INTRODUCTION

The project is to build a regression model to estimate the relative CPU performance of computer hardware dataset. Relative CPU performance of the computer hardware is described in terms of machine cycle time, main memory, cache memory and minimum and maximum channels as given in the dataset. To implement this R programming language is used.

DATASET

The computer hardware dataset consists of information about the computer vendors selling computers, model name of computers and various attributes to estimate the relative performance of CPU.

The dataset can be found at the following url –

https://archive.ics.uci.edu/ml/datasets/Computer+Hardware

Dataset description is given as follows: -

vendor name: 30
(adviser, amdahl,apollo, basf, bti, burroughs, c.r.d, cambex, cdc, dec, dg, formation, four-phase, gould, honeywell, hp, ibm, ipl, magnuson, microdata, nas, ncr, nixdorf, perkin-elmer, prime, siemens, sperry, sratus, wang)

Model Name: many unique symbols

MYCT: machine cycle time in nanoseconds (integer)

MMIN: minimum main memory in kilobytes (integer)

MMAX: maximum main memory in kilobytes (integer)

CACH: cache memory in kilobytes (integer)

CHMIN: minimum channels in units (integer)

CHMAX: maximum channels in units (integer)

PRP: published relative performance (integer)

ERP: estimated relative performance from the original article (integer)

There are 2 categorical variables and 8 numerical variables. The 2 categorical variables, Vendor Name and Model Name are 2 non-predictive attributes as given in the dataset description. All of the 8 numerical variables are of discrete type. ERP (estimated relative performance is the goal field). It is the target variable.

IMPORT THE DATASET

```
machine<- read.csv("c:/data/machine.csv")
```

EXPLORATORY DATA ANALYSIS

It provides useful insights into the dataset which is important for further analysis.

View of top 6 data from the dataset

```
head(machine)
```

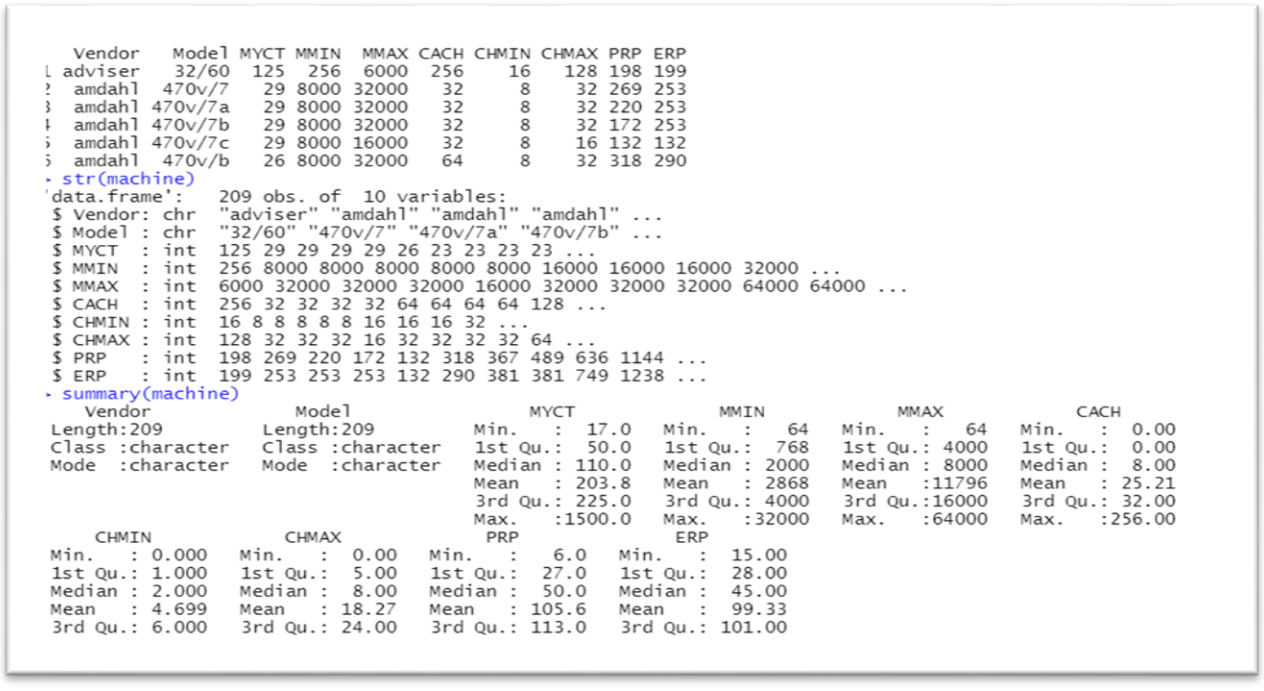
Structure of dataset can be viewed as

```
str(machine)
```

