Activity on Simple Regression

R Markdown

Task: The purpose of this analysis is to use simple regression to accurately forecast Monthly Returns based on the Earnings Per Share provide for the year 2005 in our EPS 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 EPS vs Returns.
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.

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

#loading the dataset in RStudio
EPS<-read_csv(file= "/Users/devarshipancholi/Downloads/ReturnsVsEPS05ForHandout8.csv")
```

Now lets have a look at our dataset

```r
#printing the data to the pdf file
print(EPS)
```

```
## # A tibble: 51 x 12
##    Symbol Company `EPS Q4 2004` `2004 Release D~ `Stck Price Day~
##    <chr>  <chr>           <dbl> <chr>                      <dbl>
##  1 MMM    3M Com~         0.91  18-Jan-05 BMO                81.2
##  2 ABT    Abbott~         0.67  18-Jan-05 BMO                44.5
##  3 AA     Alcoa ~         0.39  10-Jan-05 AMC                29.4
##  4 ALL    Allsta~         1.42  2-Feb-05 AMC                 49.3
##  5 AMGN   Amgen           0.580 27-Jan-05 BMO                63.6
##  6 T      AT&T I~         0.34  26-Jan-05 BMO                22.6
##  7 BHI    Baker ~         0.53  17-Feb-05 BMO                46.8
##  8 BA     Boeing~         0.23  2-Feb-05 BMO                 49.8
##  9 BNI    Burlin~         0.91  25-Jan-05 BMO                43.8
## 10 CAT    Caterp~         0.78  27-Jan-05 BMO                44.4
## # ... with 41 more rows, and 7 more variables: `Stck Prce 1 Mo Aftr Q4 04
## #   Relse` <dbl>, `Monthly Return after Q4 2004` <dbl>, `EPS Q4
## #   2005` <dbl>, `2005 Release Date` <chr>, `Stck Price Day Bfr Q4 05
## #   Relse` <dbl>, `Stck Prce 1 Mo Aftr Q4 05 Relse` <dbl>, `Monthly Return
## #   after Q4 2005` <dbl>
```

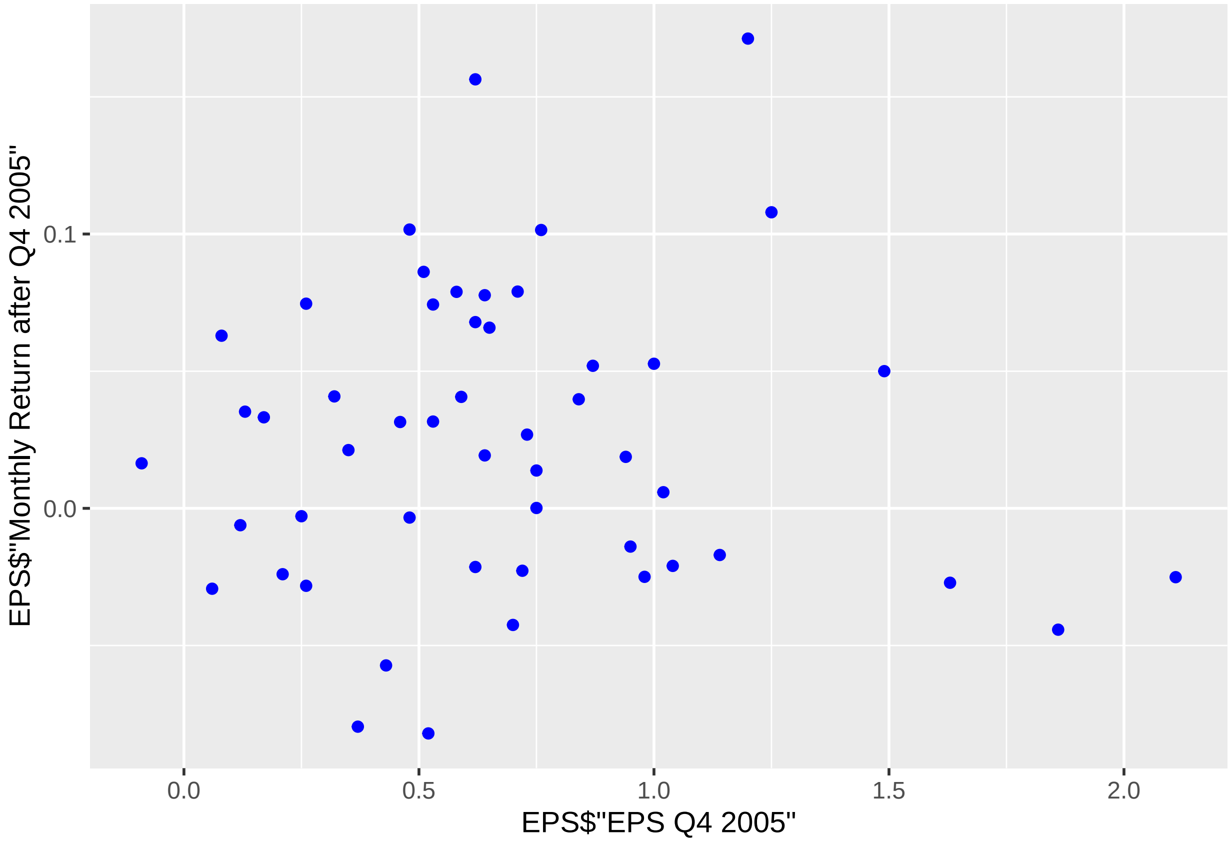
Part 2: Look at the scatterplot for Sales vs Expenditure. #the values are in 000's

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

