Lab assesment - 18BCE1003

Siddharth.S.Chandran (18BCE1003)

05/04/2021

1. Reading the dataset

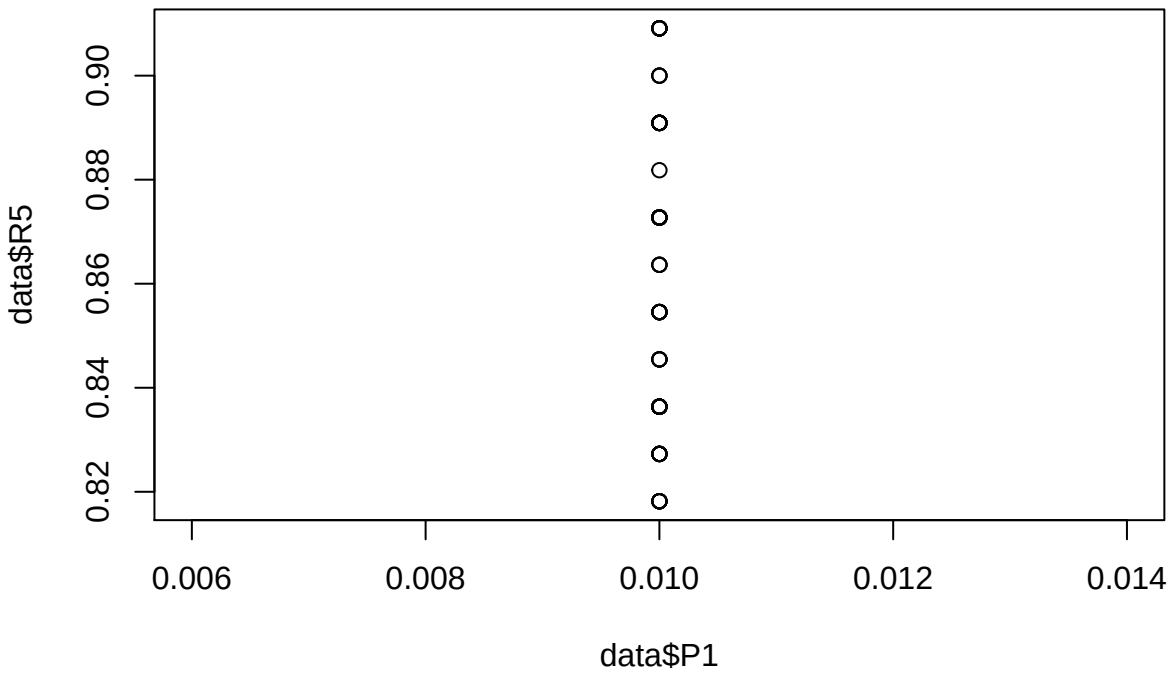
```
library(readxl)
data<-read_excel("C:/Users/Siddharth.S.Chandran/Desktop/Lab-Data-RegressionR5.xlsx")
str(data)
```

```
## tibble [80 x 18] (S3: tbl_df/tbl/data.frame)
##  $ Sample No: num [1:80] 1 2 3 4 5 6 7 8 9 10 ...
##  $ P1       : num [1:80] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
##  $ P2       : num [1:80] 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 ...
##  $ P3       : num [1:80] 0.873 0.873 0.873 0.873 0.873 ...
##  $ P4       : num [1:80] 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.0
##  $ P5       : num [1:80] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
##  $ P6       : num [1:80] 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 ...
##  $ P7       : num [1:80] 0.873 0.873 0.873 0.873 0.873 ...
##  $ P8       : num [1:80] 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.0
##  $ P9       : num [1:80] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
##  $ P10      : num [1:80] 0.0483 0.0483 0.0483 0.0483 0.0529 0.0529 0.0529 0.0529 0.0575 0.0575 ...
##  $ P11      : num [1:80] 0.873 0.873 0.873 0.873 0.935 ...
##  $ P12      : num [1:80] 0.000435 0.000435 0.000435 0.000435 0.000539 0.000539 0.000539 0.000539 0.0
##  $ P13      : num [1:80] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
##  $ P14      : num [1:80] 0.0483 0.0529 0.0575 0.0682 0.0483 0.0529 0.0575 0.0682 0.0483 0.0529 ...
##  $ P15      : num [1:80] 0.873 0.935 0.947 0.951 0.873 ...
##  $ P16      : num [1:80] 0.000435 0.000539 0.000694 0.000817 0.000435 0.000539 0.000694 0.000817 0.0
##  $ R5       : num [1:80] 0.873 0.836 0.891 0.891 0.836 ...
```

Visualizing the dataset

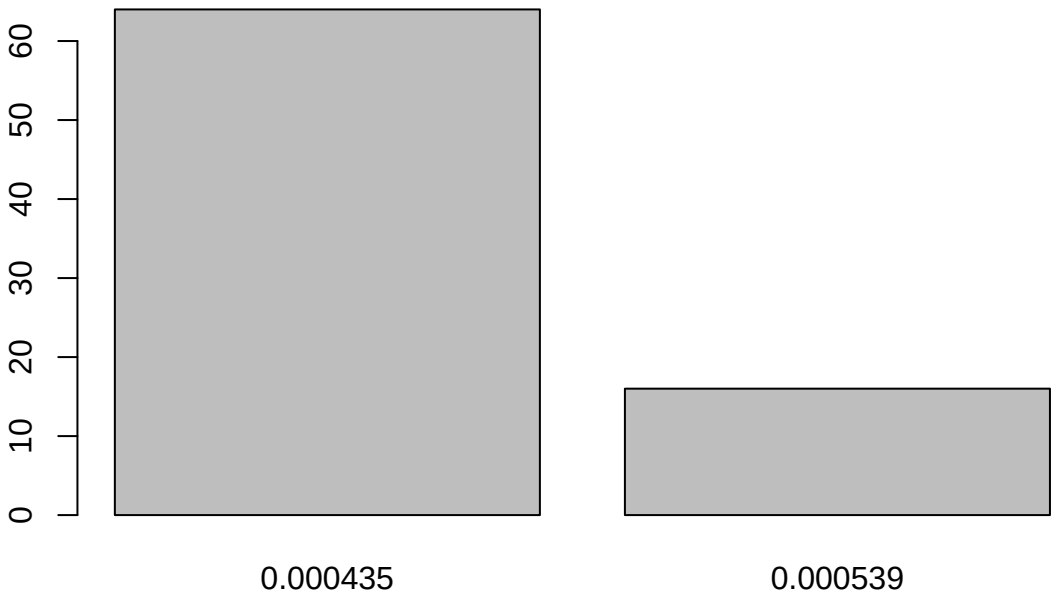
a. Scatter plot between P1and the r5

```
plot(data$P1, data$R5)
```



