

MVP Architecture Code Smell data exploration

In []:

```
# # Installing required Libraries
install.packages("correlation")
install.packages("bayestestR")
install.packages("see")
install.packages("ggplot2")
install.packages("tidyverse")
install.packages("poorman")
```

In []:

```
#Loading the installed libraries
library(readxl)          # reading in data
library(ggplot2)          # visualizing data
library(gridExtra)        # combining multiple plots
library(corrgram)         # visualizing data
library(corrplot)         # visualizing data
library(Hmisc)            # produces correlation matrices with p-values
library(ppcor)            # assesses partial correlations
```

In [20]:

```
#reading MVP input file
file_mvp <- read.csv(file = "MVP_input.csv", header=TRUE, stringsAsFactors = F)
```

In [21]:

```
#printing the top rows of file
head(file_mvp)
```

ID	Magic.Number	Long.Statement	Missing.Defaults	Complex.Conditional	Unutilized.abstraction	Cyclic.Dependency	Insufficient.Modularization	Comments
1	8	4	0	4	17	0		C
2	102	47	4	4	43	23		S
3	1	11	0	0	25	0		C
4	79	42	7	1	11	37		A
5	13	6	0	4	14	0		C
6	79	42	7	1	11	39		S

In [22]:

```
str(file_mvp) # shows the structure of the data frame
summary(file_mvp) #provides summary statistics on the columns of the data frame
```

```
'data.frame': 90 obs. of 11 variables:
 $ ID                  : int 1 2 3 4 5 6 7 8 9 10 ...
 $ Magic.Number         : int 8 102 1 79 13 79 234 16 432 10 ...
 $ Long.Statement       : int 4 47 11 42 6 42 56 77 155 40 ...
 $ Missing.Defaults     : int 0 4 0 7 0 7 4 0 13 1 ...
 $ Complex.Conditional   : int 4 4 0 1 4 1 6 1 31 0 ...
 $ Unutilized.abstraction : int 17 43 25 11 14 11 252 109 113 32 ...
 $ Cyclic.Dependency     : int 0 23 0 37 0 39 4 19 92 4 ...
 $ Insufficient.Modularization: int 0 5 0 4 0 5 12 1 25 0 ...
 $ Deficient.Encapsulation : int 5 11 5 7 4 7 198 53 36 10 ...
 $ Total.Code.Smells      : int 22 59 30 22 18 23 462 163 174 42 ...
 $ Total.Architectural.Smells: int 16 157 12 129 23 129 300 94 631 51 ...

ID          Magic.Number    Long.Statement  Missing.Defaults
Min. : 1.00  Min. : 0.00  Min. : 4.00  Min. : 0.000
1st Qu.:23.25 1st Qu.: 31.0  1st Qu.: 29.50 1st Qu.: 1.000
Median :45.50  Median : 81.0  Median : 46.00  Median : 4.000
Mean   :45.50  Mean   : 176.8  Mean   : 71.66  Mean   : 5.789
3rd Qu.:67.75 3rd Qu.: 190.2 3rd Qu.: 85.25 3rd Qu.: 7.000
Max.   :90.00  Max.   :1271.0  Max.   :480.00  Max.   :41.000

Complex.Conditional Unutilized.abstraction Cyclic.Dependency
Min. : 0.00  Min. : 7.00  Min. : 0.00
1st Qu.: 0.25 1st Qu.: 23.25 1st Qu.: 2.00
Median : 3.50  Median : 42.50  Median :10.00
Mean   : 10.94  Mean   : 64.73  Mean   :16.66
3rd Qu.: 9.00  3rd Qu.: 77.50 3rd Qu.:28.50
Max.   :199.00  Max.   :436.00  Max.   :94.00

Insufficient.Modularization Deficient.Encapsulation Total.Code.Smells
Min. : 0.000  Min. : 1.00  Min. : 18.00
1st Qu.: 2.000 1st Qu.: 7.00  1st Qu.: 68.25
Median : 5.000  Median : 14.00  Median : 118.50
Mean   : 8.711  Mean   : 27.61  Mean   : 245.74
3rd Qu.:10.000 3rd Qu.: 30.50 3rd Qu.: 285.25
Max.   :55.000  Max.   :276.00  Max.   :1655.00

Total.Architectural.Smells
Min. : 2.00
1st Qu.: 13.75
Median : 31.00
Mean   : 92.52
3rd Qu.: 64.25
Max.   :1561.00
```

checking outliers with box plot

In [26]:

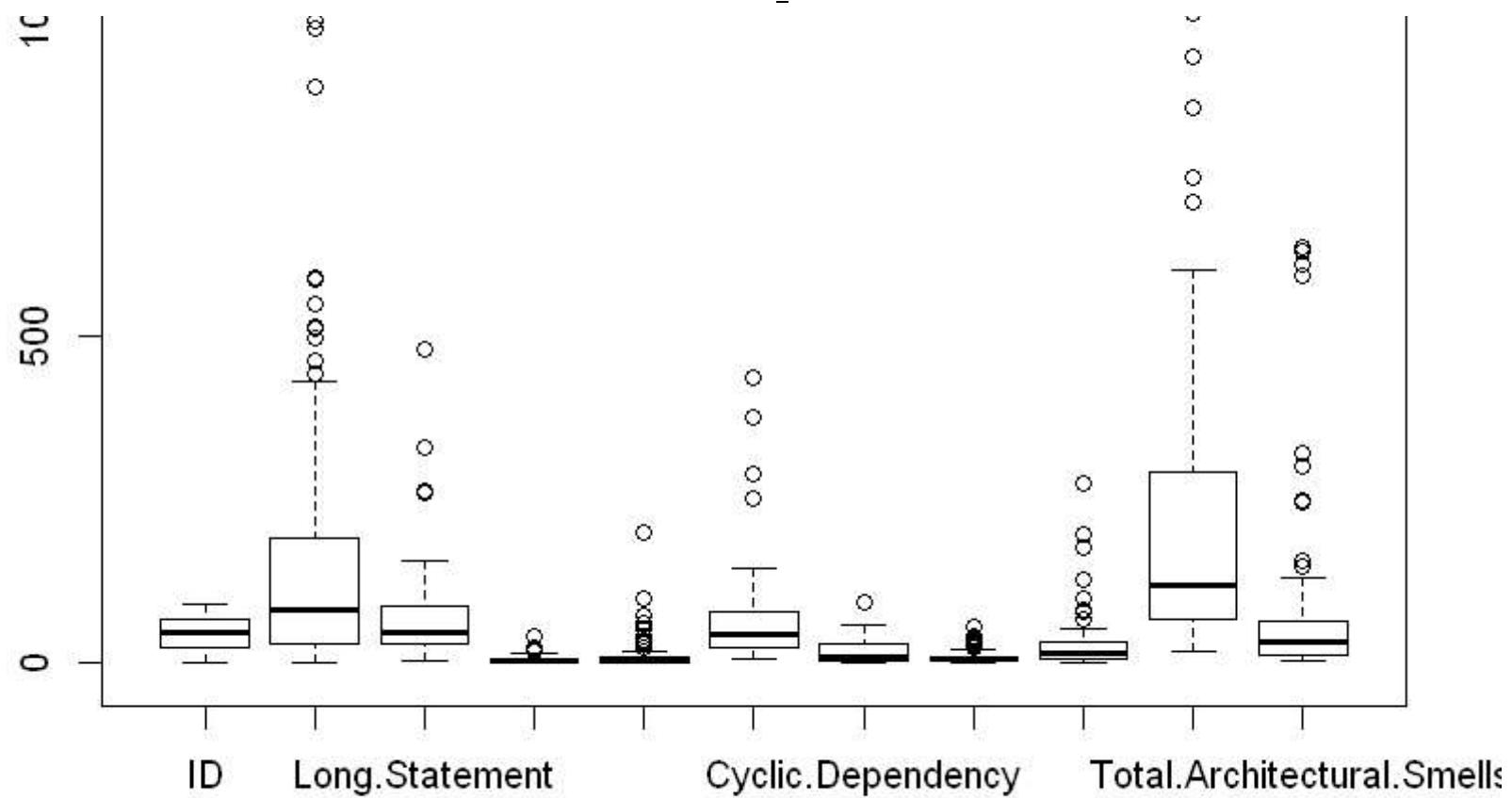
```
# Create a boxplot of the dataset, outliers are shown as two distinct points  
boxplot(file_mvp)$out
```

```
1.1271  
2.589  
3.496  
4.443  
5.971  
6.549  
7.586  
8.982  
9.461  
10.882  
11.514  
12.510  
13.259  
14.480  
15.263  
16.329  
17.20  
18.20  
19.17  
20.25  
21.41  
22.24  
23.31  
24.29  
25.72  
26.24  
27.36  
28.31
```