```
library(ggplot2)

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

ggplot(data=EPS,aes(x=EPS$'EPS Q4 2005',y=EPS$'Monthly Return after Q4 2005'))+geom_point(color='blue')
```



The scatterplot indicates there is little or no presence of a linear relationship. Lets verify this:

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(EPS$'EPS Q4 2005',EPS$'Monthly Return after Q4 2005')
```

```
## [1] -0.02468796
```

Part 4:Build the linear regression model.

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

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

linearRegModel<- lm(EPS$'Monthly Return after Q4 2005' ~ EPS$'EPS Q4 2005', data= EPS)
print(linearRegModel)
```

```
##
## Call:
## lm(formula = EPS$"Monthly Return after Q4 2005" ~ EPS$"EPS Q4 2005",
##     data = EPS)
##
## Coefficients:
##       (Intercept)  EPS$"EPS Q4 2005"
##          0.026827          -0.003015
```

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 “EPS Q4”.

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

```
##
## Call:
## lm(formula = EPS$"Monthly Return after Q4 2005" ~ EPS$"EPS Q4 2005",
##     data = EPS)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.107300 -0.045115 -0.004537  0.038672  0.148030
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.026827   0.014218   1.887   0.0651 .
## EPS$"EPS Q4 2005" -0.003015   0.017444  -0.173   0.8635
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05542 on 49 degrees of freedom
## Multiple R-squared:  0.0006095,  Adjusted R-squared:  -0.01979
## F-statistic: 0.02988 on 1 and 49 DF,  p-value: 0.8635
```

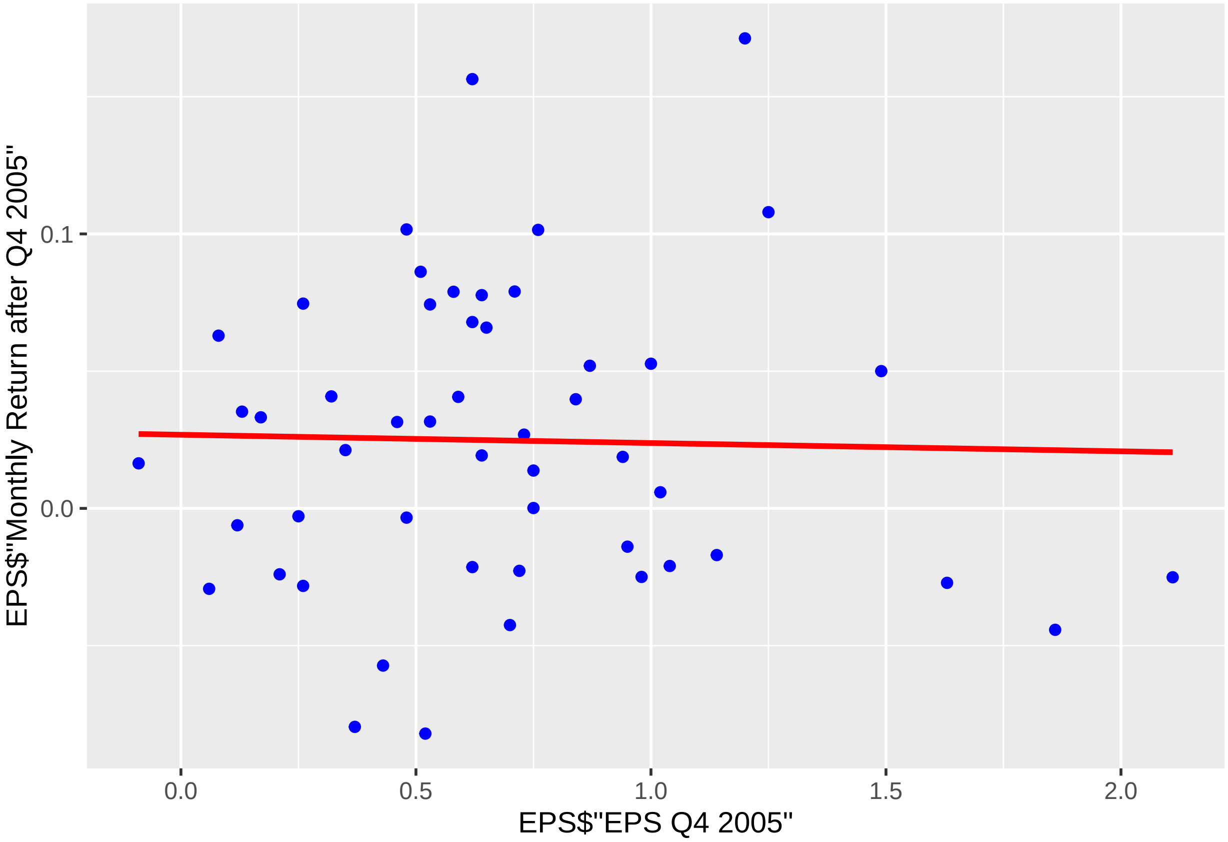
Part 5: Plot the original data and the linear regression line.

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

