Computer Hardware Data Set

Data Set Information:

The estimated relative performance values were estimated by the authors using a linear regression method.

Attribute Information:

1. 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)
2. Model Name: many unique symbols
3. MYCT: machine cycle time in nanoseconds (integer)
4. MMIN: minimum main memory in kilobytes (integer)
5. MMAX: maximum main memory in kilobytes (integer)
6. CACH: cache memory in kilobytes (integer)
7. CHMIN: minimum channels in units (integer)
8. CHMAX: maximum channels in units (integer)
9. PRP: published relative performance (integer)
10. ERP: estimated relative performance from the original article (integer)

R Code:

Linear Regression

```
setwd('~/Desktop/dm') #set working directory to desktop
machine=read.table("machine.data",sep=",",header=TRUE)
summary(machine)
```

```
      vendor           model          myct            mmin            mmax
ibm      : 32   100       :  1   Min.   :  17.0   Min.   :   64   Min.   :   64
nas      : 19   1100/61-h1:  1   1st Qu.:  50.0   1st Qu.:  768   1st Qu.: 4000
honeywell: 13   1100/81   :  1   Median : 110.0   Median : 2000   Median : 8000
ncr      : 13   1100/82   :  1   Mean   : 203.8   Mean   : 2868   Mean   :11796
sperry   : 13   1100/83   :  1   3rd Qu.: 225.0   3rd Qu.: 4000   3rd Qu.:16000
siemens  : 12   1100/84   :  1   Max.   :1500.0   Max.   :32000   Max.   :64000
(Other)  :107   (Other)   :203
     cach            chmin            chmax            prp              erp
Min.   :  0.00   Min.   : 0.000   Min.   :  0.00   Min.   :   6.0   Min.   :  15.00
1st Qu.:  0.00   1st Qu.: 1.000   1st Qu.:  5.00   1st Qu.:  27.0   1st Qu.:  28.00
Median :  8.00   Median : 2.000   Median :  8.00   Median :  50.0   Median :  45.00
Mean   : 25.21   Mean   : 4.699   Mean   : 18.27   Mean   : 105.6   Mean   :  99.33
3rd Qu.: 32.00   3rd Qu.: 6.000   3rd Qu.: 24.00   3rd Qu.: 113.0   3rd Qu.: 101.00
Max.   :256.00   Max.   :52.000   Max.   :176.00   Max.   :1150.0   Max.   :1238.00
```

The above output shows the mean 1st and 3rd quadrant with median and maximum value of each of the attributes in the table. Take an example of 'cach' attribute in the above figure which corresponds to each unique row in the dataset and as the max value is 256 it shows max as 256 with its mean as 25.21 and median as 8.

- mach<-subset(machine, select = c('prp', 'erp')) #select 2 attributes from dataset
- summary(mach)

```
      prp              erp
Min.   :   6.0   Min.   :  15.00
1st Qu.:  27.0   1st Qu.:  28.00
Median :  50.0   Median :  45.00
Mean   : 105.6   Mean   :  99.33
3rd Qu.: 113.0   3rd Qu.: 101.00
Max.   :1150.0   Max.   :1238.00
```

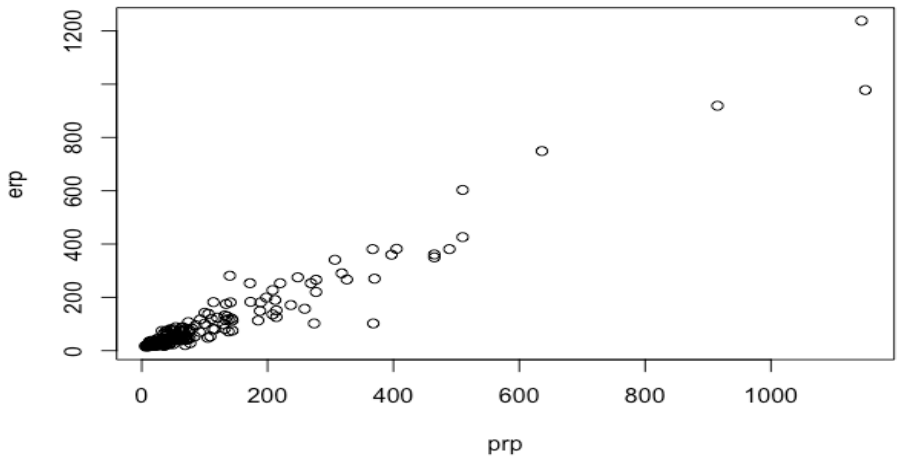
In the above 2 lines of code we select 2 attributes from the dataset and their summary is displayed as the output shown in the figure above.