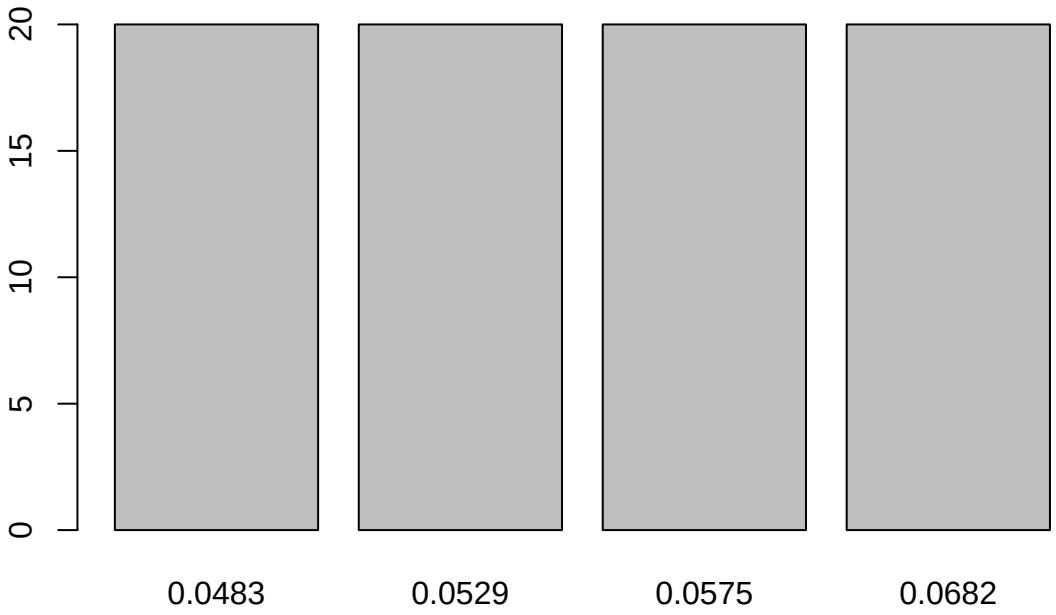
From this plot we can infer that there is no relation between the variables P1 and R5

```
barplot(table(data$P4))
```



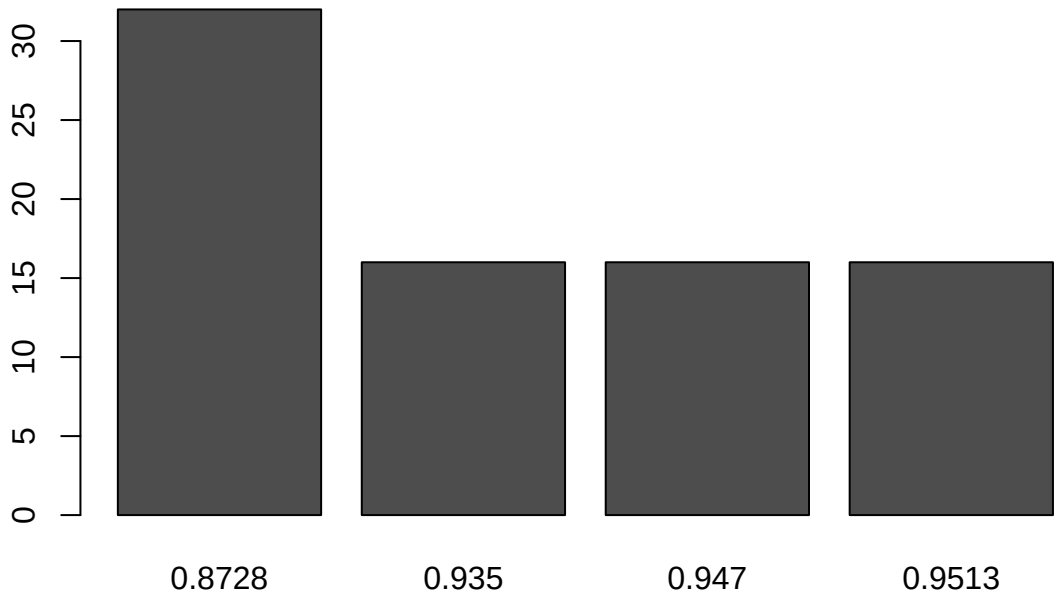
P4 is a factor variable with only 2 classes and the highest class is 0.000435

```
barplot(table(data$P10))
```



A bar chart is drawn for the frequency of each value in P10

```
barplot(table(data$P1, data$P7))
```



