VIETNAM NATIONAL UNIVERSITY

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

PROBABILITY AND STATISTICS

PROJECT

ANALYZING RELATIVE CPU PERFORMANCE FOR DIFFERENT MACHINES

Lecturer:	Dr. Nguyễn Tiến Dũng	
Class:	CC01	
Group:	3	
Students:	Lê Tuấn Hưng	2052508
	Hồng Huy Mẫn	2052593
	Mai Hữu Nghĩa	2052612
	Đình Xuân Phú	2052650
	Đỗ Hữu Thanh Thiện	2053453

Ho Chi Minh City – 2022

Table of contents

1. Introduction

1.1 Purpose

There are several situations that we need to evaluating the performance of central processing units (CPUs), including computer system selection for both original acquisitions and upgrades, computer system configuration, and computer system design. In these context, analytical and approximate models are particularly useful to perform a benchmark.

Probably the most effective solution found tat date to the problem of CPU performance evaluation is the computation of *relative performance* data, which measure the performance of CPUs in terms of a base machine

Briefly, the methodology was to collect data on the characteristics and *relative performance* of a large number of CPUs. These data were then subjected to extensive statistical analysis in which those characteristics that significantly affect relative performance were identified and isolated, primarily by factor analysis; and correlation data were examined as the basis for selecting transformations that improved performance of the statistics. A linear prediction model was then developed, using stepwise multi-variate linear regression, and its predictive accuracy
evaluated.

1.2 What is relative performance?

Relative performance is measured by the performance of CPUs in terms of a base machine - the IBM 370/158 model 3, initialized at 45. Relative performance of other CPUs is calibrated with it on the basis of vendor claims, user experience, and information supplied by independent consultants; ideally, this information is based on extensive benchmarks.

1.3 THE RELATIVE PERFORMANCE PREDICTION MODEL

The linear model for relative performance prediction is as follows

$$SQRPERF = A_0 + A_1(MAVG) + A_2(CACH) + A_3(CHCAP)$$

Where

$SQRPERF$: square root of relative performance

$MAVG$: Average memory size, calculated by $(MMIN + MMAX)/2 * 10^{-3}$

$CACH$: Cache memory size, calculated by $CACHE * 10^{-1}$

$CHCAP$: Channel capacity = $[\, INT \, [\frac{CHMIN+CHMAX}{2} + 1] \times MCYT^{-1}$

2. Data Interpretation

2.1 Data Description:

Our data set has 12 categories of data:

- Vendor name: The company supply the different models of CPU
- Model name: Name of the model of central processing unit (CPU)
- MCYT: Machine cycle time, unit: Nanoseconds (ns)
- MMIN: Minium main memory, unit: Kilobytes (KB)
- MMAX: Maximum main memory, unit: Kilobytes (KB)
- CACH: Cache memory size, which equals $CACHE \times 10^{-1}$, unit: 10 Kilobytes
- CACHE: Cache memory size, unit: Kilobytes (KB)
- CHMIN: Minimum number of I/O channels, unit: Channels
- CHMAX: Maximum number of I/O channels, unit: Channels
- PRP: Published relative performance
- SQRPERF: Square root of relative performance
- ERP: Estimated relative performance
- MAVG: Average memory size, which equals $(MMIN + MMAX)/2 \times 10^{-3}$
- CHCAP: Channel capacity, which equals CHAVG * SPEED * 10, unit: Channel executions per 10 nanoseconds

Table 2.1: CPU Specs Dataset

vendor name	model name	MCYT	MMIN	MMAX	CACH	CACHE	CHMIN	CHMAX	PRP	SQRPERF	ERP	MAVG	CHCAP
amd	470v/7	29	8000	32000	3.2	32	8	32	269	1.640.121.947	253	20	724.137.931
amd	470v/7a	29	8000	32000	3.2	32	8	32	220	1.483.239.697	253	20	724.137.931
amd	470v/7b	29	8000	32000	3.2	32	8	32	172	1.311.487.705	253	20	724.137.931
amd	470v/7c	29	8000	16000	3.2	32	8	16	132	1.148.912.529	132	12	4.482.758.621
amd	470v/b	26	8000	32000	6.4	64	8	32	318	178.325.545	290	20	8.076.923.077
amd	580-5840	23	16000	32000	6.4	64	16	32	367	1.915.724.406	381	24	1.086.956.522
amd	580-5850	23	16000	32000	6.4	64	16	32	489	2.211.334.439	381	24	1.086.956.522
amd	580-5860	23	16000	64000	6.4	64	16	32	636	2.521.904.043	749	40	1.086.956.522
amd	580-5880	23	32000	64000	12.8	128	32	64	1144	3.382.306.905	1238	48	2.130.434.783
bur	b1955	167	524	2000	0.8	8	4	15	19	4.358.898.944	23	1.262	628.742.515
bur	b2900	143	512	5000	0	0	7	32	28	5.291.502.622	29	2.756	1.433.566.434
bur	b2925	143	1000	2000	0	0	5	16	31	5.567.764.363	22	1.5	804.195.804
bur	b4955	110	5000	5000	14.2	142	8	64	120	1.095.445.115	124	5	3.363.636.364
bur	b5900	143	1500	6300	0	0	5	32	30	5.477.225.575	35	3.9	1.363.636.364
bur	b5920	143	3100	6200	0	0	5	20	33	5.744.562.647	39	4.65	944.055.944
bur	b6900	143	2300	6200	0	0	6	64	61	7.810.249.676	40	4.25	2.517.482.517
bur	b6925	110	3100	6200	0	0	6	64	76	8.717.797.887	45	4.65	3.272.727.273
c.r.d	68/10-80	320	128	6000	0	0	1	12	23	4.795.831.523	28	3.064	234.375
c.r.d	universe:2203t	320	512	2000	0.4	4	1	3	69	8.306.623.863	21	1.256	9.375
c.r.d	universe:68	320	256	6000	0	0	1	6	33	5.744.562.647	28	3.128	140.625
c.r.d	universe:68/05	320	256	3000	0.4	4	1	3	27	5.196.152.423	22	1.628	9.375
c.r.d	universe:68/137	320	512	5000	0.4	4	1	5	77	8.774.964.387	28	2.756	125
c.r.d	universe:68/37	320	256	5000	0.4	4	1	6	27	5.196.152.423	27	2.628	140.625
c.r.d	1636-1	50	1000	4000	0.8	8	3	5	26	5.099.019.514	30	2.5	1
c.r.d	1636-10	50	1000	8000	0.8	8	3	5	36	6	41	4.5	1
c.r.d	1641-1	50	2000	16000	0.8	8	3	5	40	632.455.532	74	9	1
c.r.d	1641-11	50	2000	16000	0.8	8	3	6	52	7.211.102.551	74	9	1.1
c.r.d	1651-1	50	2000	16000	0.8	8	3	6	60	7.745.966.692	74	9	1.1
cdc	cyber:170/750	25	1310	2620	13.1	131	12	24	274	1.655.294.536	102	1.965	7.6
cdc	cyber:170/760	25	1310	2620	13.1	131	12	24	368	1.918.332.609	102	1.965	7.6
cdc	cyber:170/815	50	2620	10480	3	30	12	24	32	5.656.854.249	74	6.55	3.8
cdc	cyber:170/825	50	2620	10480	3	30	12	24	63	7.937.253.933	74	6.55	3.8
cdc	cyber:170/835	56	5240	20970	3	30	12	24	106	1.029.563.014	138	13.105	3.392.857.143
cdc	cyber:170/845	64	5240	20970	3	30	12	24	208	144.222.051	136	13.105	296.875