29.61
30.199
31.41
32.98
33.51
34.252
35.251
36.436
37.375
38.290
39.92
40.94
41.25
42.38
43.55
44.42
45.28
46.25
47.26
48.36
49.41
50.31
51.198
52.177
53.79
54.81
55.98
56.68
57.276
58.128
59.1655
60.705
61.848

62.743
63.1274
64.1183
65.1198
66.928
67.993
68.157
69.300
70.631
71.147
72.250
73.320
74.1561
75.245
76.636
77.592
78.609





There are outliers present in all the columns.

```
In [ ]: #making a copy of original file before changing the column name
df_mvp <- file_mvp
```

Renaming the column name for our ease in furthur calculations

```
In [27]: #LOAD LIBRARY library(tidyverse)
#rename the column name
names(file_mvp)[names(file_mvp) == "Magic.Number"] <- "MN"
names(file_mvp)[names(file_mvp) == "Long.Statement"] <- "LS"
names(file_mvp)[names(file_mvp) == "Missing.Defaults"] <- "MD"
```

```
names(file_mvp)[names(file_mvp) == "Complex.Conditional"] <- "CC"
names(file_mvp)[names(file_mvp) == "Unutilized.abstraction"] <- "Ua"
names(file_mvp)[names(file_mvp) == "Cyclic.Dependency"] <- "CD"
names(file_mvp)[names(file_mvp) == "Insufficient.Modularization"] <- "IM"
names(file_mvp)[names(file_mvp) == "Deficient.Encapsulation"] <- "DE"
names(file_mvp)[names(file_mvp) == "Total.Architectural.Smells"] <- "TDS"
names(file_mvp)[names(file_mvp) == "Total.Code.Smells"] <- "TCS"
```

In [28]:

```
#column names
colnames(file_mvp)
```

- 1.'ID'
- 2.'MN'
- 3.'LS'
- 4.'MD'
- 5.'CC'
- 6.'Ua'
- 7.'CD'
- 8.'IM'
- 9.'DE'
- 10.'TCS'
- 11.'TDS'

In [29]:

```
#creating a copy of dataframe
df <- file_mvp
```

We can see there are lot of outliers present in our dataset.

We are targeting to remove outliers from some columns for better correlation analysis.

