# Simple Regression Tutorial

*Devarshi Pancholi*

*7/5/2019*

## R Markdown

**Task:** The purpose of this analysis/tutorial is to use simple regression to accurately forecast Sales based on the Ad Expenditure in our AdExp dataset

The Steps we will take are:

1) Load the dataset in RStudio and have a look at the data.
2) Look at the scatterplot for Sales vs Expenditure.
3) Check the correlation between variables.
4) Build the linear regression model.
5) Plot the original data and the regression line.
6) Plot the standardized residuals vs. fitted values.

### Part 1: Load the libraries and dataset in RStudio.

```
#loading the libraries we will use for this exercise
library(readr)
library(ggplot2)

#loading the dataset in RStudio
AdExp<-read_csv(file= "/Users/devarshipancholi/Desktop/AdExp.csv")
```

Now lets have a look at our dataset

```
#printing the data to the pdf file
print(AdExp)
```

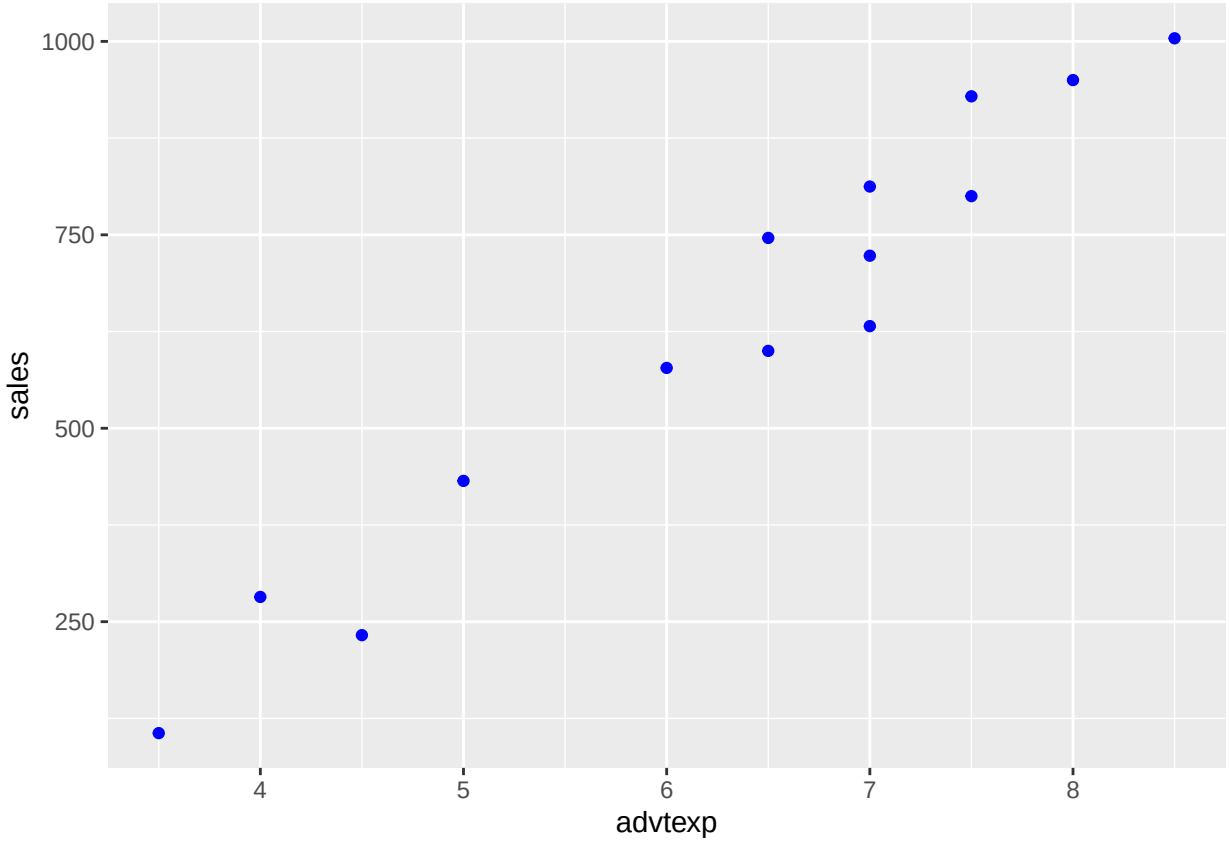
```
## # A tibble: 14 x 2
##    sales advtexp
##    <dbl>   <dbl>
##  1  432      5
##  2  723      7
##  3  578      6
##  4  600      6.5
##  5  950      8
##  6  106      3.5
##  7  282      4
##  8  233.     4.5
##  9  746      6.5
## 10  812.     7
## 11  800      7.5
## 12  929      7.5
## 13 1004      8.5
## 14  632      7
```

**Part 2: Look at the scatterplot for Sales vs Expenditure.** #the values are in 000's

To plot the regression line and scatterplot, I have used the library ggplot which we loaded in the beginning:

```
# command for loading the plot and assigning the axis + plotting the points on graph in blue

ggplot(data= AdExp, aes(x= advtexp,y= sales)) + geom_point(color= 'blue')
```




As we can see, the graph looks roughly linear.

