SIMULATION FUNCTION FOR LINEAR REGRESSION

SIMULATE_LINEAR_REGRESSION(N_ROW, N_VAR)

Objective

- A function to simulate Linear Regression from user_input of N_row(no of rows) and N_var(no of columns)

What will you get from just one function:-

- Summary of the Data
- Univariate plots for each variable no matter what value of n_var is.
- Correlation plot
- Correlation summary
- Linear Regression Model Summary

Defining a function with name Simulate_linear_regression

Default Values for rows and columns

```r
simulate_linear_regression <- function(N_row =10,N_var=2) {

    df <- as.data.frame(matrix(rnorm(N_row*N_var) , ncol = N_var, nrow = N_row))
    print("---------------DataFrame Created-----------------")
    print(df)
```

Creating a variable df which will store all the values with user defined N_row and N_var

This will print summary of the data

This loop will create hist and box plot for each variable be it 2 or 200 or N.

```
print("--------------Summary of the Data--------------")
print(summary(df))
print("--------------Graphs - check the plot window----")
for (i in (1:ncol(df))){
  par(mfrow=c(2,1))
  boxplot(df[,i], main = paste("Boxplot of", names(df)[i]),
          ylab = names(df)[i], col = "maroon", border = "grey5",
          horizontal = T)
  hist(df[,i], main = paste("Histogram of", names(df)[i]),
       xlab = names(df)[i], ylab = "Values of the variable", col = "red", border=F)

}
```

This will calculate correlation among variables

```
print("----------------Correlation among the variables--------")
res <- cor(df)
print(round(res, 2))

#correlation plot
library(ggcorrplot)
ggcorrplot(cor(df), hc.order = TRUE, type = "lower",
           lab = TRUE)
```

This will Plot correlation graph

Linear Regression in
R using lm() Function

For iterating and collecting each variable in z in order to comply with the syntax of lm()

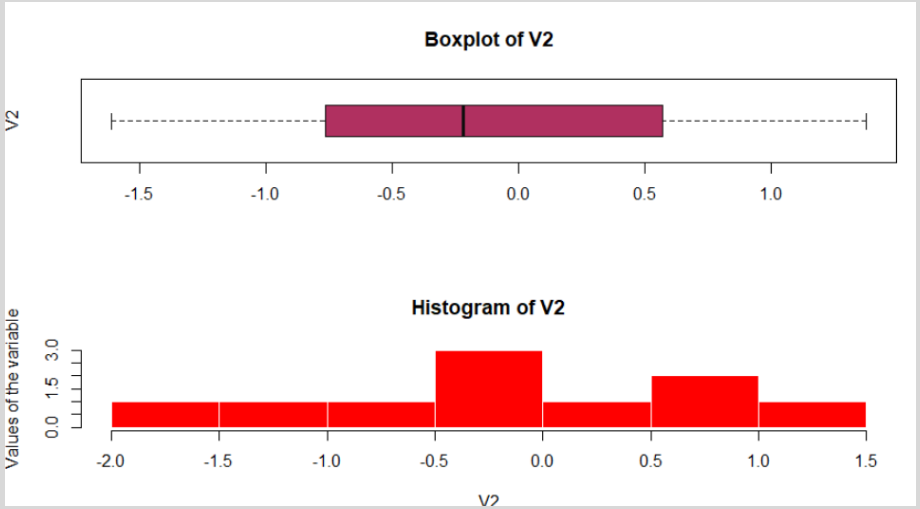
```
    # Simple/Multiple Linear Regression
    y <- names(df[ncol(df)])
    z = paste(y,paste('~',paste(colnames(df)[1:ncol(df)-1], collapse=" + ")))
    model <- lm(formula = z, df)
    print("---------------Linear Regression Summary---------")
    print(summary(model))
}
```

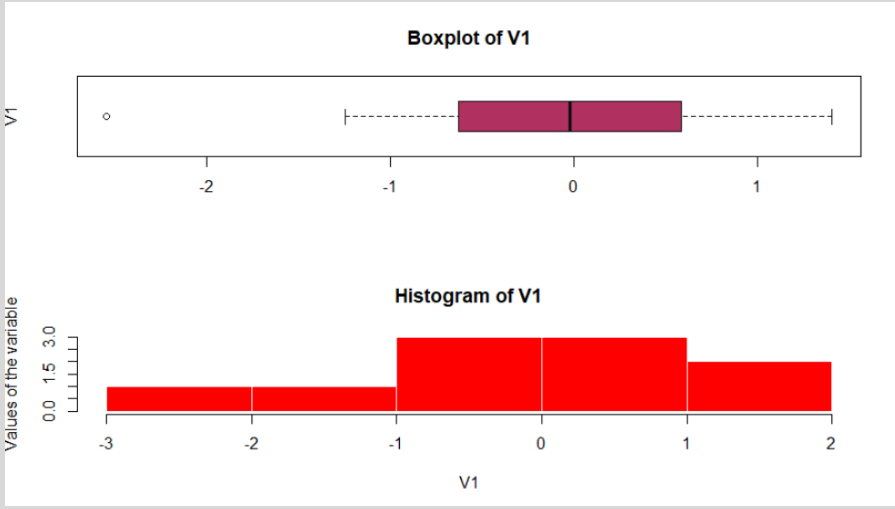
Summary of LR Model

OUTPUT

From the function
simulate_linear_regression(10,2)

- Box Plot and histogram of variables





```
> simulate_linear_regression(10,2)
[1] "--------------DataFrame Created----------------"
           V1          V2
1  -0.43915513 -0.76373320
2  -0.06569131 -1.43128057
3  -0.62894087 -0.41222760
4   0.32008068 -0.39885532
5  -2.54718233 -1.61187012
6  -1.24506272  0.56922869
7   1.23566027  1.37859271
8   1.40813431  0.81607323
9   0.02910190 -0.04022534
10  0.58457021  0.27021763
[1] "--------------Summary of the Data--------------"
       V1                  V2
 Min.   :-2.54718   Min.   :-1.6119
 1st Qu.:-0.58149   1st Qu.:-0.6759
 Median :-0.01829   Median :-0.2195
 Mean   :-0.13485   Mean   :-0.1624
 3rd Qu.: 0.51845   3rd Qu.: 0.4945
 Max.   : 1.40813   Max.   : 1.3786
[1] "--------------Graphs - check the plot window----"
[1] "---------------Correlation among the variables--------"
     V1   V2
V1 1.00 0.67
V2 0.67 1.00
[1] "---------------Linear Regression Summary---------"

Call:
lm(formula = z, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-1.30650 -0.36108  0.02599  0.11457  1.33565

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.08904    0.24125  -0.369   0.7216
V1           0.54405    0.21504   2.530   0.0353 *
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7574 on 8 degrees of freedom
Multiple R-squared:  0.4445,    Adjusted R-squared:  0.375
F-statistic: 6.401 on 1 and 8 DF,  p-value: 0.03526
```

Dataframe created with 10 rows and 2 columns

Summary of the data created

Correlation of the dataframe

Summary of the regression Fit.