In [31]:

```
# finding outliers in our dataset - Long.Statement
Q1 <- quantile(file_mvp$MN, probs=c(.25, .75), na.rm = FALSE)
Q2 <- quantile(file_mvp$LS, probs=c(.25, .75), na.rm = FALSE)
Q3 <- quantile(file_mvp$MD, probs=c(.25, .75), na.rm = FALSE)
Q4 <- quantile(file_mvp$CC, probs=c(.25, .75), na.rm = FALSE)
Q5 <- quantile(file_mvp$Ua, probs=c(.25, .75), na.rm = FALSE)
```

```

Q6 <- quantile(file_mvp$CD, probs=c(.25, .75), na.rm = FALSE)
Q7 <- quantile(file_mvp$IM, probs=c(.25, .75), na.rm = FALSE)
Q8 <- quantile(file_mvp$DE, probs=c(.25, .75), na.rm = FALSE)
Q9 <- quantile(file_mvp$TDS, probs=c(.25, .75), na.rm = FALSE)
Q10 <- quantile(file_mvp$TCS, probs=c(.25, .75), na.rm = FALSE)

# how to find outliers in r - calculate Interquartile Range
iqr1 <- IQR(file_mvp$MN)
iqr2 <- IQR(file_mvp$LS)
iqr3 <- IQR(file_mvp$MD)
iqr4 <- IQR(file_mvp$CC)
iqr5 <- IQR(file_mvp$Ua)
iqr6 <- IQR(file_mvp$CD)
iqr7 <- IQR(file_mvp$IM)
iqr8 <- IQR(file_mvp$DE)
iqr9 <- IQR(file_mvp$TDS)
iqr10 <- IQR(file_mvp$TCS)

# how to find outliers in r - upper and Lower range
#up <- Q[2]+1.5*iqr # Upper Range
#Low<- Q[1]-1.5*iqr # Lower Range

# how to remove outliers in (the removal)
file_mvp<- subset(file_mvp, file_mvp$MN > (Q1[1] - 1.5*iqr1) & file_mvp$MN <(Q1[2]+1.5*iqr1))
file_mvp<- subset(file_mvp, file_mvp$LS > (Q2[1] - 1.5*iqr2) & file_mvp$LS< (Q2[2]+1.5*iqr2))
file_mvp<- subset(file_mvp, file_mvp$MD > (Q3[1] - 1.5*iqr3) & file_mvp$MD< (Q3[2]+1.5*iqr3))
file_mvp<- subset(file_mvp, file_mvp$CC > (Q4[1] - 1.5*iqr4) & file_mvp$CC< (Q4[2]+1.5*iqr4))
file_mvp<- subset(file_mvp, file_mvp$Ua > (Q5[1] - 1.5*iqr5) & file_mvp$Ua< (Q5[2]+1.5*iqr5))
file_mvp<- subset(file_mvp, file_mvp$CD > (Q6[1] - 1.5*iqr6) & file_mvp$CD< (Q6[2]+1.5*iqr6))
file_mvp<- subset(file_mvp, file_mvp$IM > (Q7[1] - 1.5*iqr7) & file_mvp$IM< (Q7[2]+1.5*iqr7))
file_mvp<- subset(file_mvp, file_mvp$DE > (Q8[1] - 1.5*iqr8) & file_mvp$DE< (Q8[2]+1.5*iqr8))
file_mvp<- subset(file_mvp, file_mvp$TDS >(Q9[1] - 1.5*iqr9) & file_mvp$TDS<(Q9[2]+1.5*iqr9))
file_mvp<- subset(file_mvp, file_mvp$TCS >(Q10[1] - 1.5*iqr10) & file_mvp$TCS<(Q10[2]+1.5*iqr10))

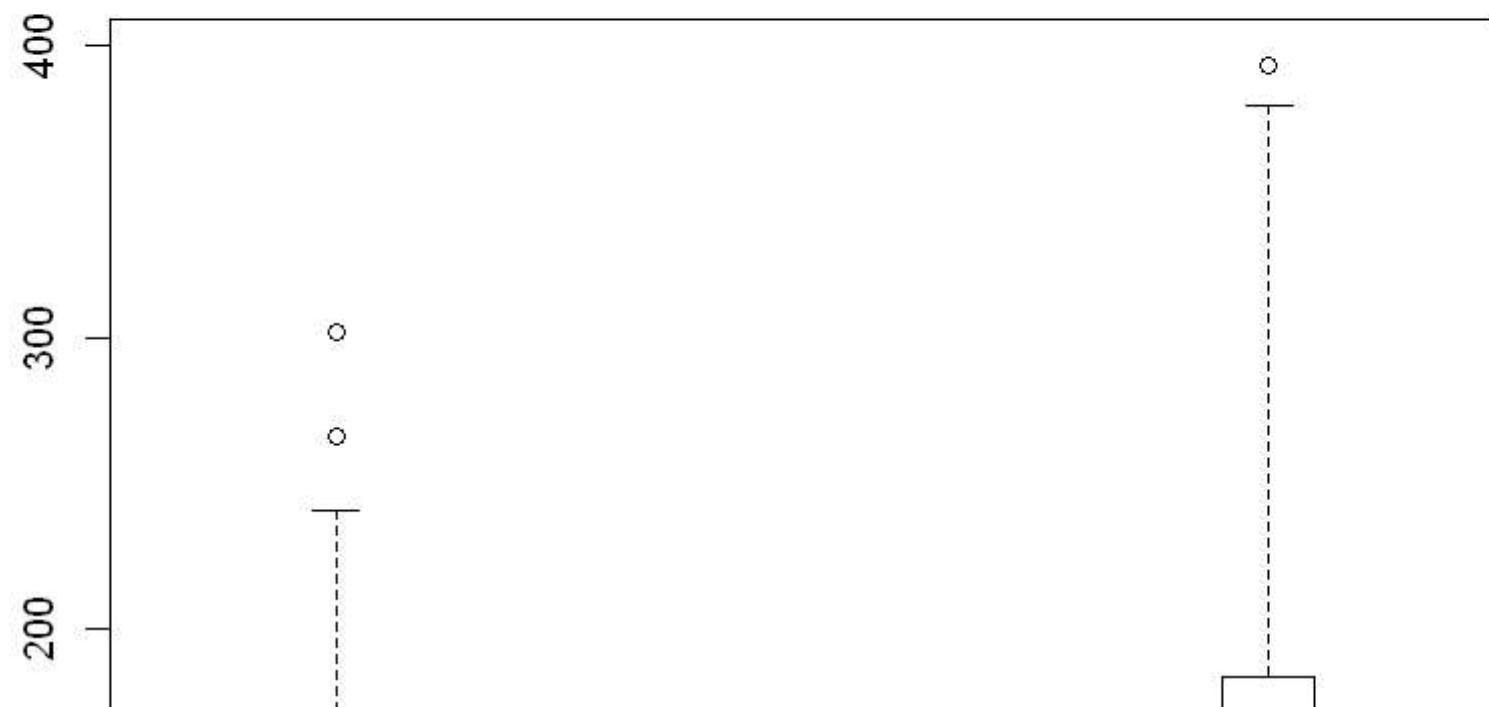
```

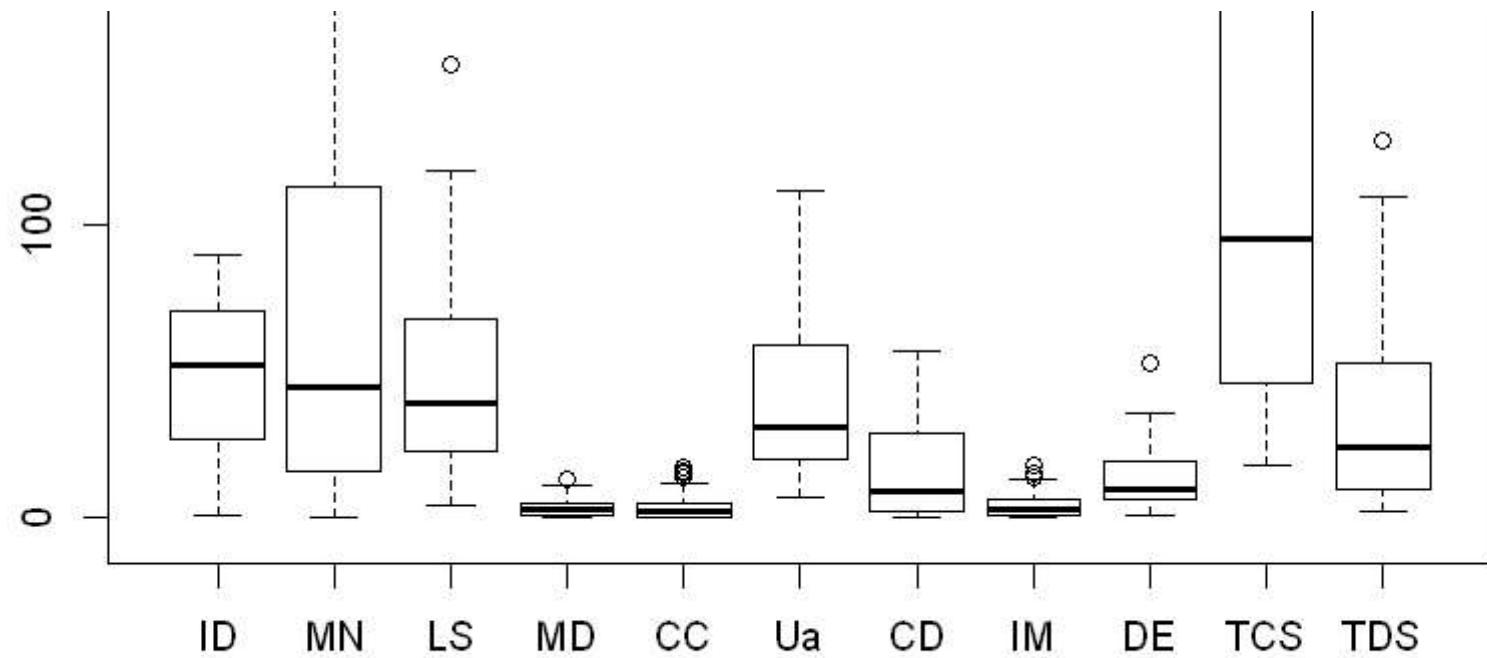
In [32]:

```
# Create a boxplot of the dataset, outliers are shown as two distinct points
boxplot(file_mvp)$out
```