A bar chart is drawn between values of P1 and P7

2. Performing exploratory data analysis

```
str(data)
```

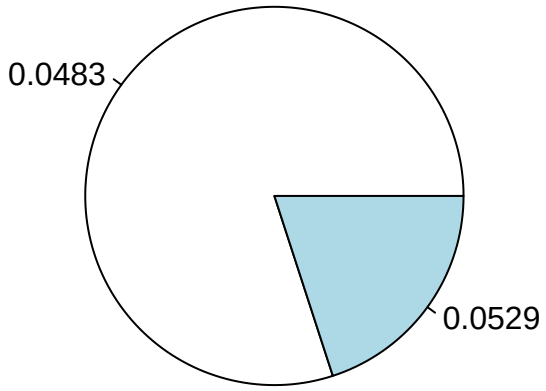
```
## tibble [80 x 18] (S3: tbl_df/tbl/data.frame)
##  $ Sample No: num [1:80] 1 2 3 4 5 6 7 8 9 10 ...
##  $ P1       : num [1:80] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
##  $ P2       : num [1:80] 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 ...
##  $ P3       : num [1:80] 0.873 0.873 0.873 0.873 0.873 ...
##  $ P4       : num [1:80] 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.0
##  $ P5       : num [1:80] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
##  $ P6       : num [1:80] 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 ...
##  $ P7       : num [1:80] 0.873 0.873 0.873 0.873 0.873 ...
##  $ P8       : num [1:80] 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.0
##  $ P9       : num [1:80] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
##  $ P10      : num [1:80] 0.0483 0.0483 0.0483 0.0483 0.0529 0.0529 0.0529 0.0529 0.0575 0.0575 ...
##  $ P11      : num [1:80] 0.873 0.873 0.873 0.873 0.935 ...
##  $ P12      : num [1:80] 0.000435 0.000435 0.000435 0.000435 0.000539 0.000539 0.000539 0.000539 0.0
##  $ P13      : num [1:80] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
##  $ P14      : num [1:80] 0.0483 0.0529 0.0575 0.0682 0.0483 0.0529 0.0575 0.0682 0.0483 0.0529 ...
##  $ P15      : num [1:80] 0.873 0.935 0.947 0.951 0.873 ...
##  $ P16      : num [1:80] 0.000435 0.000539 0.000694 0.000817 0.000435 0.000539 0.000694 0.000817 0.0
##  $ R5       : num [1:80] 0.873 0.836 0.891 0.891 0.836 ...
```

```
head(data)
```

```
## # A tibble: 6 x 18
##   `Sample No`    P1     P2    P3      P4    P5     P6    P7      P8    P9    P10
##         <dbl> <dbl>  <dbl> <dbl>   <dbl> <dbl>  <dbl> <dbl>   <dbl> <dbl>  <dbl>
## 1           1  0.01 0.0483 0.873 4.35e-4  0.01 0.0483 0.873 4.35e-4  0.01 0.0483
## 2           2  0.01 0.0483 0.873 4.35e-4  0.01 0.0483 0.873 4.35e-4  0.01 0.0483
## 3           3  0.01 0.0483 0.873 4.35e-4  0.01 0.0483 0.873 4.35e-4  0.01 0.0483
## 4           4  0.01 0.0483 0.873 4.35e-4  0.01 0.0483 0.873 4.35e-4  0.01 0.0483
## 5           5  0.01 0.0483 0.873 4.35e-4  0.01 0.0483 0.873 4.35e-4  0.01 0.0529
## 6           6  0.01 0.0483 0.873 4.35e-4  0.01 0.0483 0.873 4.35e-4  0.01 0.0529
## # ... with 7 more variables: P11 <dbl>, P12 <dbl>, P13 <dbl>, P14 <dbl>,
## #   P15 <dbl>, P16 <dbl>, R5 <dbl>
```

```
library(ggplot2)
```

```
pie(table(data$P2))
```