Summary of dataset:

```
summary(machine)
```

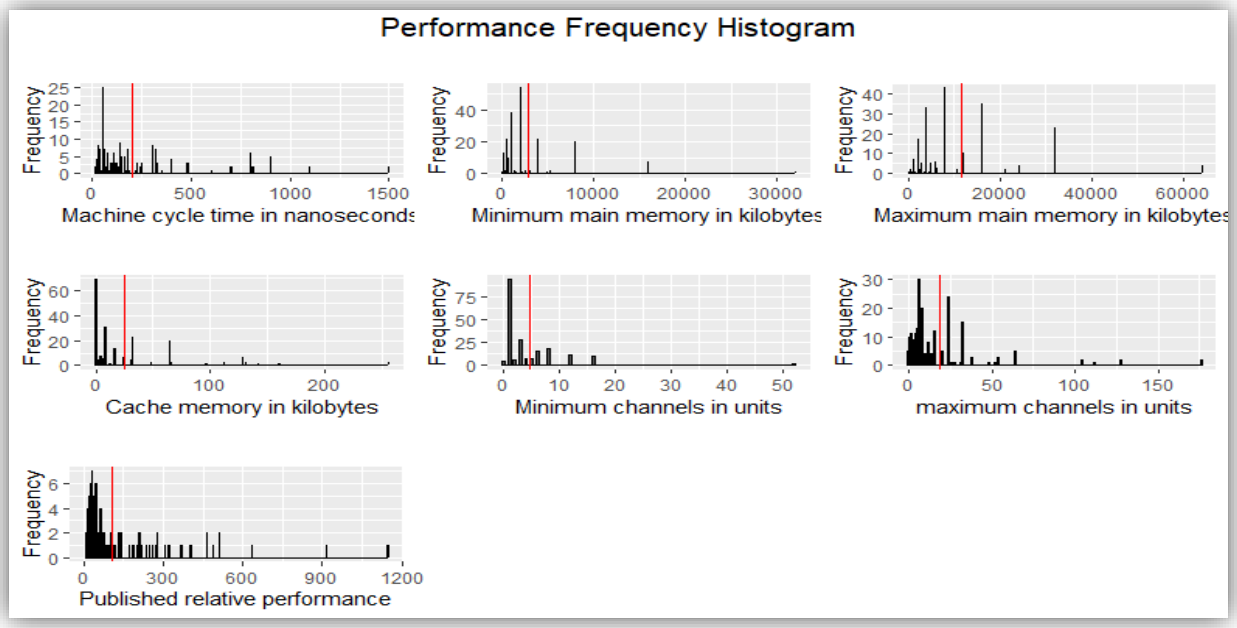
Output of above will be as follows:



```
   Vendor   Model MYCT MMIN  MMAX CACH CHMIN CHMAX PRP ERP
1 adviser   32/60  125  256  6000  256    16   128 198 199
2  amdahl  470v/7   29 8000 32000   32     8    32 269 253
3  amdahl 470v/7a   29 8000 32000   32     8    32 220 253
4  amdahl 470v/7b   29 8000 32000   32     8    32 172 253
5  amdahl 470v/7c   29 8000 16000   32     8    16 132 132
6  amdahl  470v/b   26 8000 32000   64     8    32 318 290
> str(machine)
'data.frame':   209 obs. of  10 variables:
 $ Vendor: chr  "adviser" "amdahl" "amdahl" "amdahl" ...
 $ Model : chr  "32/60" "470v/7" "470v/7a" "470v/7b" ...
 $ MYCT  : int  125 29 29 29 29 26 23 23 23 23 ...
 $ MMIN  : int  256 8000 8000 8000 8000 8000 16000 16000 16000 32000 ...
 $ MMAX  : int  6000 32000 32000 32000 16000 32000 32000 32000 64000 64000 ...
 $ CACH  : int  256 32 32 32 32 64 64 64 64 128 ...
 $ CHMIN : int  16 8 8 8 8 8 16 16 16 32 ...
 $ CHMAX : int  128 32 32 32 16 32 32 32 32 64 ...
 $ PRP   : int  198 269 220 172 132 318 367 489 636 1144 ...
 $ ERP   : int  199 253 253 253 132 290 381 381 749 1238 ...
> summary(machine)
    Vendor             Model                MYCT             MMIN             MMAX             CACH       
Length:209         Length:209         Min.   :  17.0   Min.   :   64   Min.   :   64   Min.   :  0.00  
Class :character   Class :character   1st Qu.:  50.0   1st Qu.:  768   1st Qu.: 4000   1st Qu.:  0.00  
Mode  :character   Mode  :character   Median : 110.0   Median : 2000   Median : 8000   Median :  8.00  
                                      Mean   : 203.8   Mean   : 2868   Mean   :11796   Mean   : 25.21  
                                      3rd Qu.: 225.0   3rd Qu.: 4000   3rd Qu.:16000   3rd Qu.: 32.00  
                                      Max.   :1500.0   Max.   :32000   Max.   :64000   Max.   :256.00  
     CHMIN            CHMAX             PRP              ERP        
 Min.   : 0.000   Min.   :  0.00   Min.   :   6.0   Min.   :  15.00  
 1st Qu.: 1.000   1st Qu.:  5.00   1st Qu.:  27.0   1st Qu.:  28.00  
 Median : 2.000   Median :  8.00   Median :  50.0   Median :  45.00  
 Mean   : 4.699   Mean   : 18.27   Mean   : 105.6   Mean   :  99.33  
 3rd Qu.: 6.000   3rd Qu.: 24.00   3rd Qu.: 113.0   3rd Qu.: 101.00  
```