1. 302
2. 266
3. 155
4. 13

5.14
6.16
7.15
8.17
9.18
10.18
11.14
12.15
13.53
14.393
15.129
16.129
17.129





displaying the column names

renaming the column name for our ease

In [34]:

```
# Correlation test between multiple variables
round(cor(file_mvp, use = "pairwise.complete.obs"), 2)
```

	ID	MN	LS	MD	CC	Ua	CD	IM	DE	TCS	TDS
ID	1.00	0.16	0.04	0.24	0.22	0.01	-0.21	0.17	-0.05	0.26	-0.56
MN	0.16	1.00	0.49	0.24	0.67	0.15	0.03	0.57	0.28	0.89	-0.09
LS	0.04	0.49	1.00	0.13	0.56	0.39	0.32	0.46	0.28	0.73	0.22
MD	0.24	0.24	0.13	1.00	0.28	-0.08	0.13	0.34	0.28	0.19	0.16
CC	0.22	0.67	0.56	0.28	1.00	0.30	0.17	0.58	0.28	0.76	-0.06

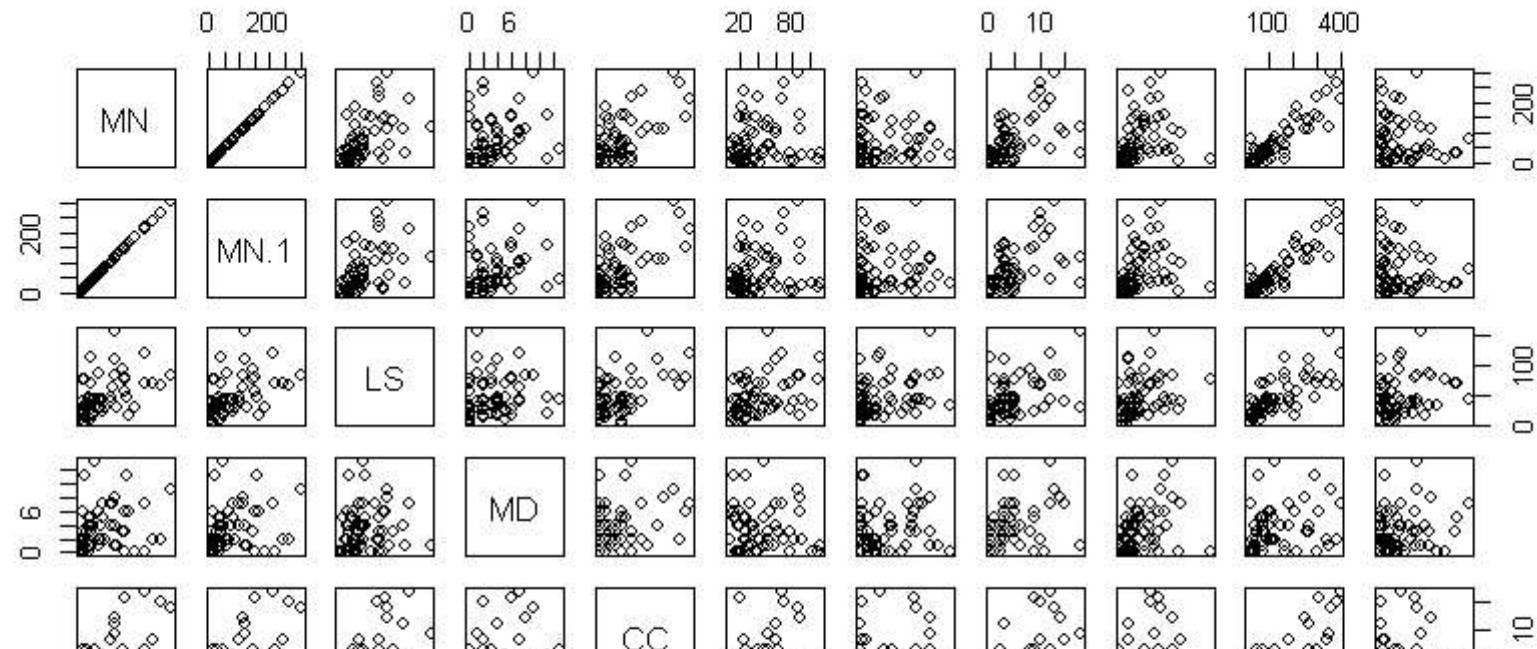
	ID	MN	LS	MD	CC	Ua	CD	IM	DE	TCS	TDS
Ua	0.01	0.15	0.39	-0.08	0.30	1.00	0.24	0.18	0.30	0.32	0.15
CD	-0.21	0.03	0.32	0.13	0.17	0.24	1.00	0.20	0.09	0.06	0.74
IM	0.17	0.57	0.46	0.34	0.58	0.18	0.20	1.00	0.32	0.61	0.05
DE	-0.05	0.28	0.28	0.28	0.28	0.30	0.09	0.32	1.00	0.38	0.12
TCS	0.26	0.89	0.73	0.19	0.76	0.32	0.06	0.61	0.38	1.00	-0.17
TDS	-0.56	-0.09	0.22	0.16	-0.06	0.15	0.74	0.05	0.12	-0.17	1.00

