Introduction to Regression - Week 1 Notes

Trenton Potgieter

Monday, April 06, 2015

Contents

Francis Galton

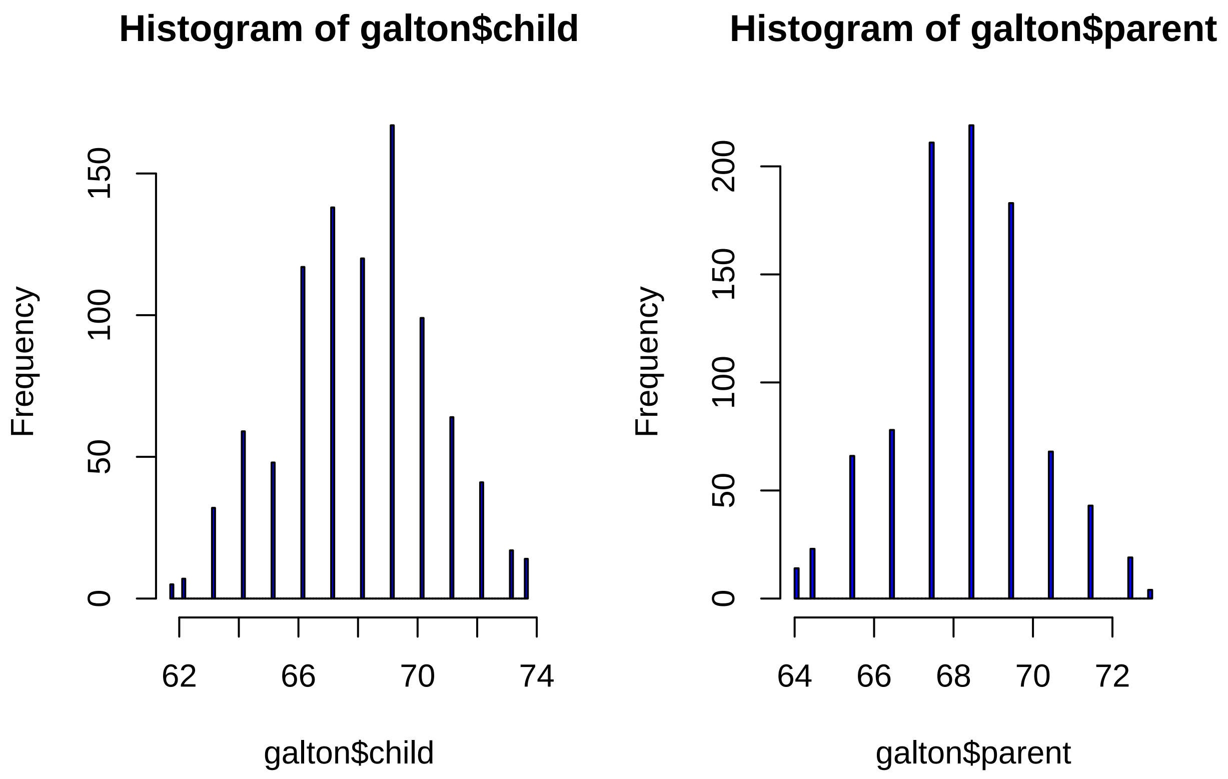
Background

Data first used by Francis Galton, who created the terms **Regression** and **Correlation** in 1885. And by making use of **Rgression**, we are provided with very interpretable models.

```
require(UsingR)
```

```
## Loading required package: UsingR

## Warning: package 'UsingR' was built under R version 3.1.3

## Loading required package: MASS

## Warning: package 'MASS' was built under R version 3.1.2

## Loading required package: HistData

## Warning: package 'HistData' was built under R version 3.1.3

## Loading required package: Hmisc

## Warning: package 'Hmisc' was built under R version 3.1.3

## Loading required package: grid
## Loading required package: lattice
## Loading required package: survival
## Loading required package: splines
## Loading required package: Formula

## Warning: package 'Formula' was built under R version 3.1.3

## 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
#Load the Data
data(galton)

#Plot the Child and parent data
par(mfrow = c(1, 2))
hist(galton$child, col = "blue", breaks = 100) #100 histogram breaks
hist(galton$parent, col = "blue", breaks = 100) #100 histogram breaks
```



The plots above do not descibe the joint relationship. To uderstand the joint relationship we need to first understand summarizing the **marginal**. The marginal is the distribution (on the histograms) of **Children**, disregarding **Parents** and the distibution of **Parents**, disregarding **Children**, so summarizing the marginal informaion in each Histogram by themselves is a way of describing the "middle" of these datasets.