```
# assigning the dataset
data(EPS)

# deciding on X and Y axis + plotting the points on graph in blue
p1 = ggplot (data= EPS, aes(x=EPS$'EPS Q4 2005',y=EPS$'Monthly Return after Q4 2005')) + geom_point(col

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



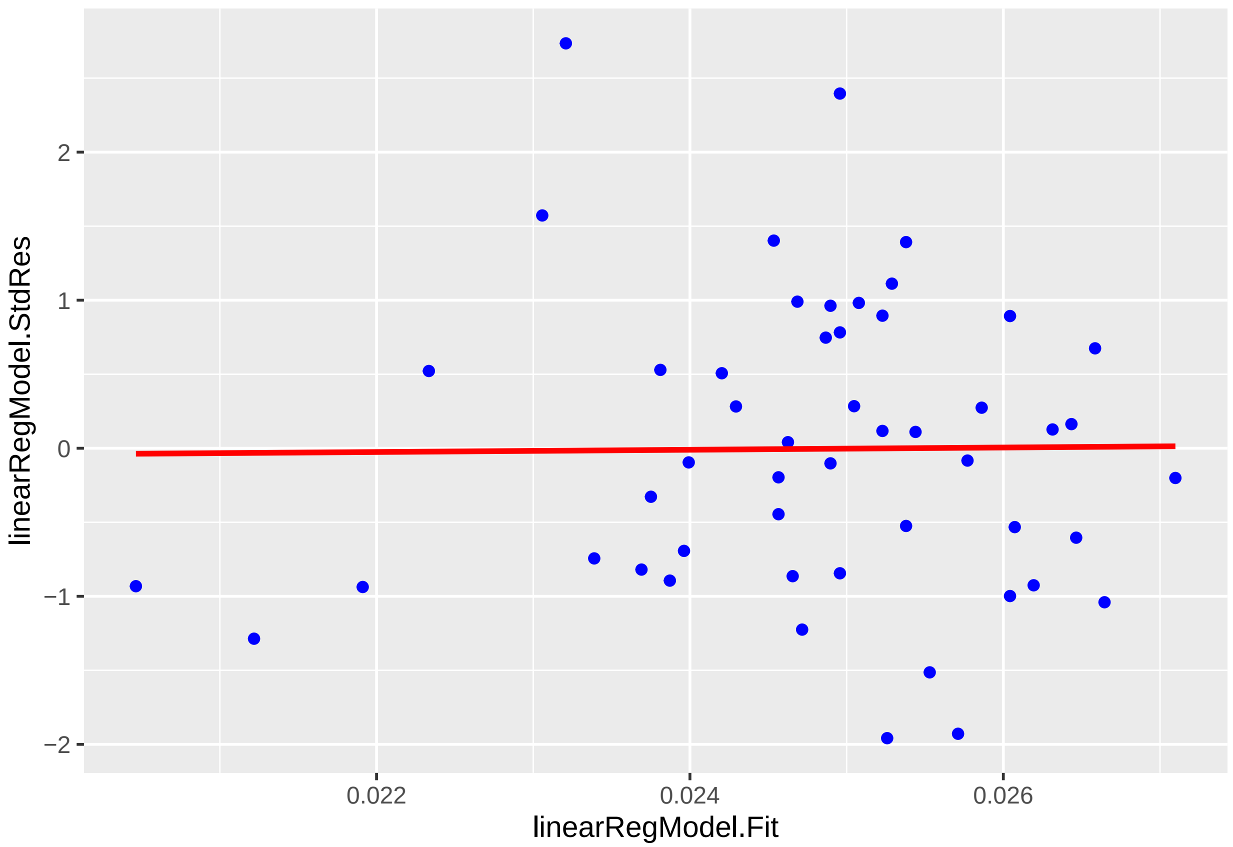
Part 6: Plot the standardized residuals vs. fitted values.

```
# 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=EPS,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")
```


2
1
0
−1
−2
0.022
0.024
0.026
linearRegModel.StdRes
linearRegModel.Fit