cdc	omega:480-i	50	500	2000	0.8	8	1	4	20	4.472.135.955	23	1.25	0.7
cdc	omega:480-ii	50	1000	4000	0.8	8	1	5	29	5.385.164.807	29	2.5	0.8
cdc	omega:480-iii	50	2000	8000	0.8	8	1	5	71	8.426.149.773	44	5	0.8
dec	decsys:10:1091	133	1000	12000	0.9	9	3	12	72	8.485.281.374	54	6.5	639.097.744
ibm	4341-12	185	2000	16000	1.6	16	1	6	76	8.717.797.887	76	9	243.243.243
ibm	4341-2	180	2000	16000	1.6	16	1	6	66	8.124.038.405	76	9	0.25
ibm	4341-9	225	1000	4000	0.2	2	3	6	24	4.898.979.486	26	2.5	244.444.444
ibm	4361-4	25	2000	12000	0.8	8	1	4	49	7	59	7	1.4
ibm	4361-5	25	2000	12000	1.6	16	3	5	66	8.124.038.405	65	7	2
ibm	4381-1	17	4000	16000	0.8	8	6	12	100	10	101	10	5.882.352.941
ibm	4381-2	17	4000	16000	3.2	32	6	12	133	1.153.256.259	116	10	5.882.352.941
ibm	8130-a	1500	768	1000	0	0	0	0	12	3.464.101.615	18	884	6.666.667
ibm	8130-b	1500	768	2000	0	0	0	0	18	4.242.640.687	20	1.384	6.666.667
ibm	8140	800	768	2000	0	0	0	0	20	4.472.135.955	20	1.384	125
ibm	4436	50	2000	4000	0	0	3	6	27	5.196.152.423	30	3	1.1
ibm	4443	50	2000	8000	0.8	8	3	6	45	6.708.203.932	44	5	1.1
ibm	4445	50	2000	8000	0.8	8	1	6	56	7.483.314.774	44	5	0.9
ibm	4446	50	2000	16000	2.4	24	1	6	70	8.366.600.265	82	9	0.9
ibm	4460	50	2000	16000	2.4	24	1	6	80	894.427.191	82	9	0.9
nas	as/30	100	1000	8000	0	0	2	6	16	4	37	4.5	0.5
nas	as/31	100	1000	8000	2.4	24	2	6	26	5.099.019.514	46	4.5	0.5
nas	as/32	100	1000	8000	2.4	24	3	6	32	5.656.854.249	46	4.5	0.55
nas	as/42	50	2000	16000	1.2	12	3	16	45	6.708.203.932	80	9	2.1
nas	as/43	50	2000	16000	2.4	24	6	16	54	7.348.469.228	88	9	2.4
nas	as/44	50	2000	16000	2.4	24	6	16	65	8.062.257.748	88	9	2.4
nas	as/3000	115	2000	8000	1.6	16	1	3	50	7.071.067.812	46	5	260.869.565
nas	as/3000-n	115	2000	4000	0.2	2	1	5	40	632.455.532	29	3	347.826.087
nas	as/5000	92	2000	8000	3.2	32	1	6	62	7.874.007.874	53	5	489.130.435
nas	as/5000-e	92	2000	8000	3.2	32	1	6	60	7.745.966.692	53	5	489.130.435
nas	as/5000-n	92	2000	8000	0.4	4	1	6	50	7.071.067.812	41	5	489.130.435
nas	as/6130	75	4000	16000	1.6	16	1	6	66	8.124.038.405	86	10	0.6
nas	as/6150	60	4000	16000	3.2	32	1	6	86	9.273.618.495	95	10	0.75
nas	as/6620	60	2000	16000	6.4	64	5	8	74	8.602.325.267	107	9	1.25
nas	as/6630	60	4000	16000	6.4	64	5	8	93	9.643.650.761	117	10	1.25
nas	as/6650	50	4000	16000	6.4	64	5	10	111	1.053.565.375	119	10	1.7
nas	as/7000	72	4000	16000	6.4	64	8	16	143	1.195.826.074	120	10	1.805.555.556
nas	as/7000-n	72	2000	8000	1.6	16	6	8	105	1.024.695.077	48	5	1.111.111.111
