formula = y ~ x - 1)
##
## Coefficients:
##      x
## 0.8263
```

```r
plot(x,y,xlim=c(0,1))
abline(0,fit$coefficients)
```



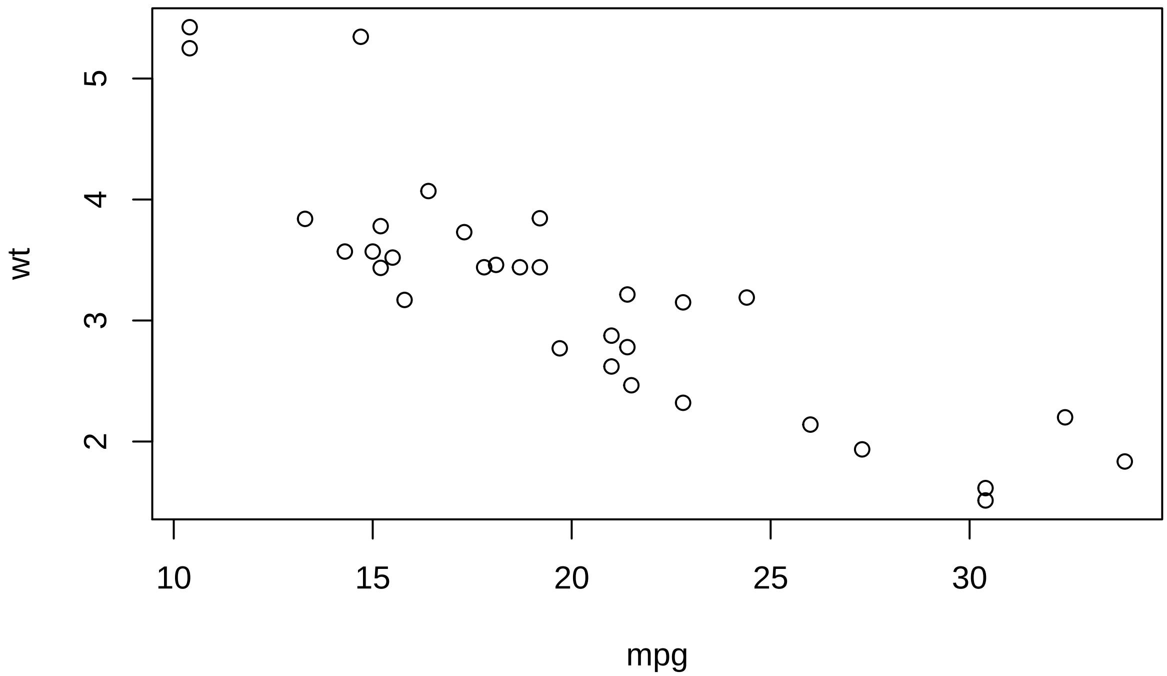
3

from data(mtcars), mpg as outcome and weight as predictor

```r
data(mtcars)
fit <- lm(mtcars$mpg ~ mtcars$wt)
fit
```

```
##
## Call:
## lm(formula = mtcars$mpg ~ mtcars$wt)
##
## Coefficients:
## (Intercept)    mtcars$wt
##      37.285       -5.344
```

```r
with(mtcars, plot(mpg, wt))
```



4

sd(x) = 1/2
sd(y) = 1
cor(y,x) = .5
the slope is

```
sdx = 1/2
sdy = 1
cor = .5
cor * sdy / sdx
```

```
## [1] 1
```

5

Students were given two hard tests and scores were normalized to have empirical mean 0 and variance 1. The correlation between the scores on the two tests was 0.4. What would be the expected score on Quiz 2 for a student who had a normalized score of 1.5 on Quiz 1?

```
1.5*0.4
```

```
## [1] 0.6
```

6

Consider the data given by the following x <- c(8.58, 10.46, 9.01, 9.64, 8.86) What is the value of the first measurement if x were normalized (to have mean 0 and variance 1)?

```r
x <- c(8.58, 10.46, 9.01, 9.64, 8.86)
(x - mean(x))/sd(x)
```

```
## [1] -0.9718658  1.5310215 -0.3993969  0.4393366 -0.5990954
```

7

Consider the following data set (used above as well). What is the intercept for fitting the model with x as the predictor and y as the outcome?

```r
x <- c(0.8, 0.47, 0.51, 0.73, 0.36, 0.58, 0.57, 0.85, 0.44, 0.42)
y <- c(1.39, 0.72, 1.55, 0.48, 1.19, -1.59, 1.23, -0.65, 1.49, 0.05)
lm(y ~ x)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##       1.567       -1.713
```

8

You know that both the predictor and response have mean 0. What can be said about the intercept when you fit a linear regression?