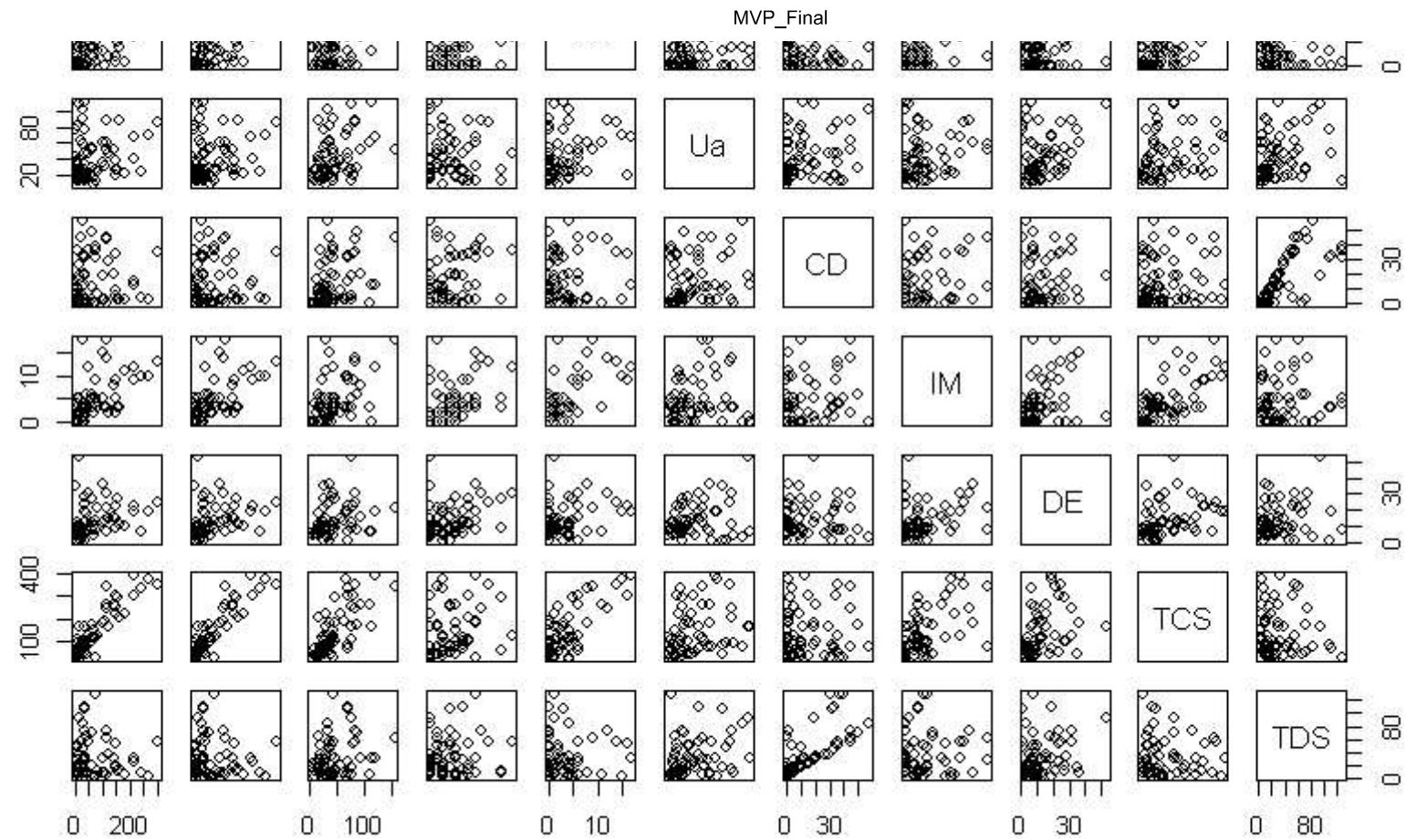
We are plotting three correlation graphs to find our relationship existance between each column present in our dataset.

- 1. Scatter plot matrices to check relationship within each columns of dataset

In [36]:

```
#Scatter plot matrices to check relationship within each columns of dataset
pairs(file_mvp[, c(2, 2:11)])
```