3. Performing data cleaning and removing the NA values

```
sum(is.na(data))
```

```
## [1] 0
```

There are no empty values so we proceed to correlation analysis

4. Performing correlation analysis

Selecting relevant columns

```
str(data)
```

```
## tibble [80 x 18] (S3: tbl_df/tbl/data.frame)
##  $ Sample No: num [1:80] 1 2 3 4 5 6 7 8 9 10 ...
##  $ P1       : num [1:80] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
##  $ P2       : num [1:80] 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 ...
##  $ P3       : num [1:80] 0.873 0.873 0.873 0.873 0.873 ...
##  $ P4       : num [1:80] 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.00
##  $ P5       : num [1:80] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
##  $ P6       : num [1:80] 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 0.0483 ...
##  $ P7       : num [1:80] 0.873 0.873 0.873 0.873 0.873 ...
##  $ P8       : num [1:80] 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.000435 0.00
##  $ P9       : num [1:80] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
##  $ P10      : num [1:80] 0.0483 0.0483 0.0483 0.0483 0.0529 0.0529 0.0529 0.0529 0.0575 0.0575 ...
##  $ P11      : num [1:80] 0.873 0.873 0.873 0.873 0.935 ...
##  $ P12      : num [1:80] 0.000435 0.000435 0.000435 0.000435 0.000539 0.000539 0.000539 0.000539 0.00
##  $ P13      : num [1:80] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
##  $ P14      : num [1:80] 0.0483 0.0529 0.0575 0.0682 0.0483 0.0529 0.0575 0.0682 0.0483 0.0529 ...
##  $ P15      : num [1:80] 0.873 0.935 0.947 0.951 0.873 ...
##  $ P16      : num [1:80] 0.000435 0.000539 0.000694 0.000817 0.000435 0.000539 0.000694 0.000817 0.00
##  $ R5       : num [1:80] 0.873 0.836 0.891 0.891 0.836 ...
```

```
library(plyr)
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
library(ggplot2)
```

```
library(Hmisc)
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:plyr':
##
##     is.discrete, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
res2<-rcorr(as.matrix(data))
res2
```

```
##           Sample No  P1    P2    P3    P4  P5    P6    P7    P8  P9  P10   P11
## Sample No      1.00 NaN  0.69  0.69  0.69 NaN  0.29  0.06  0.26 NaN 0.19  0.17
## P1              NaN   1   NaN   NaN   NaN  -1   NaN   NaN   NaN  -1  NaN   NaN
## P2             0.69 NaN  1.00  1.00  1.00 NaN -0.45 -0.61 -0.50 NaN 0.00  0.00
## P3             0.69 NaN  1.00  1.00  1.00 NaN -0.45 -0.61 -0.50 NaN 0.00  0.00
## P4             0.69 NaN  1.00  1.00  1.00 NaN -0.45 -0.61 -0.50 NaN 0.00  0.00
## P5              NaN  -1   NaN   NaN   NaN   1   NaN   NaN   NaN  -1  NaN   NaN
## P6             0.29 NaN -0.45 -0.45 -0.45 NaN  1.00  0.82  0.98 NaN 0.00  0.00
## P7             0.06 NaN -0.61 -0.61 -0.61 NaN  0.82  1.00  0.89 NaN 0.00  0.00
## P8             0.26 NaN -0.50 -0.50 -0.50 NaN  0.98  0.89  1.00 NaN 0.00  0.00
## P9              NaN  -1   NaN   NaN   NaN  -1   NaN   NaN   NaN   1  NaN   NaN
## P10            0.19 NaN  0.00  0.00  0.00 NaN  0.00  0.00  0.00 NaN 1.00  0.77
## P11            0.17 NaN  0.00  0.00  0.00 NaN  0.00  0.00  0.00 NaN 0.77  1.00
## P12            0.19 NaN  0.00  0.00  0.00 NaN  0.00  0.00  0.00 NaN 0.97  0.85
## P13             NaN  -1   NaN   NaN   NaN  -1   NaN   NaN   NaN  -1  NaN   NaN
## P14            0.05 NaN  0.00  0.00  0.00 NaN  0.00  0.00  0.00 NaN 0.00  0.00
## P15            0.04 NaN  0.00  0.00  0.00 NaN  0.00  0.00  0.00 NaN 0.00  0.00
```

```
## P16            0.05 NaN  0.00  0.00  0.00 NaN  0.00  0.00  0.00 NaN 0.00  0.00
## R5             0.10 NaN -0.05 -0.05 -0.05 NaN  0.20  0.13  0.20 NaN 0.04 -0.02
##            P12 P13  P14  P15  P16    R5
## Sample No 0.19 NaN 0.05 0.04 0.05  0.10
## P1         NaN  -1  NaN  NaN  NaN   NaN
## P2        0.00 NaN 0.00 0.00 0.00 -0.05
## P3        0.00 NaN 0.00 0.00 0.00 -0.05
## P4        0.00 NaN 0.00 0.00 0.00 -0.05
## P5         NaN  -1  NaN  NaN  NaN   NaN
## P6        0.00 NaN 0.00 0.00 0.00  0.20
## P7        0.00 NaN 0.00 0.00 0.00  0.13
## P8        0.00 NaN 0.00 0.00 0.00  0.20
## P9         NaN  -1  NaN  NaN  NaN   NaN
## P10       0.97 NaN 0.00 0.00 0.00  0.04
## P11       0.85 NaN 0.00 0.00 0.00 -0.02
## P12       1.00 NaN 0.00 0.00 0.00  0.02
## P13        NaN   1  NaN  NaN  NaN   NaN
## P14       0.00 NaN 1.00 0.77 0.97  0.03
## P15       0.00 NaN 0.77 1.00 0.85  0.10
## P16       0.00 NaN 0.97 0.85 1.00  0.06
## R5        0.02 NaN 0.03 0.10 0.06  1.00
##
## n= 80
##
##
## P
##           Sample No P1     P2     P3     P4     P5     P6     P7     P8
## Sample No                  0.0000 0.0000 0.0000        0.0101 0.5750 0.0217
## P1                                              0.0000
## P2        0.0000                 0.0000 0.0000        0.0000 0.0000 0.0000
## P3        0.0000           0.0000        0.0000        0.0000 0.0000 0.0000
## P4        0.0000           0.0000 0.0000               0.0000 0.0000 0.0000
## P5                 0.0000
## P6        0.0101           0.0000 0.0000 0.0000               0.0000 0.0000
## P7        0.5750           0.0000 0.0000 0.0000        0.0000        0.0000
## P8        0.0217           0.0000 0.0000 0.0000        0.0000 0.0000
## P9                 0.0000                         0.0000
## P10       0.0938           1.0000 1.0000 1.0000        1.0000 1.0000 1.0000
## P11       0.1325           1.0000 1.0000 1.0000        1.0000 1.0000 1.0000
## P12       0.0860           1.0000 1.0000 1.0000        1.0000 1.0000 1.0000
## P13                0.0000                         0.0000
## P14       0.6778           1.0000 1.0000 1.0000        1.0000 1.0000 1.0000
## P15       0.7087           1.0000 1.0000 1.0000        1.0000 1.0000 1.0000
## P16       0.6706           1.0000 1.0000 1.0000        1.0000 1.0000 1.0000
## R5        0.3575           0.6459 0.6459 0.6459        0.0788 0.2485 0.0770
##           P9     P10    P11    P12    P13    P14    P15    P16    R5
## Sample No        0.0938 0.1325 0.0860        0.6778 0.7087 0.6706 0.3575
## P1        0.0000                       0.0000
## P2              1.0000 1.0000 1.0000        1.0000 1.0000 1.0000 0.6459
## P3              1.0000 1.0000 1.0000        1.0000 1.0000 1.0000 0.6459
## P4              1.0000 1.0000 1.0000        1.0000 1.0000 1.0000 0.6459
## P5        0.0000                       0.0000
## P6              1.0000 1.0000 1.0000        1.0000 1.0000 1.0000 0.0788
## P7              1.0000 1.0000 1.0000        1.0000 1.0000 1.0000 0.2485
```

```
## P8               1.0000 1.0000 1.0000        1.0000 1.0000 1.0000 0.0770
## P9                                     0.0000
## P10                    0.0000 0.0000        1.0000 1.0000 1.0000 0.6929
## P11              0.0000        0.0000        1.0000 1.0000 1.0000 0.8687
## P12              0.0000 0.0000               1.0000 1.0000 1.0000 0.8473
## P13       0.0000
## P14              1.0000 1.0000 1.0000              0.0000 0.0000 0.8139
## P15              1.0000 1.0000 1.0000        0.0000        0.0000 0.3851
## P16              1.0000 1.0000 1.0000        0.0000 0.0000        0.6198
## R5               0.6929 0.8687 0.8473        0.8139 0.3851 0.6198
```