To do this, let's consider the **children's heights**. So let Y_i be the height of a particular **child** i for $i = 1, \ldots, n$, where $n = 928$ (the number of **Children**).

So to define the middle we look for the value of μ that mimimizes

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

The sum of the squared distances between the data and the "middle" value.

This turns out to be the center of mass of the histogram, the point the mimimizes the average squared distance from all the other points (Least Squares). So in this case the answer is the **sample mean** or $\mu = \bar{X}$.

Remember that in Statistics, $\bar{X}$ is the **sample mean** and μ is the **population mean**.

Proof

To prove this we will use the `manipulate()` function in `R` to se what value of μ minimizes the sum of squared deviations.

```
require(manipulate)
```

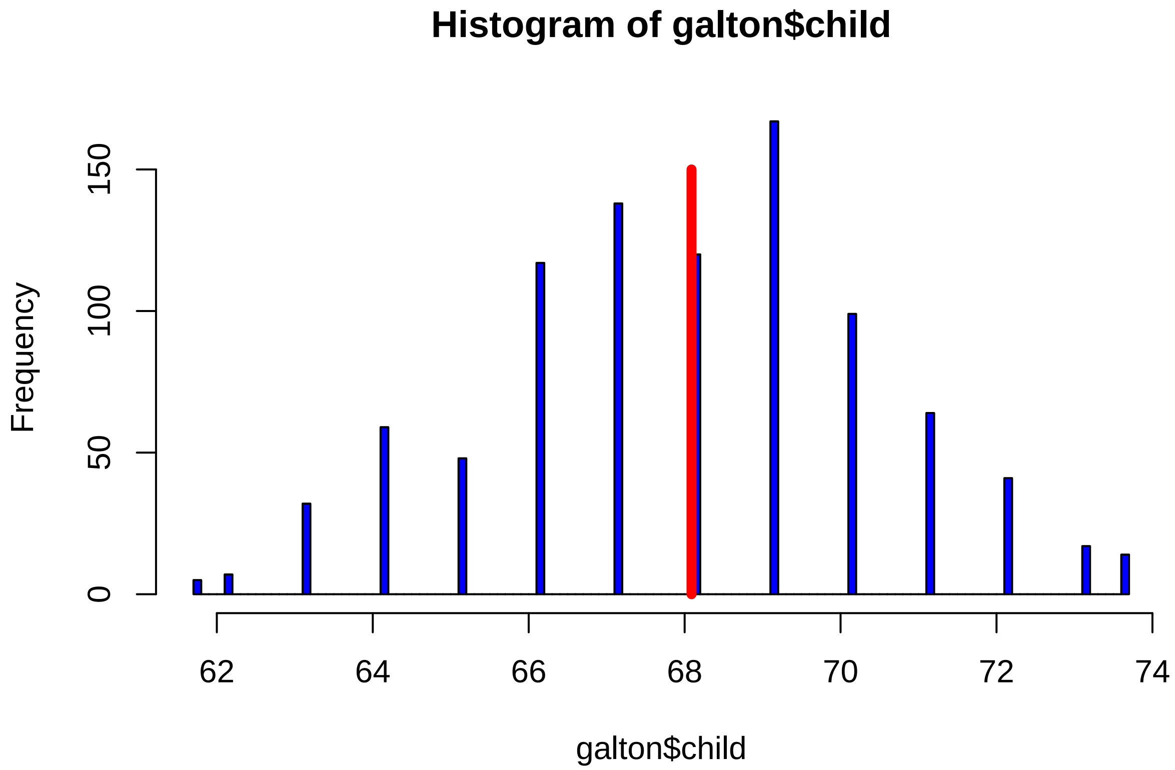
```
## Loading required package: manipulate
```

```
#Create the Hist() Function
Hist <- function(mu){
        #Create a histogram of the child data as before
        hist(galton$child, col = "blue", breaks = 100)
        #Draw the line that can be used by manipulate
        lines(c(mu, mu), c(0, 150), col = "red", lwd = 5)
        #Clculate the mean squared error
        mse <- mean((galton$child - mu)^2)
        #Create the labels
        text(63, 150, paste("mu = ", mu))
        text(63, 140, paste("MSE = ", round(mse, 2)))
}

#To call this and use manipulate in R, simply run the following:
#manipulate(Hist(mu), mu = slider(62, 74, step = 0.5))
```