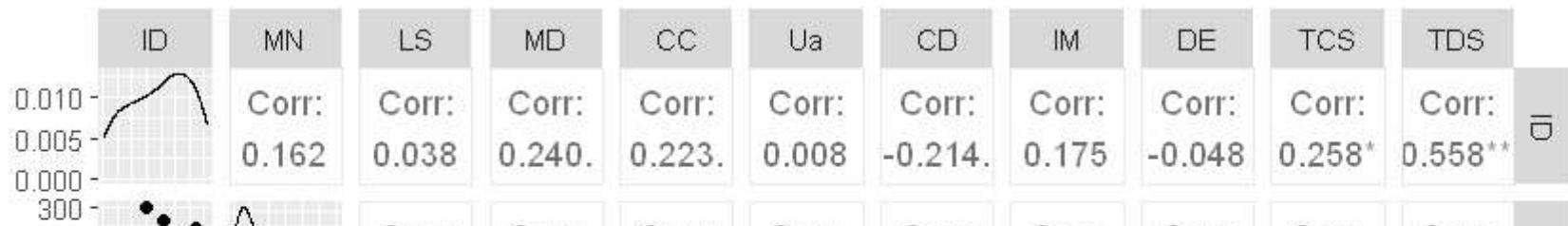


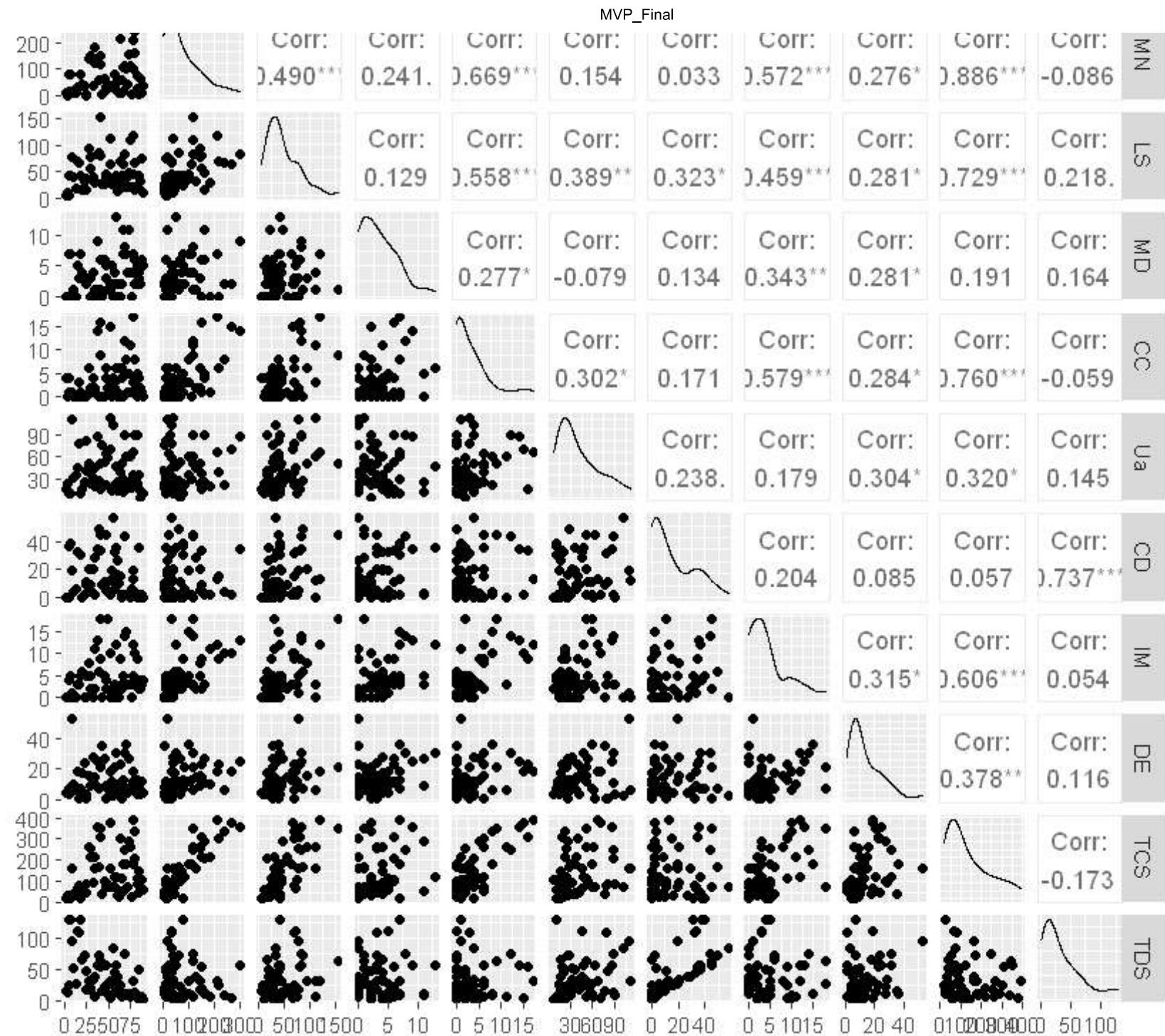


- 1. pairs plot to check correlation pattern between multiple variables

In [40]:

```
#create pairs plot
library(GGally)
ggpairs(file_mvp)
```

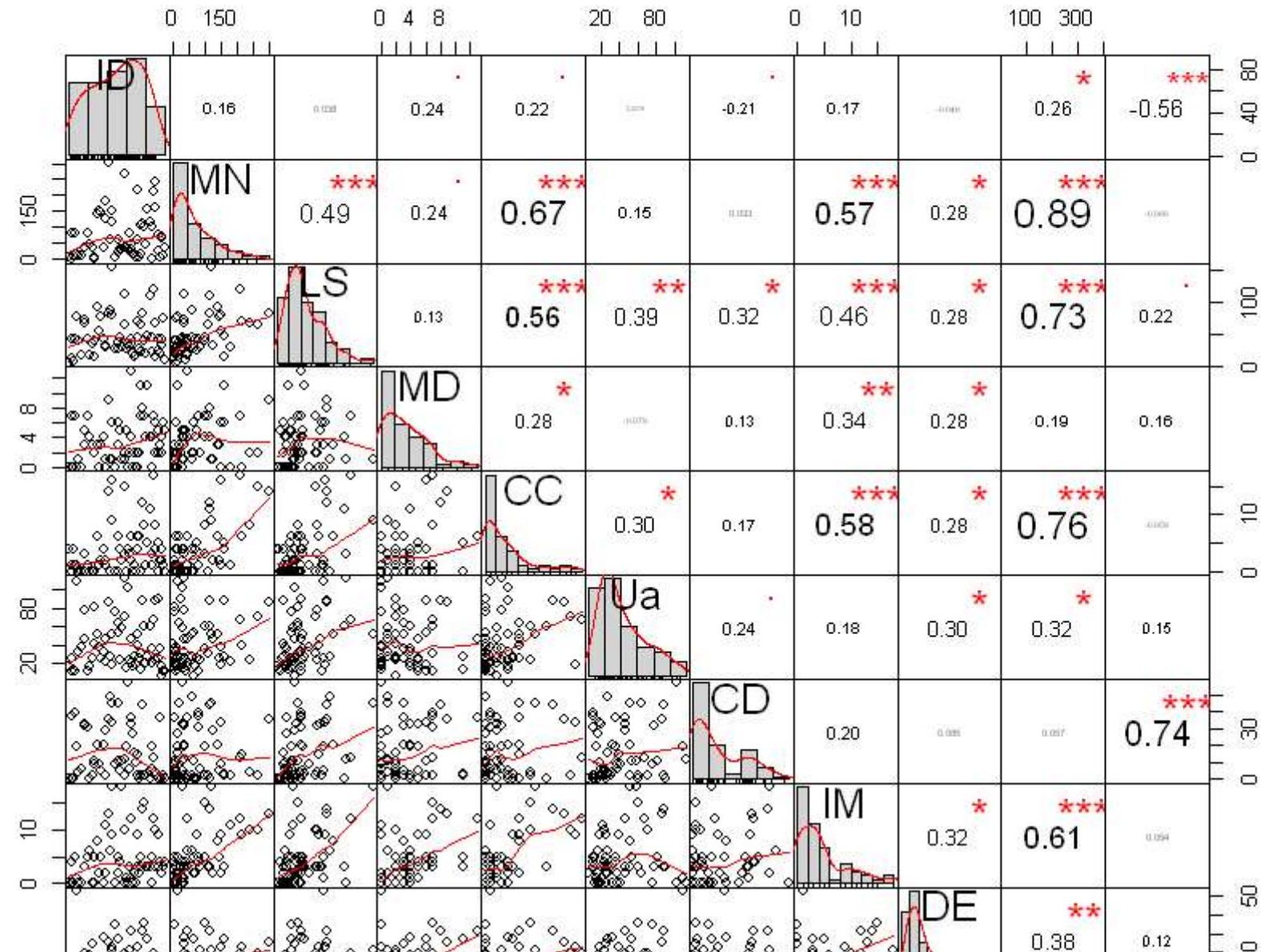


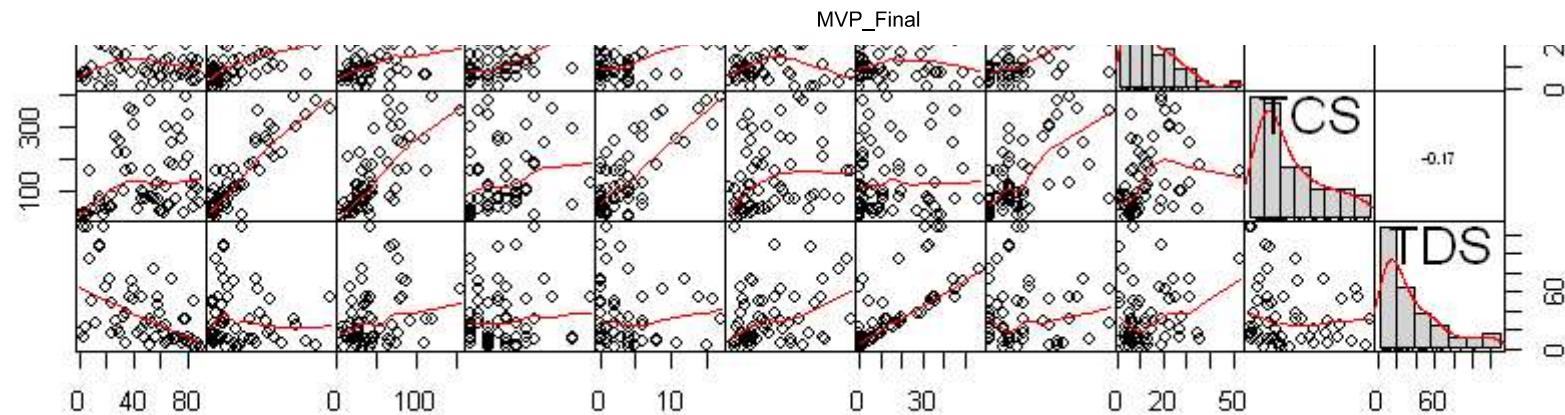


- 1. displaying PerformanceAnalytics chart of correlation matrix between variables