DATA VISUALIZATION

Visualizing each attribute to get a clear view of dataset.



REGRESSION

- **LINEAR REGRESSION**

For linear regression, first build a linear model. The function used for linear regression is lm(). Summary statistics can be viewed using summary().

```
machine_lm  <-   lm(ERP~ MYCT+MMIN + MMAX + CACH + CHMIN + CHMAX+ PRP , data = machine )
summary(machine_lm)
```

```
> summary(machine_lm)

Call:
lm(formula = ERP ~ MYCT + MMIN + MMAX + CACH + CHMIN + CHMAX +
    PRP, data = machine)

Residuals:
     Min       1Q   Median       3Q      Max
-117.478   -9.546    2.864   15.257  182.251

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.423e+01  4.732e+00  -7.234 9.68e-12 ***
MYCT         3.777e-02  9.434e-03   4.004 8.77e-05 ***
MMIN         5.483e-03  1.120e-03   4.894 2.02e-06 ***
MMAX         3.375e-03  3.974e-04   8.493 4.45e-15 ***
CACH         1.244e-01  7.751e-02   1.605  0.11016
CHMIN       -1.634e-02  4.523e-01  -0.036  0.97122
CHMAX        3.458e-01  1.287e-01   2.687  0.00781 **
PRP          5.770e-01  3.718e-02  15.519  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 31.7 on 201 degrees of freedom
Multiple R-squared:  0.9595,    Adjusted R-squared:  0.958
F-statistic: 679.5 on 7 and 201 DF,  p-value: < 2.2e-16
```