Strong correlation between R5 and the variables P6, P7, P8, P15

These columns are hence taken into consideration for regression Splitting the dataset into training and testing

```
dat<-data
set.seed(100)

index = sample(1:nrow(dat), 0.8*nrow(dat))

train = dat[index,] # Create the training data
test = dat[-index,] # Create the test data

dim(train)
```

```
## [1] 64 18
```

```
dim(test)
```

```
## [1] 16 18
```

Scaling the numeric features

```
cols<-c("P6", "P7", "P8", "P15", "R5")
pre_proc_val <- preProcess(train[,cols], method = c("center", "scale"))

train[,cols] = predict(pre_proc_val, train[,cols])
test[,cols] = predict(pre_proc_val, test[,cols])

summary(train)
```

```
##    Sample No           P1              P2                P3
##  Min.   : 1.00   Min.   :0.01   Min.   :0.04830   Min.   :0.8728
##  1st Qu.:19.75   1st Qu.:0.01   1st Qu.:0.04830   1st Qu.:0.8728
##  Median :39.50   Median :0.01   Median :0.04830   Median :0.8728
##  Mean   :39.98   Mean   :0.01   Mean   :0.04931   Mean   :0.8864
##  3rd Qu.:61.25   3rd Qu.:0.01   3rd Qu.:0.04830   3rd Qu.:0.8728
##  Max.   :80.00   Max.   :0.01   Max.   :0.05290   Max.   :0.9350
##        P4                 P5             P6                P7
##  Min.   :0.0004350   Min.   :0.01   Min.   :-0.8246   Min.   :-1.1118
##  1st Qu.:0.0004350   1st Qu.:0.01   1st Qu.:-0.8246   1st Qu.:-1.1118
##  Median :0.0004350   Median :0.01   Median :-0.1856   Median : 0.6317
```

```
##  Mean   :0.0004577   Mean   :0.01   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.:0.0004350   3rd Qu.:0.01   3rd Qu.: 0.4534   3rd Qu.: 0.9680
##  Max.   :0.0005390   Max.   :0.01   Max.   : 1.9398   Max.   : 1.0886
##        P8                 P9             P10                P11
##  Min.   :-0.8921   Min.   :0.01   Min.   :0.04830   Min.   :0.8728
##  1st Qu.:-0.8921   1st Qu.:0.01   1st Qu.:0.05290   1st Qu.:0.9350
##  Median :-0.1811   Median :0.01   Median :0.05750   Median :0.9470
##  Mean   : 0.0000   Mean   :0.01   Mean   :0.05749   Mean   :0.9301
##  3rd Qu.: 0.8784   3rd Qu.:0.01   3rd Qu.:0.06820   3rd Qu.:0.9513
##  Max.   : 1.7192   Max.   :0.01   Max.   :0.06820   Max.   :0.9513
##       P12                P13             P14                P15
##  Min.   :0.0004350   Min.   :0.01   Min.   :0.04830   Min.   :-1.6240
##  1st Qu.:0.0005390   1st Qu.:0.01   1st Qu.:0.04830   1st Qu.:-1.6240
##  Median :0.0006940   Median :0.01   Median :0.05750   Median : 0.6550
##  Mean   :0.0006372   Mean   :0.01   Mean   :0.05656   Mean   : 0.0000
##  3rd Qu.:0.0008170   3rd Qu.:0.01   3rd Qu.:0.05750   3rd Qu.: 0.6550
##  Max.   :0.0008170   Max.   :0.01   Max.   :0.06820   Max.   : 0.7871
##       P16                 R5
##  Min.   :0.0004350   Min.   :-1.48026
##  1st Qu.:0.0004350   1st Qu.:-0.84868
##  Median :0.0006940   Median :-0.05921
##  Mean   :0.0006201   Mean   : 0.00000
##  3rd Qu.:0.0006940   3rd Qu.: 1.04605
##  Max.   :0.0008170   Max.   : 1.67763
```