**Part 3: Checking the co-relation**

```
# 'all.obs' is used here as there are no missing data.
# 'pearson' method is used as our data is linear and normally distributed

cor(AdExp, use="all.obs", method="pearson")
```

```
##             sales   advtexp
## sales   1.0000000 0.9743532
## advtexp 0.9743532 1.0000000
```

**Part 4:Build the linear regression model.**

The lm() function is used here to build linear regression models for comparison (sales vs. advtexp). summary() is used for the model to get the R-Squared values.

```r
# simple regression model stored in the variable mamed "linearRegModel"

linearRegModel<- lm(sales ~ advtexp, data= AdExp)
print(linearRegModel)
```

```
##
## Call:
## lm(formula = sales ~ advtexp, data = AdExp)
##
## Coefficients:
## (Intercept)      advtexp
##      -495.3        178.1
```

Next, we get the summary for our regression model. This function allows us to observe a number of values like R-Squared, t-values and p-values for our varialbe "advtexp".

```r
#getting the summary
summary(linearRegModel)
```

```
##
## Call:
## lm(formula = sales ~ advtexp, data = AdExp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -119.346  -37.380   -4.869   54.998   88.609
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -495.28      77.06  -6.427 3.27e-05 ***
## advtexp       178.09      11.87  15.000 3.89e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 65.36 on 12 degrees of freedom
## Multiple R-squared:  0.9494, Adjusted R-squared:  0.9451
## F-statistic:   225 on 1 and 12 DF,  p-value: 3.888e-09
```

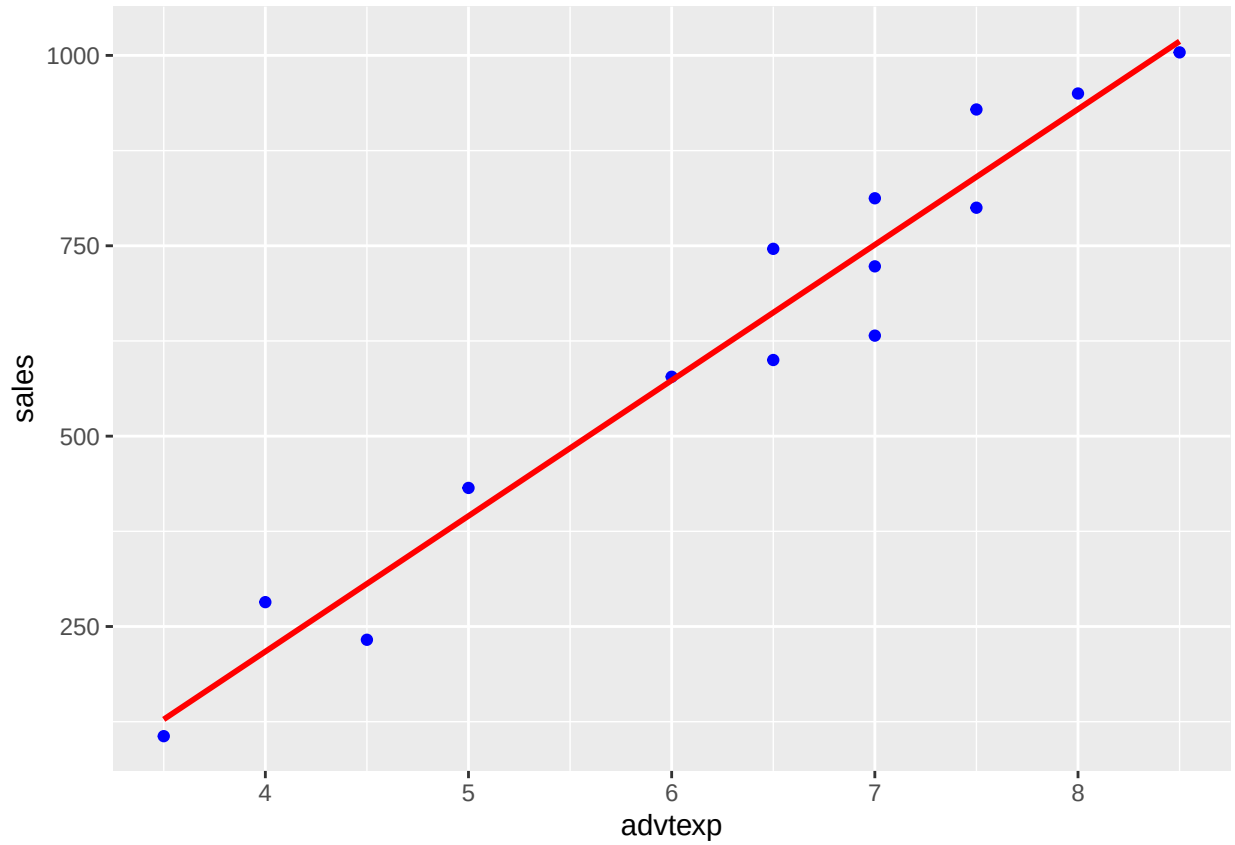
***Part 5: Plot the original data and the linear regression line.***