In [43]:

```
#install.packages("PerformanceAnalytics")
library(PerformanceAnalytics)
chart.Correlation(file_mvp, histogram=TRUE, pch=19)
```





In the above plot:

- The distribution of each variable is shown on the diagonal.
- On the bottom of the diagonal : the bivariate scatter plots with a fitted line are displayed
- On the top of the diagonal : the value of the correlation plus the significance level as stars
- Each significance level is associated to a symbol : p-values(0, 0.001, 0.01, 0.05, 0.1, 1) <=> symbols("****", "***", "**", "*", ".", " ")

From the above two plot we can see few positive correlation , which are :

- CC and MN
- CC and LS
- IM and MN
- CC and IM
- TDS and TCS are showing high correlation as its a summation column values for the rest.

Hence we will furtur do the pearson, kendall, spearman on these columns.

- Pearson correlation evaluates the linear relationship between two continuous variables.
- Kendall and spearman correlation evaluates the monotonic relationship between variables.

In [76]:

```
#pearson1 - CC and MN
p1 <- cor(file_mvp$CC, file_mvp$MN, method = "pearson")
p1
cor.test(file_mvp$CC, file_mvp$MN, use = 'complete.obs')
```

```
0.668660178668618
```

Pearson's product-moment correlation

```
data: file_mvp$CC and file_mvp$MN
t = 6.9656, df = 60, p-value = 2.863e-09
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.5028776 0.7869927
sample estimates:
cor
0.6686602
```

Observation: Our p-value is significant and the 95% confidence interval confirms this as the range does not contain zero.

This suggests the correlation between CC(Complex.Conditional) and MN(Magic.Number) per architecture is $r = 0.67$ with 95% confidence of being between 0.50 and 0.77.

In [45]:

```
#pearson2 - CC and LS
p2 <- cor(file_mvp$CC, file_mvp$LS, method = "pearson")
p2
cor.test(file_mvp$CC, file_mvp$LS, use = 'complete.obs')
```

```
0.557872976071721
```

Pearson's product-moment correlation

```
data: file_mvp$CC and file_mvp$LS
t = 5.2068, df = 60, p-value = 2.469e-06
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.3579862 0.7088682
sample estimates:
cor
0.557873
```

Observation: Our p-value is significant and the 95% confidence interval confirms this as the range does not contain zero.

This suggests the correlation between CC(Complex.Conditional) and LS(long.Statement) per architecture is $r = 0.56$ with 95% confidence of being between 0.36 and 0.71.

In [46]:

```
#pearson3 - IM and MN
p3 <- cor(file_mvp$IM, file_mvp$MN, method = "pearson")
p3
cor.test(file_mvp$IM, file_mvp$MN, use = 'complete.obs')
```

```
0.572212614096902
Pearson's product-moment correlation
```

```
data: file_mvp$IM and file_mvp$MN
t = 5.4046, df = 60, p-value = 1.181e-06
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.3762130 0.7191933
sample estimates:
cor
0.5722126
```

Observation: Our p-value is significant and the 95% confidence interval confirms this as the range does not contain zero.

This suggests the correlation between IM(Insufficient.Modularization) and MN(Magic.Number) per architecture is $r = 0.57$ with 95% confidence of being between 0.38 and 0.72.

```
In [60]: #pearson4 - CC and IM
p4 <- cor(file_mvp$CC, file_mvp$IM, method = "pearson")
p4
cor.test(file_mvp$CC, file_mvp$IM, use = 'complete.obs')
```

```
0.57905607588186
Pearson's product-moment correlation
```

```
data: file_mvp$CC and file_mvp$IM
t = 5.5016, df = 60, p-value = 8.204e-07
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.3849653 0.7240981
sample estimates:
cor
0.5790561
```

Observation: Our p-value is significant and the 95% confidence interval confirms this as the range does not contain zero.

This suggests the correlation between CC(Complex.Conditional) and IM(Insufficient.Modularization) per architecture is $r = 0.58$ with 95% confidence of being between 0.38 and 0.72.

```
In [48]: #spearman1 - CC and MN
s1 <- cor(file_mvp$CC, file_mvp$MN, method = "spearman")
```

```
s1  
cor.test(file_mvp$CC, file_mvp$MN, use = 'complete.obs', method = "spearman")
```

0.543974326982902

Warning message in cor.test.default(file_mvp\$CC, file_mvp\$MN, use = "complete.obs", :
"Cannot compute exact p-value with ties"

Spearman's rank correlation rho

data: file_mvp\$CC and file_mvp\$MN
S = 18109, p-value = 4.887e-06
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
0.5439743

Observation: Our p-value is significant .

This suggests the Spearman rank correlation rho between CC(Complex.Conditional) and MN(Magic.Number) per architecture is r = 0.54 with P-value less than 0.05 we can reject the null hypothesis and say there is some relatiopn between two variables.

In [49]:

```
#spearman2 - CC and LS  
s2 <- cor(file_mvp$CC, file_mvp$LS, method = "spearman")  
s2  
cor.test(file_mvp$CC, file_mvp$LS, use = 'complete.obs', method = "spearman")
```

0.457559758924576

Warning message in cor.test.default(file_mvp\$CC, file_mvp\$LS, use = "complete.obs", :
"Cannot compute exact p-value with ties"

Spearman's rank correlation rho

data: file_mvp\$CC and file_mvp\$LS
S = 21541, p-value = 0.0001848
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
0.4575598

Observation: Our p-value is significant .

This suggests the Spearman rank correlation rho between CC(Complex.Conditional) and LS(long.Statement) per architecture is r = 0.46 with P-value less than 0.05 we can reject the null hypothesis and say there is some relatiopn between two variables.

```
In [50]: #spearman3 - IM and MN
s3 <- cor(file_mvp$IM, file_mvp$MN, method = "spearman")
s3
cor.test(file_mvp$IM, file_mvp$MN, use = 'complete.obs', method = "spearman")
```

0.645678822432376

Warning message in cor.test.default(file_mvp\$IM, file_mvp\$MN, use = "complete.obs", :
 "Cannot compute exact p-value with ties"

Spearman's rank correlation rho

data: file_mvp\$IM and file_mvp\$MN
 S = 14070, p-value = 1.459e-08
 alternative hypothesis: true rho is not equal to 0
 sample estimates:
 rho
 0.6456788

Observation: Our p-value is significant .

