Solving Linear Regression Equations

Nouman Riaz

September 29, 2016

The objective to find 'Beta' parameters of Linear Regression equation for both Simple and Multiple Linear Regression techniques.

Part 01 - Simple Linear Regression

The dataset we are using is advertisment dataset from the ISLR website.

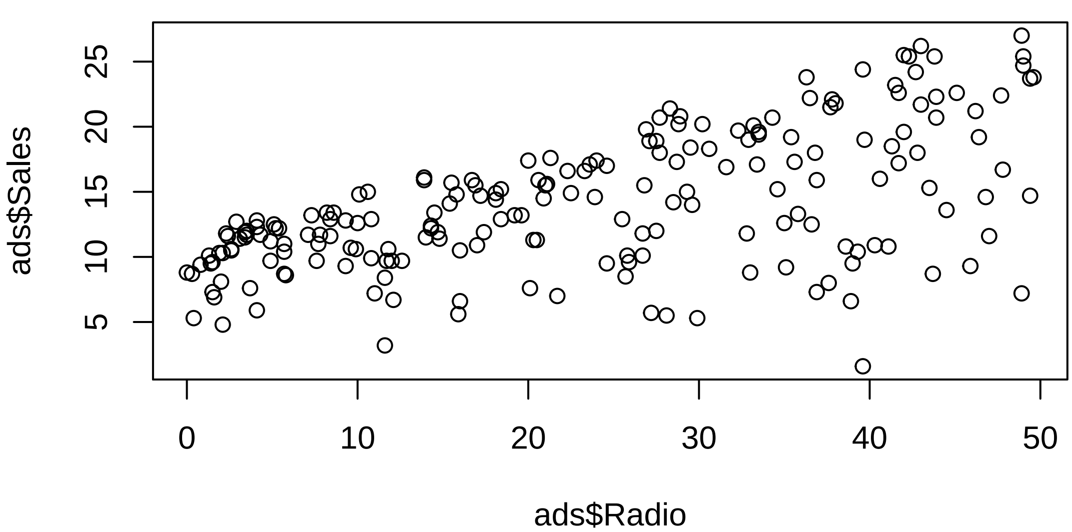
```r
ads <- read.csv("Advertising.csv", stringsAsFactors=FALSE)
head(ads, n=5) #Observing few dataset entries
```

```
##   X    TV Radio Newspaper Sales
## 1 1 230.1  37.8      69.2  22.1
## 2 2  44.5  39.3      45.1  10.4
## 3 3  17.2  45.9      69.3   9.3
## 4 4 151.5  41.3      58.5  18.5
## 5 5 180.8  10.8      58.4  12.9
```

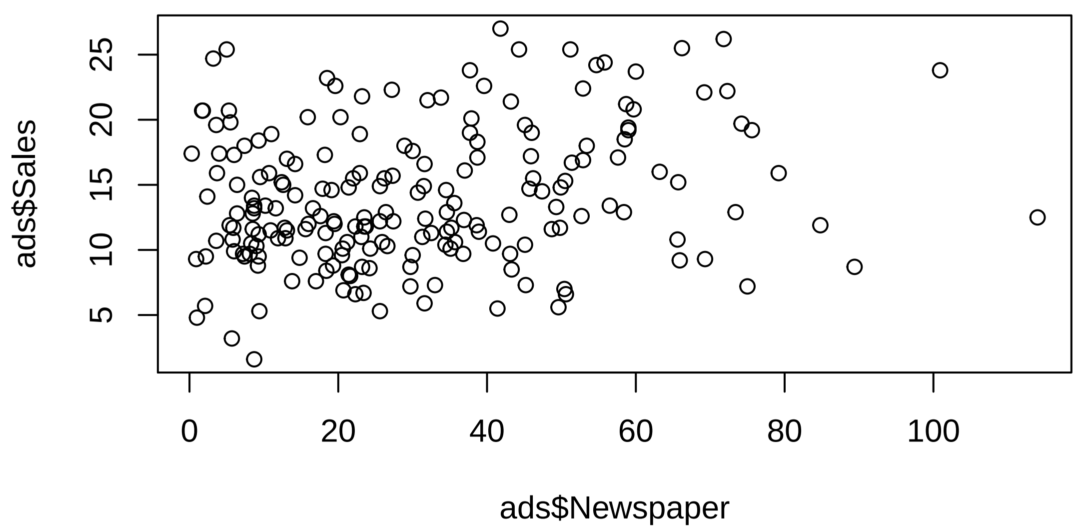
Now we got to find parameters of two linear equations:

1. Sales ~ Radio
2. Sales ~ Newspaper

```r
#Visualizing these input variables agains the output
plot(ads$Radio, ads$Sales)
```



```r
plot(ads$Newspaper, ads$Sales)
```



Now solving the following equations to estimate our Beta Values

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - x^{-})(y_i - y^{-})}{\sum_{i=1}^{n}(x_i - x^{-})^2}$$

$$\beta_0 = y^{-} - \beta_1 x^{-}$$

1. Sales ~ Radio

```r
x_bar = sum(ads$Radio)/length(ads$Radio) #Calculating xbar, i.e. mean of x
y_bar = sum(ads$Sales)/length(ads$Sales) #Calculating ybar, i.e. mean of y

t = ads$Radio - x_bar #Solving numerator of the equation
s = ads$Sales - y_bar #Solving denominator of the equation

beta_1 <- sum(t*s)/sum(t*t) #Calculating Beta1
beta_0 <- y_bar - (beta_1 * x_bar) #Calculating Beta0

cat(sprintf("Beta_one = %s\n",beta_1))
```

```
## Beta_one = 0.20249578339244
```

```r
cat(sprintf("Beta_zero = %s",beta_0))
```

```
## Beta_zero = 9.31163809515828
```

Now calculating β values for second variable

2. Sales ~ Newspaper

```r
x_bar = sum(ads$Newspaper)/length(ads$Newspaper) #Calculating xbar, i.e. mean of x
y_bar = sum(ads$Sales)/length(ads$Sales)#Calculating ybar, i.e. mean of y

t = ads$Newspaper - x_bar #Solving numerator of the equation
s = ads$Sales - y_bar #Solving denominator of the equation

beta_1 <- sum(t*s)/sum(t*t) #Calculating Beta1
beta_0 <- y_bar - (beta_1 * x_bar) #Calculating Beta0

cat(sprintf("Beta_one = %s\n",beta_1))
```

```
## Beta_one = 0.0546930984722734
```

```r
cat(sprintf("Beta_zero = %s",beta_0))
```

```
## Beta_zero = 12.3514070692782
```

Part 02 - Multiple Linear Regression

i.e. Sales ~ TV+Radio+Newspaper

```r
library(MASS)
X <- matrix(c(seq(from=1,to=1,length.out = nrow(ads)),ads$TV,ads$Radio,ads$Newspaper), nrow=nrow(ads), n
Xt <- t(X) #Taking transpose of input matrix
y <- ads$Sales #Output variable

a <- Xt %*% X #Multiplaying input matrix with its transpose
b <- Xt %*% y #Multiplaying input matrix with output variable
inv <- ginv(a) #Taking inverse
BetaV <- inv %*% b #Caculating Beta vector
cat(BetaV)
```

```
## 2.938889 0.04576465 0.18853 -0.001037493
```

Verifying

```r
model <- lm(formula=Sales~TV+Radio+Newspaper,data=ads)
summary(model)
```

```
##
## Call:
## lm(formula = Sales ~ TV + Radio + Newspaper, data = ads)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8277 -0.8908  0.2418  1.1893  2.8292
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.938889   0.311908   9.422   <2e-16 ***
## TV           0.045765   0.001395  32.809   <2e-16 ***
## Radio        0.188530   0.008611  21.893   <2e-16 ***
## Newspaper   -0.001037   0.005871  -0.177     0.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.686 on 196 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956
## F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

We can see that the 'Estimate' parameter of our Model is exactly same as our calculated Beta values.