nas	as/8040	40	8000	16000	3.2	32	8	16	214	1.462.873.884	126	12	3.25
nas	as/8050	40	8000	32000	6.4	64	8	24	277	1.664.331.698	266	20	4.25
hwell	dps:8/62	140	2000	32000	3.2	32	1	54	189	1.374.772.708	181	17	2.035.714.286
nas	as/8060	35	8000	32000	6.4	64	8	24	370	1.923.538.406	270	20	4.857.142.857
nas	as/9000-dpc	38	16000	32000	12.8	128	16	32	510	2.258.317.958	426	24	6.578.947.368
nas	as/9000-n	48	4000	24000	3.2	32	8	24	214	1.462.873.884	151	14	3.541.666.667
nas	as/9040	38	8000	32000	6.4	64	8	24	326	1.805.547.009	267	20	4.473.684.211
nas	as/9060	30	16000	32000	25.6	256	16	24	510	2.258.317.958	603	24	7
ncr	v8535:ii	112	1000	1000	0	0	1	4	8	2.828.427.125	19	1	3.125
ncr	v8545:ii	84	1000	2000	0	0	1	6	12	3.464.101.615	21	1.5	535.714.286
ncr	v8555:ii	56	1000	4000	0	0	1	6	17	4.123.105.626	26	2.5	803.571.429
ncr	v8565:ii	56	2000	6000	0	0	1	8	21	4.582.575.695	35	4	982.142.857
ncr	v8565:ii-e	56	2000	8000	0	0	1	8	24	4.898.979.486	41	5	982.142.857
ncr	v8575:ii	56	4000	8000	0	0	1	8	34	5.830.951.895	47	6	982.142.857
ncr	v8585:ii	56	4000	12000	0	0	1	8	42	6.480.740.698	62	8	982.142.857
ncr	v8595:ii	56	4000	16000	0	0	1	8	46	6.782.329.983	78	10	982.142.857
ncr	v8635	38	4000	8000	3.2	32	16	32	51	7.141.428.429	80	6	6.578.947.368
ncr	v8650	38	4000	8000	3.2	32	16	32	116	1.077.032.961	80	6	6.578.947.368
ncr	v8655	38	8000	16000	6.4	64	4	8	100	10	142	12	1.842.105.263
ncr	v8665	38	8000	24000	16	160	4	8	140	1.183.215.957	281	16	1.842.105.263
ncr	v8670	38	4000	16000	12.8	128	16	32	212	1.456.021.978	190	10	6.578.947.368
spe	1100/61-h1	116	2000	8000	3.2	32	5	28	70	8.366.600.265	56	5	150.862.069
spe	1100/81	50	2000	32000	2.4	24	6	26	114	1.067.707.825	182	17	3.4
spe	1100/82	50	2000	32000	4.8	48	26	52	208	144.222.051	227	17	8
spe	1100/83	50	2000	32000	11.2	112	52	104	307	1.752.141.547	341	17	15.8
spe	1100/84	50	4000	32000	11.2	112	52	104	397	1.992.485.885	360	18	15.8
spe	1100/93	30	8000	64000	9.6	96	12	176	915	3.024.896.692	919	36	3.166.666.667
spe	1100/94	30	8000	64000	12.8	128	12	176	1150	3.391.164.992	978	36	3.166.666.667
spe	80/3	180	262	4000	0	0	1	3	12	3.464.101.615	24	2.131	166.666.667
spe	80/4	180	512	4000	0	0	1	3	14	3.741.657.387	24	2.256	166.666.667
spe	80/5	180	262	4000	0	0	1	3	18	4.242.640.687	24	2.131	166.666.667
spe	80/6	180	512	4000	0	0	1	3	21	4.582.575.695	24	2.256	166.666.667
spe	80/8	124	1000	8000	0	0	1	8	42	6.480.740.698	37	4.5	443.548.387
spe	90/80-model-3	98	1000	8000	3.2	32	2	8	46	6.782.329.983	50	4.5	612.244.898