In the above table, the more the stars beside the variable's p-Value, the more significant the variable. Other variables can be eliminated because otherwise it will lead to overfitting.

```
machine_lm_ <-  lm(ERP~ MYCT + MMIN + MMAX + CHMAX + PRP ,  data = machine)
 summary(machine_lm_)
```

```
> summary(machine_lm_)

Call:
lm(formula = ERP ~ MYCT + MMIN + MMAX + CHMAX + PRP, data = machine)

Residuals:
    Min      1Q  Median      3Q     Max
-110.78  -10.44    2.62   14.22  173.18

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.254e+01  4.620e+00  -7.044 2.85e-11 ***
MYCT         3.521e-02  9.313e-03   3.781 0.000205 ***
MMIN         5.648e-03  1.099e-03   5.138 6.50e-07 ***
MMAX         3.275e-03  3.929e-04   8.334 1.16e-14 ***
CHMAX        3.772e-01  1.212e-01   3.112 0.002124 **
PRP          5.962e-01  3.537e-02  16.855  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 31.76 on 203 degrees of freedom
Multiple R-squared:  0.9589,    Adjusted R-squared:  0.9579
F-statistic: 947.3 on 5 and 203 DF,  p-value: < 2.2e-16
```

The multiple R-squared value is 0.9589. So, the accuracy for this model is 95.89%.

95% of confidence interval on parameters:

```
confint(machine_lm_,level = .95)
```

```
> confint(machine_lm_,level = .95)
                    2.5 %        97.5 %
(Intercept) -41.650503942 -23.431934734
MYCT          0.016849206   0.053573673
MMIN          0.003480802   0.007815813
MMAX          0.002500041   0.004049503
CHMAX         0.138252183   0.616234390
PRP           0.526476293   0.665965654
```

In the earlier estimated value of PRP coefficient was 0.59. Using confint(), the confidence interval is (0.52, 0.66), which provides the amount of uncertainty in the estimate. Using a set of input variable values, the predict() function provides a 95%confidence interval.

```
MYCT <-  30
MMIN <- 8500
MMAX <- 20000
CHMAX <- 32
 PRP <- 150
 new_pt <-  data.frame(MYCT+MMIN + MMAX + CHMAX +PRP )
 conf_interval <-  predict(machine_lm_ , new_pt, level=.95, interval="confidence")
 conf_interval
```

```
> conf_interval
       fit      lwr      upr
1 183.5261 171.5998 195.4524
```

The estimated relative performance is 183 with a 95%confidence interval of (171, 195).

PREDICTION INTERVAL:

The predict() function provides the ability to compute upper and lower bounds on a particular outcome.

```
pred_interval <- predict(machine_lm_, new_pt, level=.95, interval="prediction")
pred_interval
```

```
> pred_interval
       fit      lwr      upr
1 183.5261 119.7856 247.2666
```

Again, the estimated relative performance is 183. The 95% prediction interval is (119, 247).

- **RIDGE REGRESSION**

For Ridge regression, define response variable and matrix of predictor variables. To build the ridge regression glmnetfunction from glmnet package is used. Using ridge regression, we can predict the estimated relative performance of computer.

```
res_data <- machine$ERP
pred_data <- data.matrix(machine[, c( 'MYCT','MMIN','MMAX','CACH','CHMIN','CHMAX',
'PRP')])
model <- glmnet(pred_data, res_data, alpha = 0)
summary(model)
```

```
> summary(model)
          Length Class     Mode
a0        100    -none-    numeric
beta      700    dgCMatrix S4
df        100    -none-    numeric
dim         2    -none-    numeric
lambda    100    -none-    numeric
dev.ratio 100    -none-    numeric
nulldev     1    -none-    numeric
npasses     1    -none-    numeric
jerr        1    -none-    numeric
offset      1    -none-    logical
call        4    -none-    call
nobs        1    -none-    numeric
```