#correlation of mach object

- cor(mach)
  we correlate the two attributes to look for any errors

```
            prp       erp
prp 1.0000000 0.9664717
erp 0.9664717 1.0000000
```

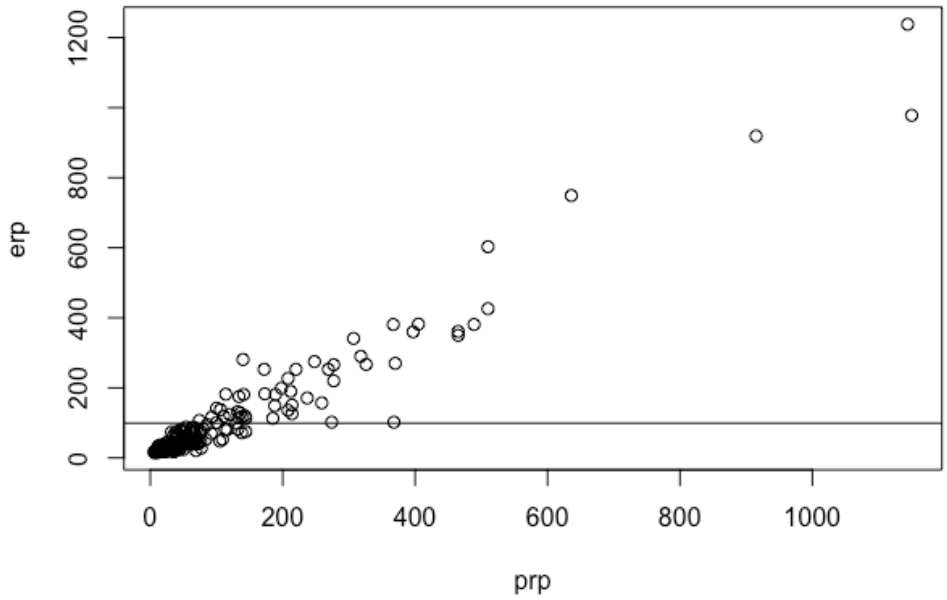
- plot(mach)



The above plot is plotted with erp on y axis and prp on x axis.

- mean.erp<-mean(machine$erp) #calculating the mean
- abline(h=mean.erp) #plot mean line

The Mean is calculated on y axis for erp label and is plotted on the graph as shown below:



The Linear Regression Function is given by:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

The goal is to find some values of θ (known as coefficients), so we can minimize the difference between real and predicted values of dependent variable(y).

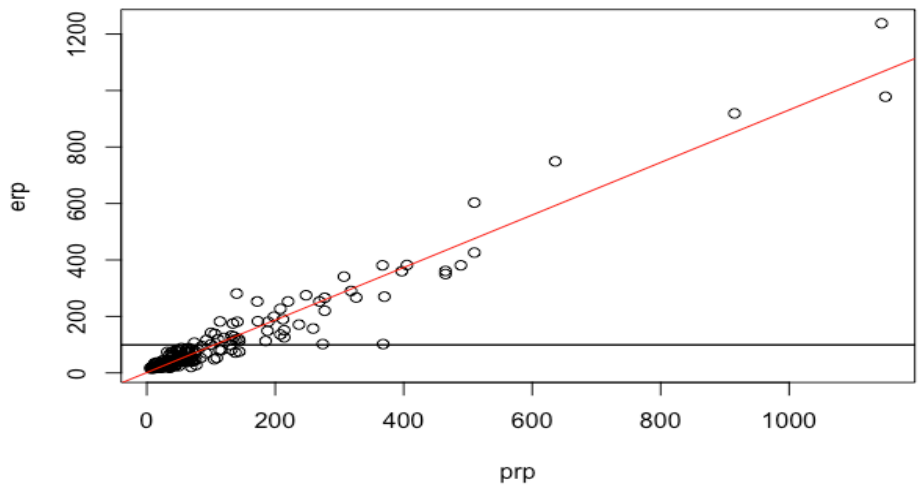
- lmreg <- lm(erp ~ prp,data=machine) # linear regression on dataset
  # Summarize and print the results
- summary(lmreg) # show regression coefficients table
  The summary after the lm function is stored in 'lmreg' variable and its output is as shown below:

```
Call:
lm(formula = erp ~ prp, data = machine)

Residuals:
     Min       1Q   Median       3Q      Max
-241.335  -10.304    1.856   12.327  173.006

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.10445    3.29877   0.335    0.738
prp          0.92997    0.01717  54.153   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 39.83 on 207 degrees of freedom
Multiple R-squared:  0.9341,    Adjusted R-squared:  0.9337
F-statistic:  2933 on 1 and 207 DF,  p-value: < 2.2e-16
```

Values of coefficients (θs) are 1.10445 and 0.92997, hence prediction equation for model is as below:
*prp = 1.10445 + 0.92997*erp*
In the output, *residual standard error* is cost which is 39.83.
We map the line in red color on the graph with code below:

- abline(lmreg,col="red") #linear regression line

The output for the above code is as follows:



Now we compare the predicted values that are generated from the lm function with the actual values.