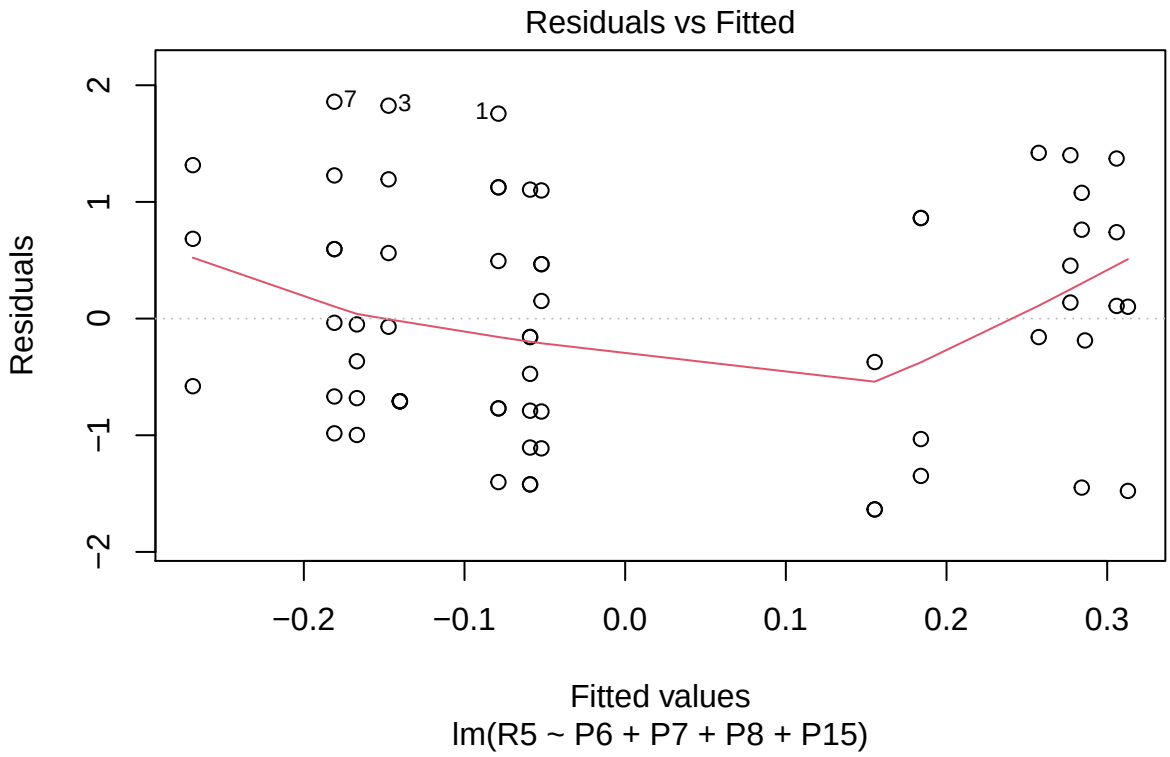
Applying linear regression between loan amount and lender count

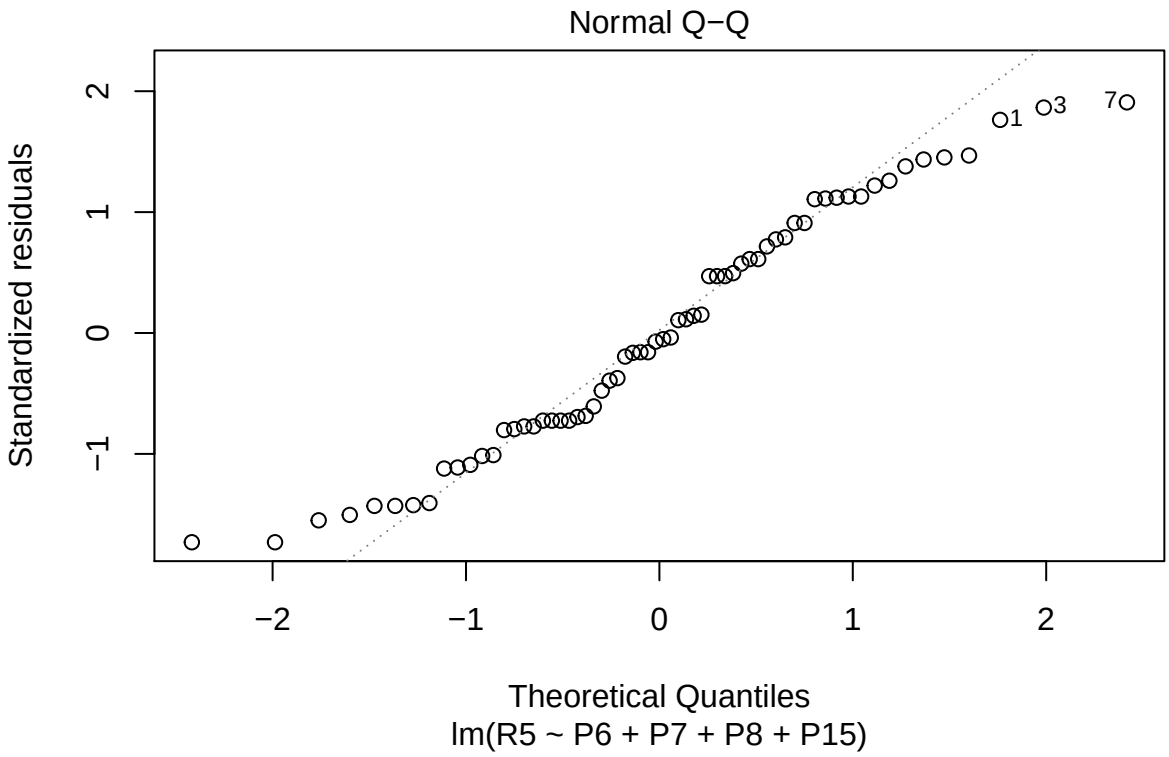
```
lr = lm(R5 ~ P6+P7+P8+P15, data = train)
summary(lr)
```