The next task is to identify the optimal value of lambda that will result in a minimum error. This can be achieved automatically by using cv.glmnet() function.

```
cv <- cv.glmnet(pred_data, res_data, alpha = 0)
Lambda_value <- cv$lambda.min
lambda_value
```

```
> lambda_value
[1] 14.92101
```

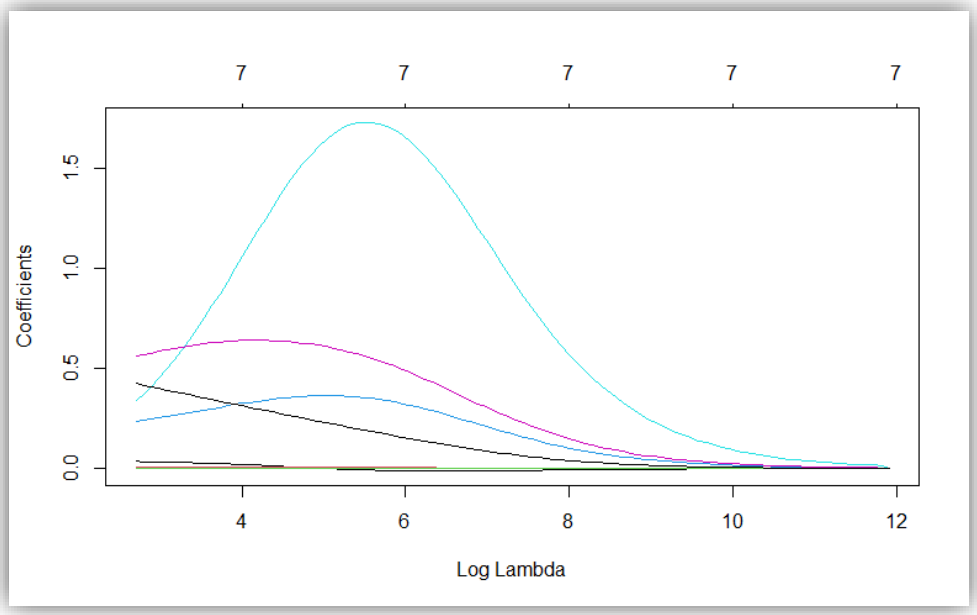
Rebuilding the model with optimal lambda value and checking the coefficients.

```
machine_model <- glmnet(pred_data, res_data, alpha = 0, lambda = lambda_value)
coef(machine_model)
```

```
> coef(machine_model)
8 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept) -35.869917372
MYCT          0.035315733
MMIN          0.007674494
MMAX          0.003672641
CACH          0.234682682
CHMIN         0.338282890
CHMAX         0.560395388
PRP           0.425351637
```

Plot to visualize how the coefficient estimates changed as a result of increasing lambda

```
plot(model, xvar = "lambda")
```



The next task is to use the predict function and compute the R2 value.

```
res_predicted  <-  predict(model, s = lambda_value, newx = pred_data)
sse <- sum((res_predicted - res_data)^2)          #sum of squared errors
sst <- sum((res_data - mean(res_data))^2)         #sum of squared total
rs  <- 1 - sse/sst
rs
```

```
> rs <- 1 - sse/sst
> rs
[1] 0.9548386
```

The R squared turns to be 0.9548 which results in 95.48% accuracy

CONCLUSION

The R squared obtained from Linear regression is 0.9589 and Ridge regression is 0.9548.

Results in similar accuracy for both the models but linear regression is slightly better than Ridge regression.