- machine$erp #original values

This code displays the actual 'Water' attribute values from the dataset. The output is as depicted below:

```
  [1]  199  253  253  253  132  290  381  381  749 1238   23   24   70  117   15   64   23   29
 [19]   22  124   35   39   40   45   28   21   28   22   28   27  102  102   74   74  138  136
 [37]   23   29   44   30   41   74   74   74   54   41   18   28   36   38   34   19   72   36
 [55]   30   56   42   34   34   34   34   34   19   75  113  157   18   20   28   33   47   54
 [73]   20   23   25   52   27   50   18   53   23   30   73   20   25   28   29   32  175   57
 [91]  181  181   32   82  171  361  350  220  113   15   21   35   18   20   20   28   45   18
[109]   17   26   28   28   31   31   42   76   76   26   59   65  101  116   18   20   20   30
[127]   44   44   82   82  128   37   46   46   80   88   88   33   46   29   53   53   41   86
[145]   95  107  117  119  120   48  126  266  270  426  151  267  603   19   21   26   35   41
[163]   47   62   78   80   80  142  281  190   21   25   67   24   24   64   25   20   29   43
[181]   53   19   22   31   41   47   99   67   81  149  183  275  382   56  182  227  341  360
[199]  919  978   24   24   24   24   37   50   41   47   25
>
```

- fitted(lmreg) #predicted values

The fitted method shows the predicted values obtained from the linear regression method and are as shown below:

1	2	3	4	5	6	7	8
185.239246	251.267379	205.698667	161.059929	123.860981	296.836091	342.404802	455.861595
9	10	11	12	13	14	15	16
592.567730	1064.994374	36.443453	38.303400	86.662033	129.440823	10.404189	33.653531
17	18	19	20	21	22	23	24
18.773952	27.143715	29.933637	112.701297	29.003663	31.793584	57.832848	71.782453
25	26	27	28	29	30	31	32
22.493847	65.272638	31.793584	26.213742	72.712427	26.213742	255.917248	343.334776
33	34	35	36	37	38	39	40
30.863610	59.692795	99.681665	194.538983	19.703926	28.073689	67.132585	25.283768
41	42	43	44	45	46	47	48
34.583505	38.303400	49.463084	56.902874	68.062559	68.062559	17.843978	19.703926
49	50	51	52	53	54	55	56
38.303400	58.762822	23.423821	23.423821	129.440823	34.583505	25.283768	56.902874
57	58	59	60	61	62	63	64
67.132585	12.264136	14.124084	19.703926	15.984031	21.563873	34.583505	135.020666
65	66	67	68	69	70	71	72
135.020666	241.967642	16.914005	25.283768	30.863610	30.863610	58.762822	60.622769
73	74	75	76	77	78	79	80
21.563873	34.583505	42.023295	47.603137	42.953269	50.393058	34.583505	79.222243
81	82	83	84	85	86	87	88
15.984031	36.443453	36.443453	15.984031	21.563873	28.073689	38.303400	33.653531
89	90	91	92	93	94	95	96
125.720929	62.482716	132.230744	176.869482	21.563873	123.860981	221.508220	433.542226
97	98	99	100	101	102	103	104
433.542226	258.707169	173.149588	6.684294	23.423821	42.953269	7.614268	13.194110
105	106	107	108	109	110	111	112
15.984031	30.863610	30.863610	11.334162	11.334162	17.843978	21.563873	35.513479
113	114	115	116	117	118	119	120
38.303400	32.723558	47.603137	71.782453	62.482716	23.423821	46.673163	62.482716
121	122	123	124	125	126	127	128
94.101822	124.790955	12.264136	17.843978	19.703926	26.213742	42.953269	53.182979
129	130	131	132	133	134	135	136
66.202611	75.502348	127.580876	15.984031	25.283768	30.863610	42.953269	51.323032
137	138	139	140	141	142	143	144
61.552743	29.003663	47.603137	38.303400	58.762822	56.902874	47.603137	62.482716
145	146	147	148	149	150	151	152
81.082191	69.922506	87.592007	104.331533	134.090692	98.751691	200.118825	258.707169
153	154	155	156	157	158	159	160
345.194724	475.391043	200.118825	304.275880	475.391043	8.544241	12.264136	16.914005
161	162	163	164	165	166	167	168
20.633900	23.423821	32.723558	40.163347	43.883242	48.533111	108.981402	94.101822
169	170	171	172	173	174	175	176
131.300771	198.258878	24.353794	29.003663	39.233374	24.353794	47.603137	47.603137
177	178	179	180	181	182	183	184
29.003663	30.863610	36.443453	56.902874	102.471586	6.684294	11.334162	21.563873
185	186	187	188	189	190	191	192
31.793584	55.042927	122.001034	70.852480	106.191481	175.939509	161.989903	231.737931
193	194	195	196	197	198	199	200
377.743803	66.202611	107.121454	194.538983	286.606380	370.304014	852.030394	1070.574216
201	202	203	204	205	206	207	208
12.264136	14.124084	17.843978	20.633900	40.163347	43.883242	49.463084	63.412690
209							
42.953269							