So even though this can be seen visually, by using the `manipulate()` function and manually moving around the "red line", to find the exact optimal place that will balance out the Histogram.

Below is the actualy Histrogram of the **Empirical Mean** of **68.0884698**.

```
#Plot the Empirical Mean
hist(galton$child, col = "blue", breaks = 100)
mean.child <- mean(galton$child)
lines(rep(mean.child, 100), seq(0, 150, length = 100), col = "red", lwd = 5)
```

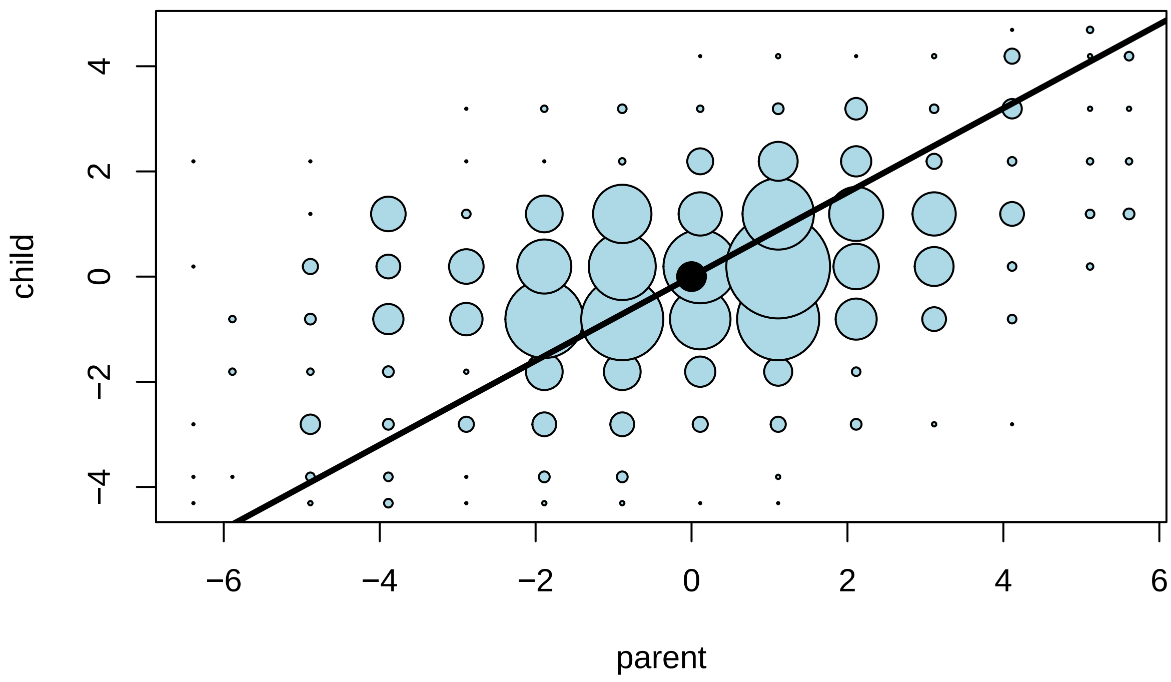


Comparing Children's Heights vs. Parebt's Heights

Doing this comparision is the heart of regression, basically how do we draw a line through Galton's Data as the following example plot shows:

NOTE: Size of point represents number of points at the (X, Y) combination. Additionally the regression line is centered through the average value.