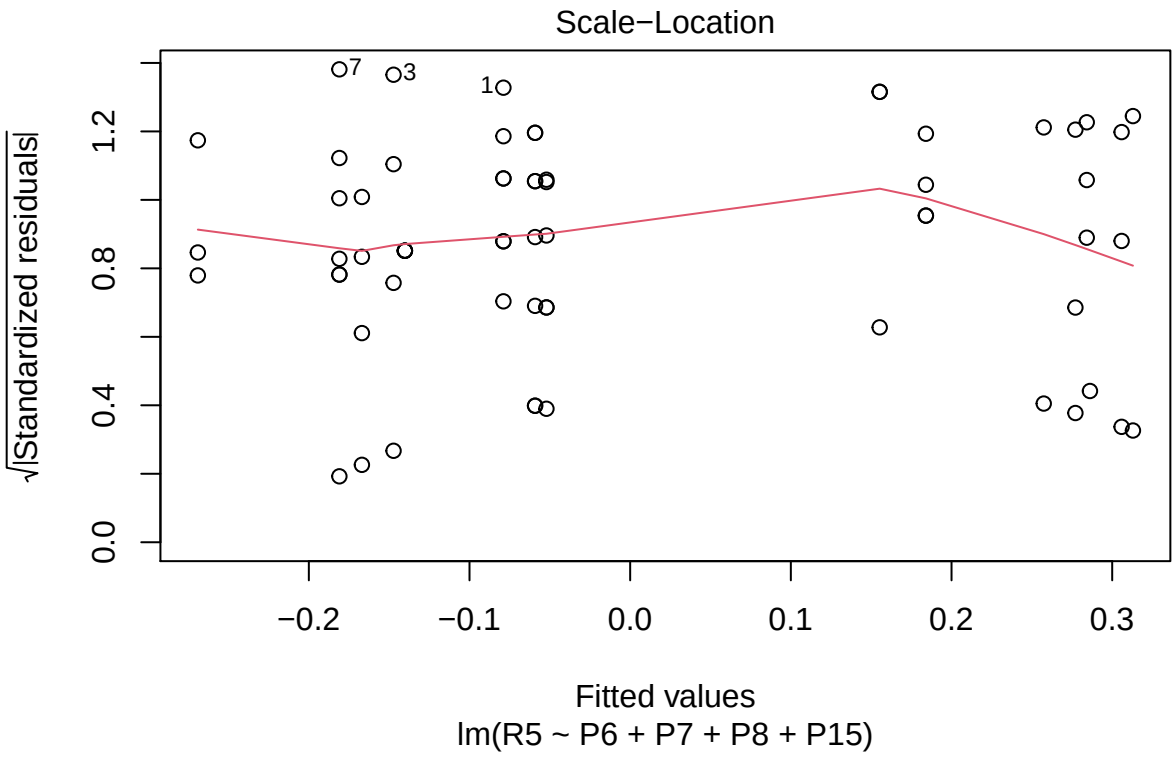
```
##
## Call:
## lm(formula = R5 ~ P6 + P7 + P8 + P15, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.63556 -0.76980 -0.05999  0.78692  1.85862
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.268e-15  1.269e-01   0.000    1.000
## P6          -4.263e-01  7.208e-01  -0.591    0.556
## P7          -1.993e-01  2.926e-01  -0.681    0.498
## P8           7.481e-01  8.586e-01   0.871    0.387
## P15          5.345e-02  1.289e-01   0.415    0.680
##
## Residual standard error: 1.015 on 59 degrees of freedom
## Multiple R-squared:  0.03464,    Adjusted R-squared:  -0.0308
## F-statistic: 0.5293 on 4 and 59 DF,  p-value: 0.7146
```

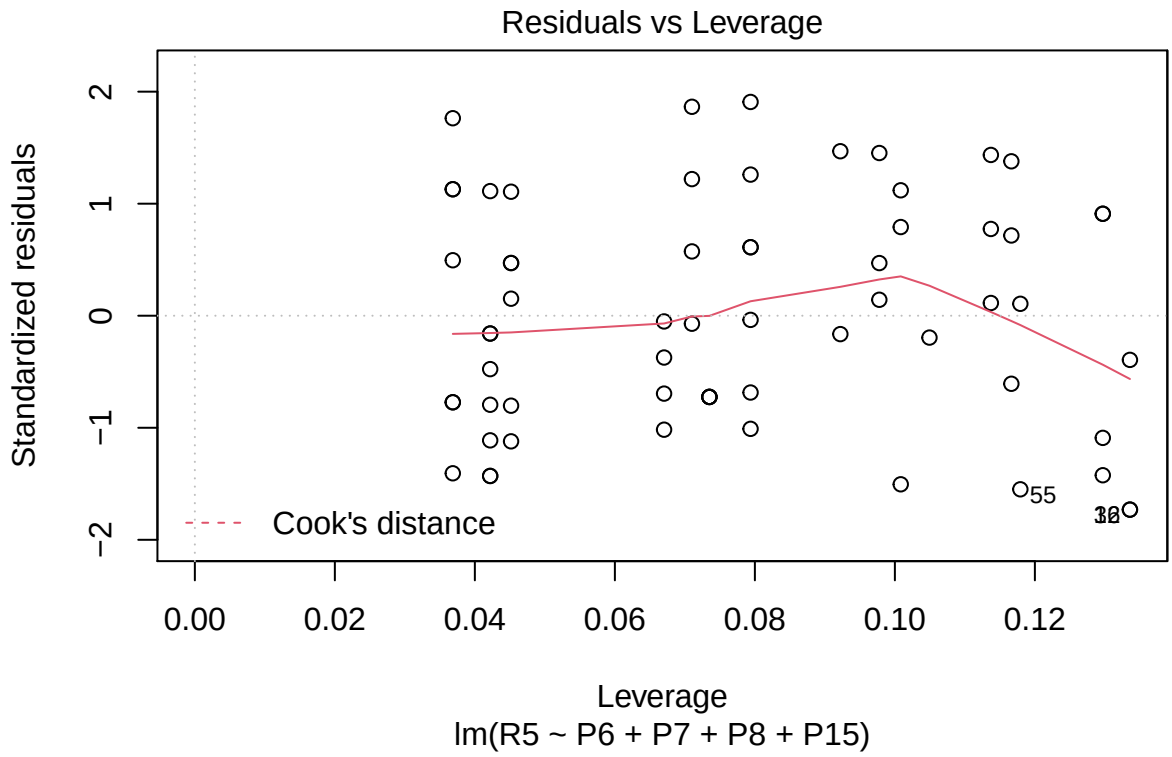
Evaluating the model

```
plot(lr)
```


Residuals vs Fitted
Residuals
Fitted values
lm(R5 ~ P6 + P7 + P8 + P15)
7
3
1
2
1
0
−1
−2
−0.2
−0.1
0.0
0.1
0.2
0.3