We can see comparing both the outputs that the predicted values are nearer to the actual values and hence we can predict values for one attribute from other attribute.

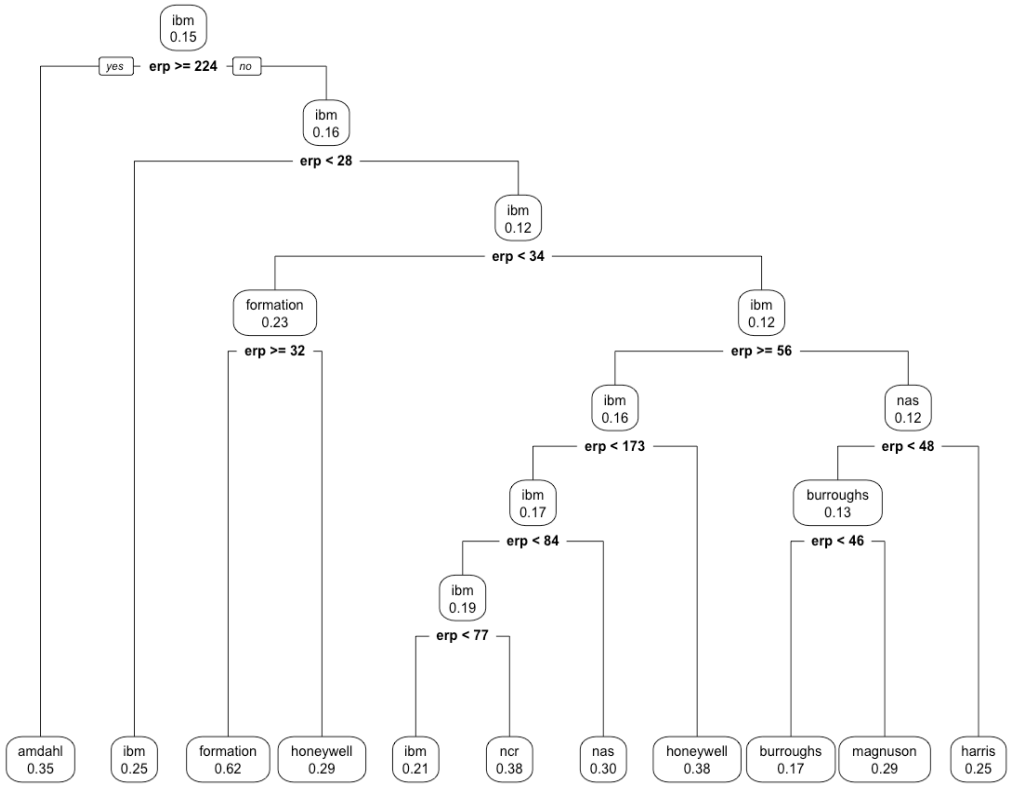
Decision Tree

```
#import packages
```

- library(rpart)
- library(rpart.plot)
- library(party)
- str(machine) #show details of each attribute

```
'data.frame':   209 obs. of  10 variables:
 $ vendor: Factor w/ 30 levels "adviser","amdahl",..: 1 2 2 2 2 2 2 2 2 ...
 $ model : Factor w/ 209 levels "100","1100/61-h1",..: 30 63 64 65 66 67 75 76 77 78 ...
 $ 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 ...
```

- mach.tree<-rpart(vendor~erp,machine)
- rpart.plot(mach.tree, extra = 8) #plot the decision tree



The above graph shows that the probability of ibm vendor among all others is 15%. If the value of erp is greater than or equal to 224 it also contains the value of Amdahl whose probability of the fitted value in the table is 35% and so on.