We will again use the library ggplot2 to add the regression line

```r
# assigning the dataset
data(AdExp)

# deciding on X and Y axis + plotting the points on graph in blue
p1 = ggplot( data= AdExp,aes( x= advtexp,y= sales)) + geom_point( color= 'blue')

# plotting the regression line through the points
p1 + geom_smooth( method= 'lm', se= F, col= "red")
```



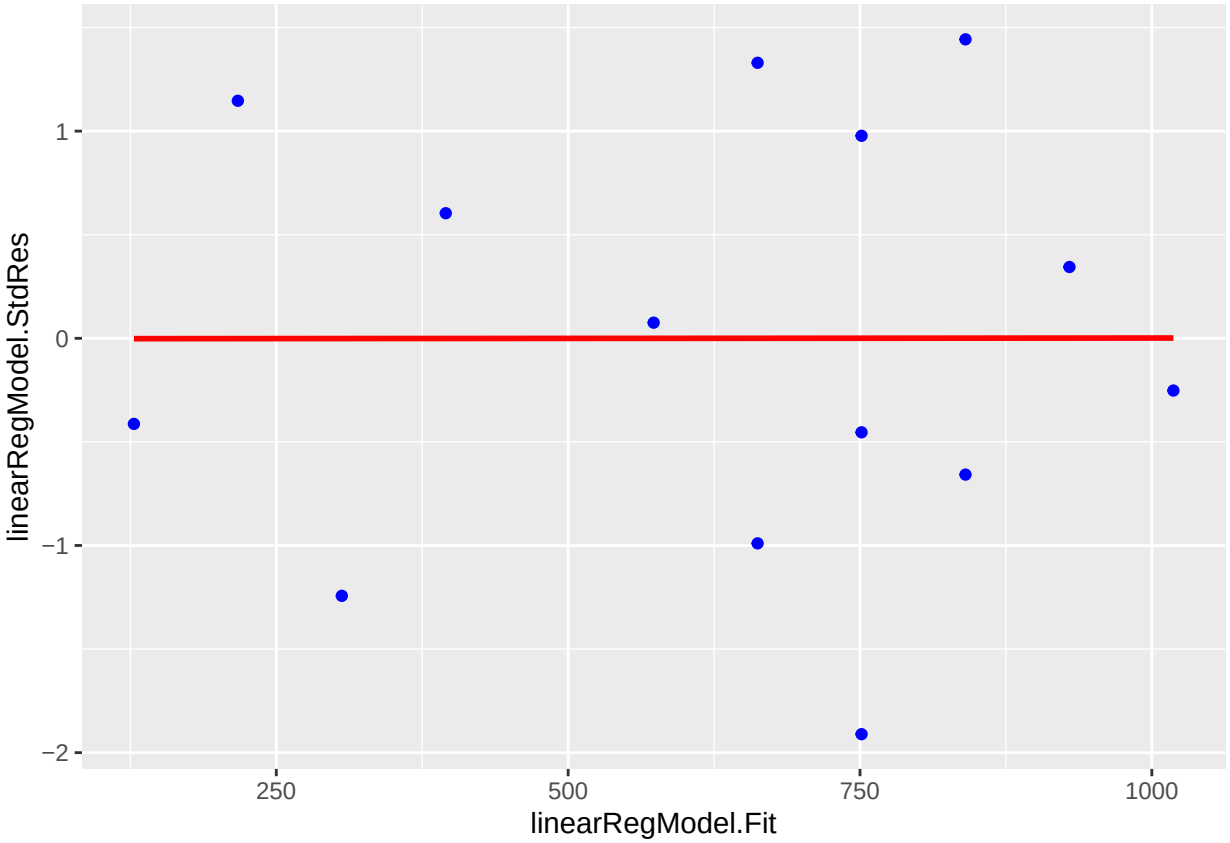

***Part 6: Plot the standardized residuals vs. fitted values.***

```
# obtaining standard residuals
linearRegModel.StdRes <- rstandard(linearRegModel)

# obtaining fitted values
linearRegModel.Fit <- fitted.values(linearRegModel)

# deciding on X and Y axis + plotting the points on graph in blue
p3=ggplot(data=AdExp,aes(x=linearRegModel.Fit,y=linearRegModel.StdRes))+geom_point(color='blue')

# plotting the best fitting line through the points in red
p3 + geom_smooth( method= 'lm', se= F, col= "red")
```


linearRegModel.StdRes
1
0
−1
−2
250
500
750
1000
linearRegModel.Fit