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

#To call this and use manipulate in R, simply run the following:
#manipulate(myPlot(beta), beta = slider(0.8, 1.2, step = 0.02))

#sample plot
myPlot(0.8)
```

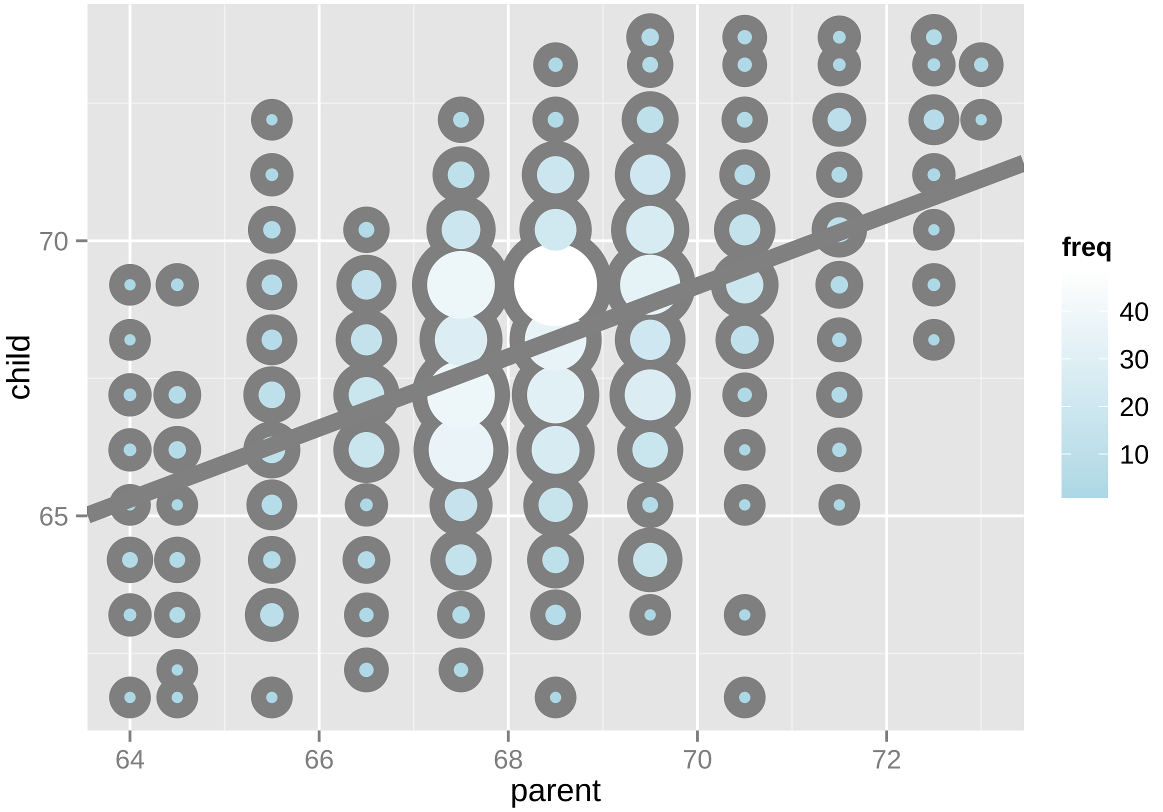


```
require(dplyr)
```

```
## Loading required package: dplyr

## Warning: package 'dplyr' was built under R version 3.1.3

##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:Hmisc':
##
##     combine, src, summarize
##
## The following object is masked from 'package:MASS':
##
##     select
```

```
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
freqData <- as.data.frame(table(galton$child, galton$parent))
names(freqData) <- c("child", "parent", "freq")
freqData$child <- as.numeric(as.character(freqData$child))
freqData$parent <- as.numeric(as.character(freqData$parent))
g <- ggplot(filter(freqData, freq > 0), aes(x = parent, y = child))
g <- g  + scale_size(range = c(2, 20), guide = "none" )
g <- g + geom_point(colour="grey50", aes(size = freq+20, show_guide = FALSE))
g <- g + geom_point(aes(colour=freq, size = freq))
g <- g + scale_colour_gradient(low = "lightblue", high="white")
lm1 <- lm(galton$child ~ galton$parent)
g <- g + geom_abline(intercept = coef(lm1)[1], slope = coef(lm1)[2], size = 3, colour = grey(.5))
g
```



```
lm1
```

```
##
```

```
## Call:
## lm(formula = galton$child ~ galton$parent)
##
## Coefficients:
##   (Intercept)  galton$parent
##       23.9415         0.6463
```