The dataset above includes the values in each variable relating to the CPU performance, including 12 vendors, its different types of models and 12 specifications of each CPU.
We decided to use this data for the report, analyzing the variables and the relations to one another. Such as checking their similarities, differences, and finding a conclusion if the relative performance prediction equation can be applied to our data.

Table 2.2: Frequency table of table 2.1

Vendor name	Frequency
amd	9
bur	8
c.r.d	11
cdc	9
dec	6
dg	12
hp	7
hwell	20
ibm	37
nas	25
ncr	13
spe	13
Total	**170**

2.2 Import Data in RStudio

After importing 'cpu_time.csv' into RStudio, we will receive information from the program about which column is the factor or the numerical data with the following code:

	ï..vendor.name	model.name	MCYT	MMIN	MMAX	CACH	CACHE	CHMIN	CHMAX	PRP	SQRPERF	ERP	MAVG	CHCAP
1	amd	470v/7	29	8000	32000	3.2	32	8	32	269	16.401219	253	20.000	7.241379310
2	amd	470v/7a	29	8000	32000	3.2	32	8	32	220	14.832397	253	20.000	7.241379310
3	amd	470v/7b	29	8000	32000	3.2	32	8	32	172	13.114877	253	20.000	7.241379310
4	amd	470v/7c	29	8000	16000	3.2	32	8	16	132	11.489125	132	12.000	4.482758621
5	amd	470v/b	26	8000	32000	6.4	64	8	32	318	17.832555	290	20.000	8.076923077
6	amd	580-5840	23	16000	32000	6.4	64	16	32	367	19.157244	381	24.000	10.869565220
7	amd	580-5850	23	16000	32000	6.4	64	16	32	489	22.113344	381	24.000	10.869565220
8	amd	580-5860	23	16000	64000	6.4	64	16	32	636	25.219040	749	40.000	10.869565220
9	amd	580-5880	23	32000	64000	12.8	128	32	64	1144	33.823069	1238	48.000	21.304347830
10	bur	b1955	167	524	2000	0.8	8	4	15	19	4.358899	23	1.262	0.628742515
11	bur	b2900	143	512	5000	0.0	0	7	32	28	5.291503	29	2.756	1.433566434
12	bur	b2925	143	1000	2000	0.0	0	5	16	31	5.567764	22	1.500	0.804195804
13	bur	b4955	110	5000	5000	14.2	142	8	64	120	10.954451	124	5.000	3.363636364
14	bur	b5900	143	1500	6300	0.0	0	5	32	30	5.477226	35	3.900	1.363636364
15	bur	b5920	143	3100	6200	0.0	0	5	20	33	5.744563	39	4.650	0.944055944
16	bur	b6900	143	2300	6200	0.0	0	6	64	61	7.810250	40	4.250	2.517482517
17	bur	b6925	110	3100	6200	0.0	0	6	64	76	8.717798	45	4.650	3.272727273
18	c.r.d	68/10-80	320	128	6000	0.0	0	1	12	23	4.795832	28	3.064	0.234375000
19	c.r.d	universe:2203t	320	512	2000	0.4	4	1	3	69	8.306624	21	1.256	0.093750000
20	c.r.d	universe:68	320	256	6000	0.0	0	1	6	33	5.744563	28	3.128	0.140625000

Showing 1 to 20 of 170 entries, 14 total columns

```
cpu <- read.csv("C:/Users/EmChes/OneDrive - wtpvf/Desktop/zdfg/cpu_time.csv")
cpu <- read.csv("C:/Users/EmChes/OneDrive - wtpvf/Desktop/zdfg/cpu_time.csv", header = TRUE, colClasses =
c("factor", "factor", "numeric","numeric", "numeric", "numeric", "numeric", "numeric","numeric", "numeric",
"numeric", "numeric", "numeric", "numeric"), fileEncoding='UTF-8-BOM')
summary(cpu)
```

After running:

```
 vendor.name      model.name        MCYT             MMIN             MMAX             CACH
ibm     :37   100        :  1   Min.   :  17.0   Min.   :   96   Min.   :  512   Min.   : 0.000
nas     :25   1100/61-h1:  1   1st Qu.:  50.0   1st Qu.: 1000   1st Qu.: 4000   1st Qu.: 0.000
hwell   :20   1100/81    :  1   Median : 105.0   Median : 2000   Median : 8000   Median : 0.800
ncr     :13   1100/82    :  1   Mean   : 212.4   Mean   : 3003   Mean   :12370   Mean   : 2.354
spe     :13   1100/83    :  1   3rd Qu.: 225.0   3rd Qu.: 4000   3rd Qu.:16000   3rd Qu.: 3.200
dg      :12   1100/84    :  1   Max.   :1500.0   Max.   :32000   Max.   :64000   Max.   :25.600
(Other):50   (Other)    :164
    CACHE            CHMIN            CHMAX             PRP              SQRPERF            ERP
Min.   :  0.00   Min.   : 0.000   Min.   :  0.00   Min.   :   6.0   Min.   : 2.449   Min.   :  15.0
1st Qu.:  0.00   1st Qu.: 1.000   1st Qu.:  5.00   1st Qu.:  26.0   1st Qu.: 5.099   1st Qu.:  28.0
Median :  8.00   Median : 3.000   Median :  8.00   Median :  47.5   Median : 6.891   Median :  45.0
Mean   : 23.54   Mean   : 4.894   Mean   : 18.18   Mean   : 109.8   Mean   : 8.787   Mean   : 104.8
3rd Qu.: 32.00   3rd Qu.: 6.000   3rd Qu.: 24.00   3rd Qu.: 105.8   3rd Qu.:10.283   3rd Qu.: 101.8
Max.   :256.00   Max.   :52.000   Max.   :176.00   Max.   :1150.0   Max.   :33.912   Max.   :1238.0

      MAVG             CHCAP
Min.   : 0.304   Min.   : 0.00667
1st Qu.: 2.532   1st Qu.: 0.23659
Median : 5.000   Median : 0.82500
Mean   : 7.686   Mean   : 2.44560
3rd Qu.:10.000   3rd Qu.: 2.40000
Max.   :48.000   Max.   :31.66667
```

Using this code gives us the overview of the figures in each variable which will be used later on when coming to analyzing and modeling our data.

3. Analysis & Models

3.1. Histogram Plot

A histogram is used to summarize discrete or continuous data, it helps provide us a visual interpretation of numerical data by showing the number of data points that fall within a specified range of values. Simplifying it by giving us the figure summary of data distribution in each variable for this report. This can help us see the median, outliers or gaps in our data as well.

Histogram plot is used to describe the frequency of an outcome. Here, we will use histogram to study the pattern of certain variables in different CPUs with the following codes:

3.1.1. For Cache Memory in kilobytes

```
par(mfrow=c(2,3))
hist(cpu$CACH[cpu$vendor.name=="amd"], xlab="amdahl" ,main="")
hist(cpu$CACH[cpu$vendor.name=="bur"], xlab="burroughs" ,main="")
hist(cpu$CACH[cpu$vendor.name=="c.r.d"], xlab="c.r.d" ,main="")
hist(cpu$CACH[cpu$vendor.name=="cdc"], xlab="cdc" ,main="")
hist(cpu$CACH[cpu$vendor.name=="dec"], xlab="dec" ,main="")
hist(cpu$CACH[cpu$vendor.name=="dg"], xlab="dg" ,main="")
hist(cpu$CACH[cpu$vendor.name=="hwell"], xlab="honeywell" ,main="")
hist(cpu$CACH[cpu$vendor.name=="hp"], xlab="hp" ,main="")
hist(cpu$CACH[cpu$vendor.name=="ibm"], xlab="ibm" ,main="")
hist(cpu$CACH[cpu$vendor.name=="nas"], xlab="nas" ,main="")
hist(cpu$CACH[cpu$vendor.name=="ncr"], xlab="ncr" ,main="")
hist(cpu$CACH[cpu$vendor.name=="spe"], xlab="sperry" ,main="")
```