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(R5 ~ P6 + P7 + P8 + P15)
1
3
7


Scale−Location
√|Standardized residuals|
0.0
0.4
0.8
1.2
7
3
1
−0.2
−0.1
0.0
0.1
0.2
0.3
Fitted values
lm(R5 ~ P6 + P7 + P8 + P15)




```
eval_metrics = function(model, df, predictions, target){
    resids = df[,target] - predictions
    resids2 = resids**2
    N = length(predictions)
    r2 = as.character(round(summary(model)$r.squared, 2))
    adj_r2 = as.character(round(summary(model)$adj.r.squared, 2))
    print(adj_r2) #Adjusted R-squared
    print(as.character(round(sqrt(sum(resids2)/N), 2))) #RMSE
}

# Step 2 - predicting and evaluating the model on train data
predictions = predict(lr, newdata = train)
eval_metrics(lr, train, predictions, target = 'R5')
```

```
## [1] "-0.03"
## [1] "0.97"
```

```
# Step 3 - predicting and evaluating the model on test data
predictions = predict(lr, newdata = test)
eval_metrics(lr, test, predictions, target = 'R5')
```

```
## [1] "-0.03"
## [1] "0.89"
```

```
d<-predictions - test[,c('R5')]
mse = mean((d[,c('R5')])^2)
mse
```

```
## [1] 0.7913583
```

```
mae = mean(abs(d[,c('R5')]))
mae
```

```
## [1] 0.7630055
```

RMSE for training -> -0.03 adj R squared for training -> 0.97

RMSE for testing -> -0.03 adj R squared for testing -> 0.89 MSE -> 0.79 MAE -> 0.76

Performign reguliarization

```
dummies <- dummyVars(R5 ~ ., data = dat[,cols])

train_dummies = predict(dummies, newdata = train[,cols])

test_dummies = predict(dummies, newdata = test[,cols])

print(dim(train_dummies))
```

```
## [1] 64  4
```

```
print(dim(test_dummies))
```

```
## [1] 16  4
```

Applying ridge regression

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-1
```

```
x = as.matrix(train_dummies)
y_train = train$R5

x_test = as.matrix(test_dummies)
y_test = test$R5

lambdas <- 10^seq(2, -3, by = -.1)
ridge_reg = glmnet(x, y_train, nlambda = 25, alpha = 0, family = 'gaussian', lambda = lambdas)

summary(ridge_reg)
```

```
##           Length Class     Mode
## a0         51    -none-    numeric
## beta      204    dgCMatrix S4
## df         51    -none-    numeric
## dim         2    -none-    numeric
## lambda     51    -none-    numeric
## dev.ratio  51    -none-    numeric
## nulldev     1    -none-    numeric
## npasses     1    -none-    numeric
## jerr        1    -none-    numeric
## offset      1    -none-    logical
## call        7    -none-    call
## nobs        1    -none-    numeric
```

Finding the optimal lambda

```
cv_ridge <- cv.glmnet(x, y_train, alpha = 0, lambda = lambdas)
optimal_lambda <- cv_ridge$lambda.min
optimal_lambda
```

```
## [1] 100
```

Evaluation metrics

```
# Compute R^2 from true and predicted values
eval_results <- function(true, predicted, df) {
  SSE <- sum((predicted - true)^2)
  SST <- sum((true - mean(true))^2)
  R_square <- 1 - SSE / SST
  RMSE = sqrt(SSE/nrow(df))


  # Model performance metrics
data.frame(
  RMSE = RMSE,
  Rsquare = R_square
)

}

# Prediction and evaluation on train data
predictions_train <- predict(ridge_reg, s = optimal_lambda, newx = x)
eval_results(y_train, predictions_train, train)
```

```
##        RMSE     Rsquare
## 1 0.9916272 0.001067218
```

```
# Prediction and evaluation on test data
predictions_test <- predict(ridge_reg, s = optimal_lambda, newx = x_test)
eval_results(y_test, predictions_test, test)
```

```
##        RMSE     Rsquare
## 1 0.9492317 0.002986305
```

```r
d<-predictions - test[,c('R5')]
mse = mean((d[,c('R5')])^2)
mse
```

```
## [1] 0.7913583
```

```r
mae = mean(abs(d[,c('R5')]))
mae
```

```
## [1] 0.7630055
```

RMSE -> 0.94 R squared -> 0.0029 MSE -> 0.79 MAE -> 0.76

Performing Lasso regression

```r
lambdas <- 10^seq(2, -3, by = -.1)

# Setting alpha = 1 implements lasso regression
lasso_reg <- cv.glmnet(x, y_train, alpha = 1, lambda = lambdas, standardize = TRUE, nfolds = 5)

# Best
lambda_best <- lasso_reg$lambda.min
lambda_best
```

```
## [1] 100
```

Evaluation metrics

```r
lasso_model <- glmnet(x, y_train, alpha = 1, lambda = lambda_best, standardize = TRUE)

predictions_train <- predict(lasso_model, s = lambda_best, newx = x)
eval_results(y_train, predictions_train, train)
```

```
##        RMSE Rsquare
## 1 0.9921567       0
```

```r
predictions_test <- predict(lasso_model, s = lambda_best, newx = x_test)
eval_results(y_test, predictions_test, test)
```

```
##        RMSE       Rsquare
## 1 0.9508571 -0.0004310356
```

```r
d<-predictions - test[,c('R5')]
mse = mean((d[,c('R5')])^2)
mse
```

```
## [1] 0.7913583
```

```r
mae = mean(abs(d[,c('R5')]))
mae
```

```
## [1] 0.7630055
```

RMSE -> 0.95 R squared -> -0.0004 MSE -> 0.79 MAE -> 0.76

Result Based on all the RMSE, R squared, MSE and MAE values we got we can see that lineaar regression has the highest R squared value and lowest RMSE value making it a good model for R5.