This suggests the Spearman rank correlation rho between IM(Insufficient.Modularization) and MN(Magic.Number) per architecture is r = 0.64 with P-value less than 0.05 we can reject the null hypothesis and say there is some relatiopn between two variables.

```
In [59]: #spearman4 - CC and IM
s4 <- cor(file_mvp$CC, file_mvp$IM, method = "spearman")
s4
cor.test(file_mvp$CC, file_mvp$IM, use = 'complete.obs', method = "spearman")
```

0.429813647019742

Warning message in cor.test.default(file_mvp\$CC, file_mvp\$IM, use = "complete.obs", :
 "Cannot compute exact p-value with ties"

Spearman's rank correlation rho

data: file_mvp\$CC and file_mvp\$IM
 S = 22643, p-value = 0.0004896
 alternative hypothesis: true rho is not equal to 0
 sample estimates:
 rho
 0.4298136

Observation: Our p-value is significant .

This suggests the Spearman rank correlation rho between CC(Complex.Conditional) and IM(Insufficient.Modularization) per architecture is $r = 0.43$ with P-value less than 0.05 we can reject the null hypothesis and say there is some relatiopn between two variables.

In [52]:

```
#kendall1 - CC and MN
k1 <- cor(file_mvp$CC, file_mvp$MN, method = "kendall")
k1
cor.test(file_mvp$CC, file_mvp$MN, use = 'complete.obs', method = "kendall")
```

0.406205078990181

Kendall's rank correlation tau

```
data: file_mvp$CC and file_mvp$MN
z = 4.3912, p-value = 1.127e-05
alternative hypothesis: true tau is not equal to 0
sample estimates:
tau
0.4062051
```

Observation: Our p-value is significant and the 95% confidence interval confirms this as the range does not contain zero.

This suggests the kendall rank correlation tau between MN(Magic.Number) and CC(Complex.Conditional) per architecture is $r = 0.41$ with P-vale less than 0.05 we can reject the null hypothesis and say there is some relatiopn between two variables.

In [53]:

```
#kendall2 -CC and LS
k2 <- cor(file_mvp$CC, file_mvp$LS, method = "kendall")
k2
cor.test(file_mvp$CC, file_mvp$LS, use = 'complete.obs', method = "kendall")
```

0.342471483949403

Kendall's rank correlation tau

```
data: file_mvp$CC and file_mvp$LS
z = 3.6937, p-value = 0.000221
alternative hypothesis: true tau is not equal to 0
sample estimates:
tau
0.3424715
```

Observation: Our p-value is significant and the 95% confidence interval confirms this as the range does not contain zero.

This suggests the kendall rank correlation tau between CC(Complex.Conditional) and LS(Long.Statement) per architecture is $r = 0.34$ with P-vale less than 0.05 we can reject the null hypothesis and say there is some relatiopn between two variables.

In [54]:

```
#kendall3 - IM and MN
k3 <- cor(file_mvp$IM, file_mvp$MN, method = "kendall")
k3
cor.test(file_mvp$IM, file_mvp$MN, use = 'complete.obs', method = "kendall")
```

0.485624212876279

Kendall's rank correlation tau

```
data: file_mvp$IM and file_mvp$MN
z = 5.3036, p-value = 1.136e-07
alternative hypothesis: true tau is not equal to 0
sample estimates:
tau
0.4856242
```

Observation: Our p-value is significant and the 95% confidence interval confirms this as the range does not contain zero.

This suggests the kendall rank correlation tau between IM(Insufficient.Modularization) and MN(Magic.Number) per architecture is $r = 0.47$ with P-vale less than 0.05 we can reject the null hypothesis and say there is some relatiopn between two variables.

In [58]:

```
#kendall4 - CC and IM
k4 <- cor(file_mvp$CC, file_mvp$IM, method = "kendall")
k4
cor.test(file_mvp$CC, file_mvp$IM, use = 'complete.obs', method = "kendall")
```

0.320622513038143

Kendall's rank correlation tau

```
data: file_mvp$CC and file_mvp$IM
z = 3.3255, p-value = 0.0008827
alternative hypothesis: true tau is not equal to 0
sample estimates:
tau
0.3206225
```

Observation: Our p-value is significant and the 95% confidence interval confirms this as the range does not contain zero.

This suggests the kendall rank correlation tau between CC(Complex.Conditional)and IM(Insufficient.Modularization) per architecture is $r = 0.32$ with P-value less than 0.05 we can reject the null hypothesis and say there is some relatiopn between two variables.

In [65]:

```
#printing all the correlation values
cat(p1,s1,k1, p2,s2,k2, p3,s3,k3, p4,s4,k4)
```

```
0.6686602 0.5439743 0.4062051 0.557873 0.4575598 0.3424715 0.5790561 0.4298136 0.3206225 0.5790561 0.4298136 0.3206225
```

Summarization of Correlation Processes

In [70]:

```
corr_df <- data.frame(
  Sl_No = c(1,2,3,4,5,6,7,8,9,10,11,12),
  Corr_Method = c("Pearson", "spearman", "kendall", "Pearson", "spearman", "kendall", "Pearson",
  Column_related = c("CC and MN", "CC and MN", "CC and MN", "CC and LS",
                      "CC and LS", "CC and LS", "IM and MN", "IM and MN", "IM and MN", "CC and IM", "CC and IM", "CC and IM"),
  corr_value = c(0.6686602, 0.5439743, 0.4062051, 0.557873, 0.4575598, 0.3424715, 0.5790561, 0.4298136, 0.3206225, 0.5790561,
  ))
```

In [71]:

```
corr_df
```

Sl_No	Corr_Method	Column_related	corr_value
1	Pearson	CC and MN	0.6686602
2	spearman	CC and MN	0.5439743
3	kendall	CC and MN	0.4062051
4	Pearson	CC and LS	0.5578730
5	spearman	CC and LS	0.4575598
6	kendall	CC and LS	0.3424715
7	Pearson	IM and MN	0.5790561
8	spearman	IM and MN	0.4298136
9	kendall	IM and MN	0.3206225
10	Pearson	CC and IM	0.5790561
11	spearman	CC and IM	0.4298136

SI_No	Corr_Method	Column_related	corr_value
12	kendall	CC and IM	0.3206225

```
12      kendall      CC and IM  0.3206225
```

```
In [72]: knitr::kable(corr_df, format = "markdown")
```

S1_No	Corr_Method	Column_related	corr_value
1	Pearson	CC and MN	0.6686602
2	spearman	CC and MN	0.5439743
3	kendall	CC and MN	0.4062051
4	Pearson	CC and LS	0.5578730
5	spearman	CC and LS	0.4575598
6	kendall	CC and LS	0.3424715
7	Pearson	IM and MN	0.5790561
8	spearman	IM and MN	0.4298136
9	kendall	IM and MN	0.3206225
10	Pearson	CC and IM	0.5790561
11	spearman	CC and IM	0.4298136
12	kendall	CC and IM	0.3206225

- From the above correlation tests we can conclude that there is high correlation present within MVP architecture between Complex.Conditional & Magic.Number, Complex.Conditional & Long.Statement, Insufficient.Modularization & Magic.Number and Complex.Conditional & Insufficient.Modularization.