Result:

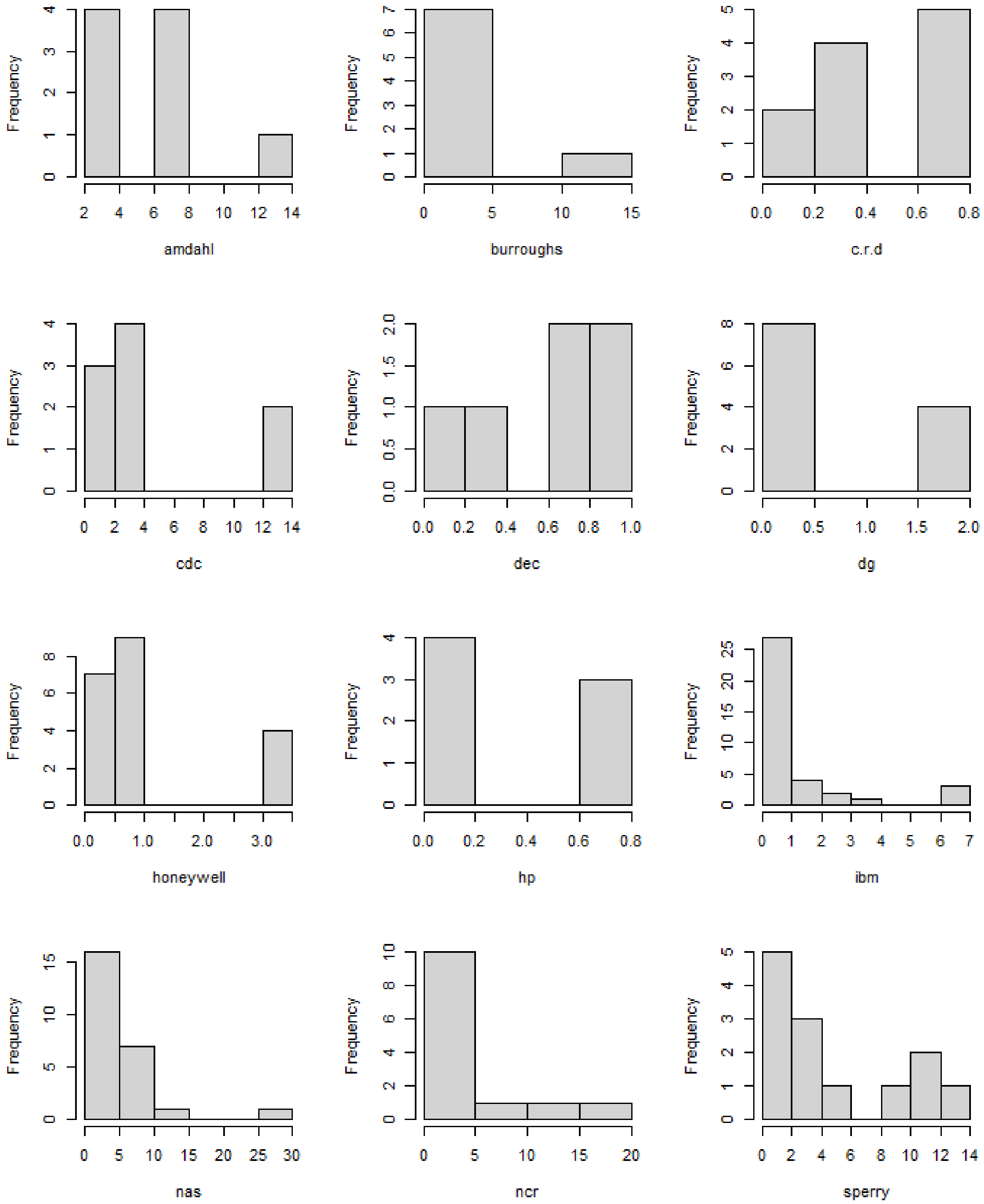

Figure 3.1.1: Cache Memory Frequency

3.1.2. For Average Memory Size

```
par(mfrow=c(2,3))
hist(cpu$MAVG[cpu$vendor.name=="amd"], xlab="amdahl" ,main="")
hist(cpu$MAVG[cpu$vendor.name=="bur"], xlab="burroughs" ,main="")
hist(cpu$MAVG[cpu$vendor.name=="c.r.d"], xlab="c.r.d" ,main="")
hist(cpu$MAVG[cpu$vendor.name=="cdc"], xlab="cdc" ,main="")
hist(cpu$MAVG[cpu$vendor.name=="dec"], xlab="dec" ,main="")
hist(cpu$MAVG[cpu$vendor.name=="dg"], xlab="dg" ,main="")
hist(cpu$MAVG[cpu$vendor.name=="hwell"], xlab="honeywell" ,main="")
hist(cpu$MAVG[cpu$vendor.name=="hp"], xlab="hp" ,main="")
hist(cpu$MAVG[cpu$vendor.name=="ibm"], xlab="ibm" ,main="")
hist(cpu$MAVG[cpu$vendor.name=="nas"], xlab="nas" ,main="")
hist(cpu$MAVG[cpu$vendor.name=="ncr"], xlab="ncr" ,main="")
hist(cpu$MAVG[cpu$vendor.name=="spe"], xlab="sperry" ,main="")
```

Result:

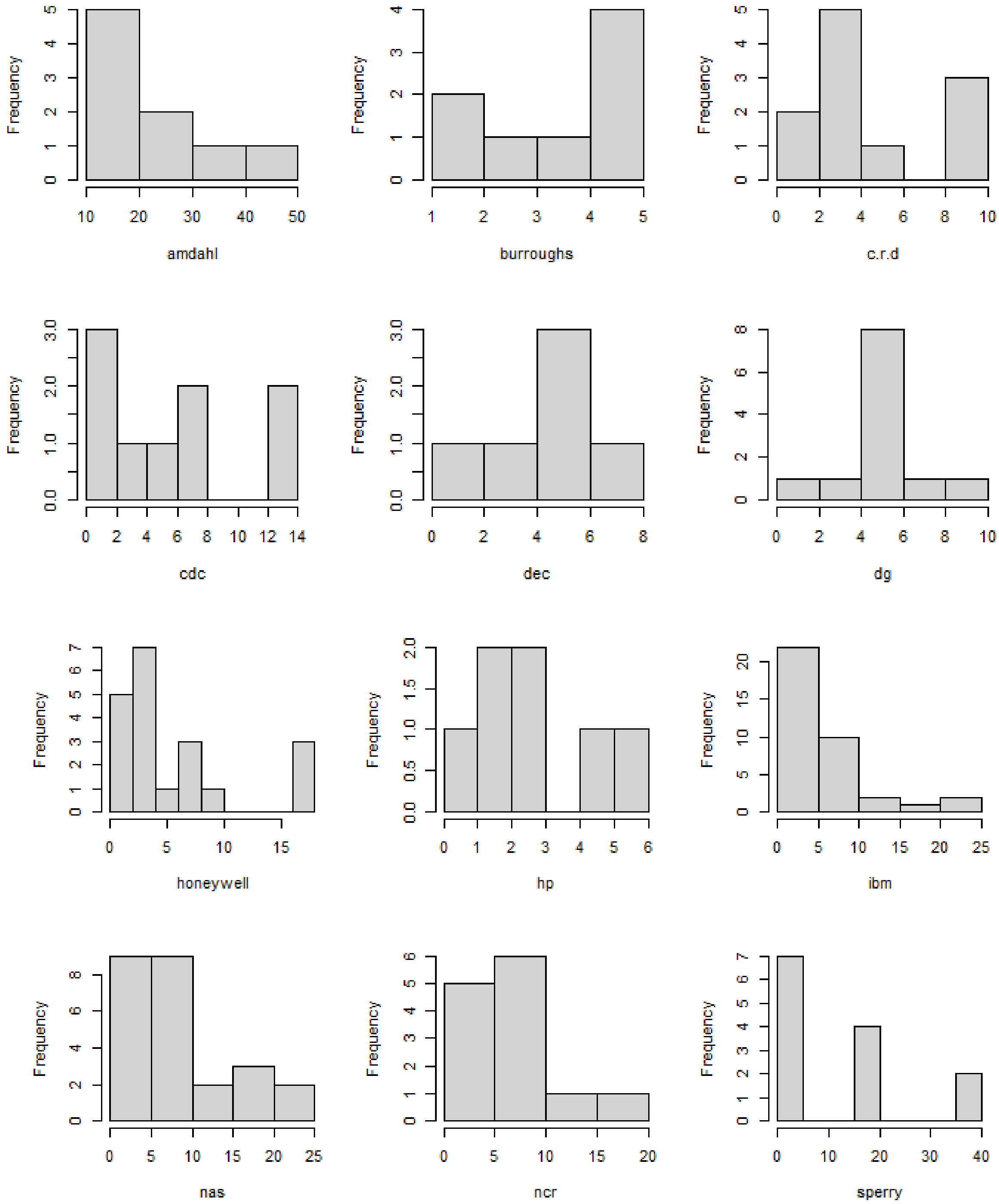

Figure 3.1.2: Average Memory Size Frequency

3.1.3. For Channel Capacity

```
par(mfrow=c(2,3))
hist(cpu$CHCAP[cpu$vendor.name=="amd"], xlab="amdahl" ,main="")
hist(cpu$CHCAP[cpu$vendor.name=="bur"], xlab="burroughs" ,main="")
hist(cpu$CHCAP[cpu$vendor.name=="c.r.d"], xlab="c.r.d" ,main="")
hist(cpu$CHCAP[cpu$vendor.name=="cdc"], xlab="cdc" ,main="")
hist(cpu$CHCAP[cpu$vendor.name=="dec"], xlab="dec" ,main="")
hist(cpu$CHCAP[cpu$vendor.name=="dg"], xlab="dg" ,main="")
hist(cpu$CHCAP[cpu$vendor.name=="hwell"], xlab="honeywell" ,main="")
hist(cpu$CHCAP[cpu$vendor.name=="hp"], xlab="hp" ,main="")
hist(cpu$CHCAP[cpu$vendor.name=="ibm"], xlab="ibm" ,main="")
hist(cpu$CHCAP[cpu$vendor.name=="nas"], xlab="nas" ,main="")
hist(cpu$CHCAP[cpu$vendor.name=="ncr"], xlab="ncr" ,main="")
hist(cpu$CHCAP[cpu$vendor.name=="spe"], xlab="sperry" ,main="")
```

Result:

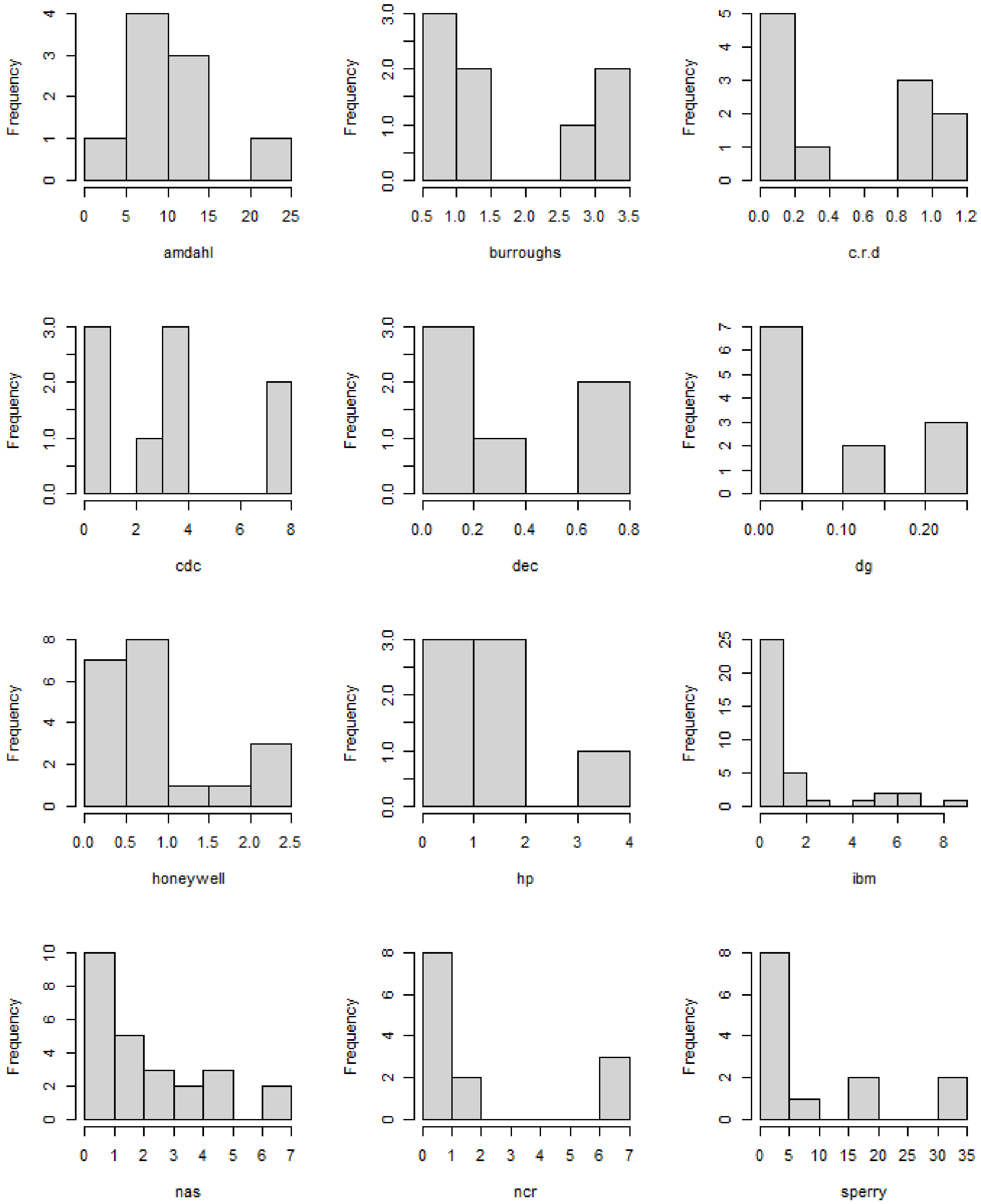

Figure 3.1.3: Channel Capacity Frequency

3.1.4. For Published Relative Performance

```
par(mfrow=c(2,3))
hist(cpu$PRP[cpu$vendor.name=="amd"], xlab="amdahl" ,main="")
hist(cpu$PRP[cpu$vendor.name=="bur"], xlab="burroughs" ,main="")
hist(cpu$PRP[cpu$vendor.name=="c.r.d"], xlab="c.r.d" ,main="")
hist(cpu$PRP[cpu$vendor.name=="cdc"], xlab="cdc" ,main="")
hist(cpu$PRP[cpu$vendor.name=="dec"], xlab="dec" ,main="")
hist(cpu$PRP[cpu$vendor.name=="dg"], xlab="dg" ,main="")
hist(cpu$PRP[cpu$vendor.name=="hwell"], xlab="honeywell" ,main="")
hist(cpu$PRP[cpu$vendor.name=="hp"], xlab="hp" ,main="")
hist(cpu$PRP[cpu$vendor.name=="ibm"], xlab="ibm" ,main="")
hist(cpu$PRP[cpu$vendor.name=="nas"], xlab="nas" ,main="")
hist(cpu$PRP[cpu$vendor.name=="ncr"], xlab="ncr" ,main="")
hist(cpu$PRP[cpu$vendor.name=="spe"], xlab="sperry" ,main="")
```

Result:

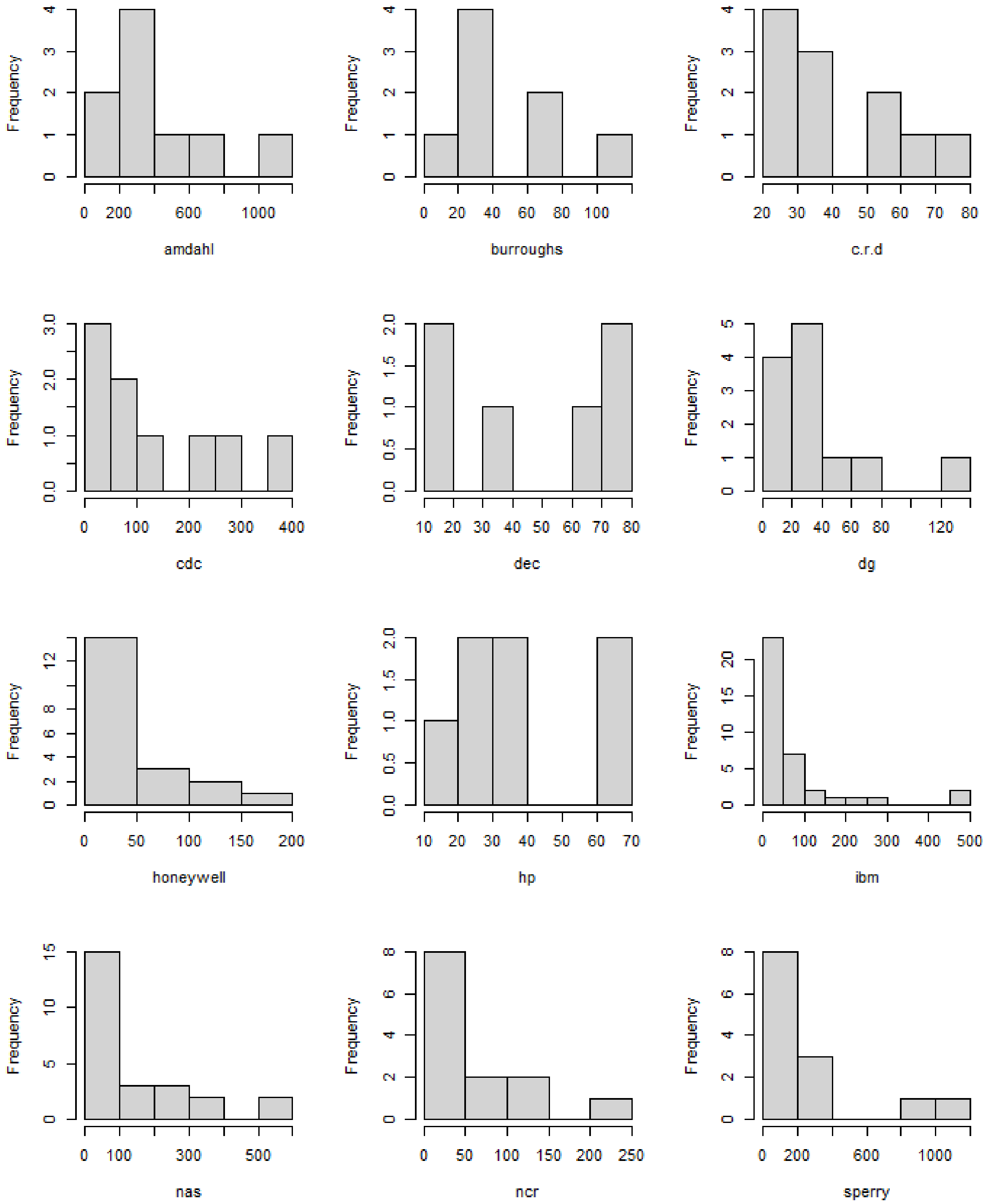

Figure 3.1.4: Published Relative Performance Frequency

3.2. Box Plot

Box plots provide a visual summary of analyzing by quickly identifying the mean values, dispersion of the data set as well as signs of skewness. It will help us show the dispersion and outliers within a data set. An outlier is an observation that is numerically distant from the rest of the data. When reviewing a box plot, an outlier is defined as a data point that is located outside the whiskers of the box plot.

Box plot is a method for description by mapping the group data numbers through their private section. A typical box plot will look like this:

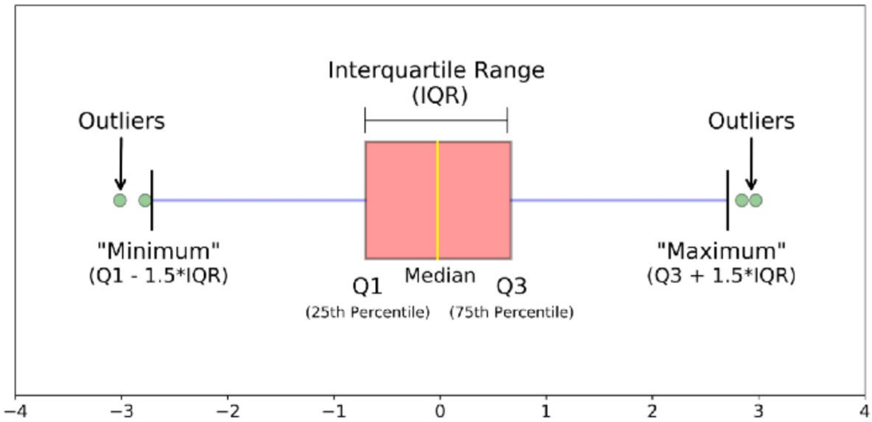

Figure 3.2: Box Plot Model for Normal Distribution

Boxplots are a standardized way of displaying the distribution of data based on a five number summary **("minimum", first quartile (Q1), median, third quartile (Q3),** and **"maximum")**:

- *Median* (Q2/50th *Percentile*): the middle value of the dataset.
- *First Quartile* (Q1/25th *Percentile*): the middle number between the smallest number (not the "minimum") and the median of the dataset.
- *Third Quartile* (Q3/75th *Percentile*): the middle value between the median and the highest value (not the "maximum") of the dataset.
- *Interquartile Range* (IQR): 25th to the 75th percentile.
- *Whiskers* (shown in blue)
- *Outliers* (shown as green circles)
- *"Maximum"*: Q3 + 1.5*IQR
- *"Minimum"*: Q1 - 1.5*IQR

3.2.1 Box Plot Model:

We will now draw the box plot to demonstrate the variables of each CPU in our data set:

```
boxplot(CACH~vendor.name, data=cpu , col=blues9)
boxplot(PRP~vendor.name, data=cpu , col=blues9)
boxplot(MAVG~vendor.name, data=cpu , col=blues9)
boxplot(CHCAP~vendor.name, data=cpu , col=blues9)
```

Result:

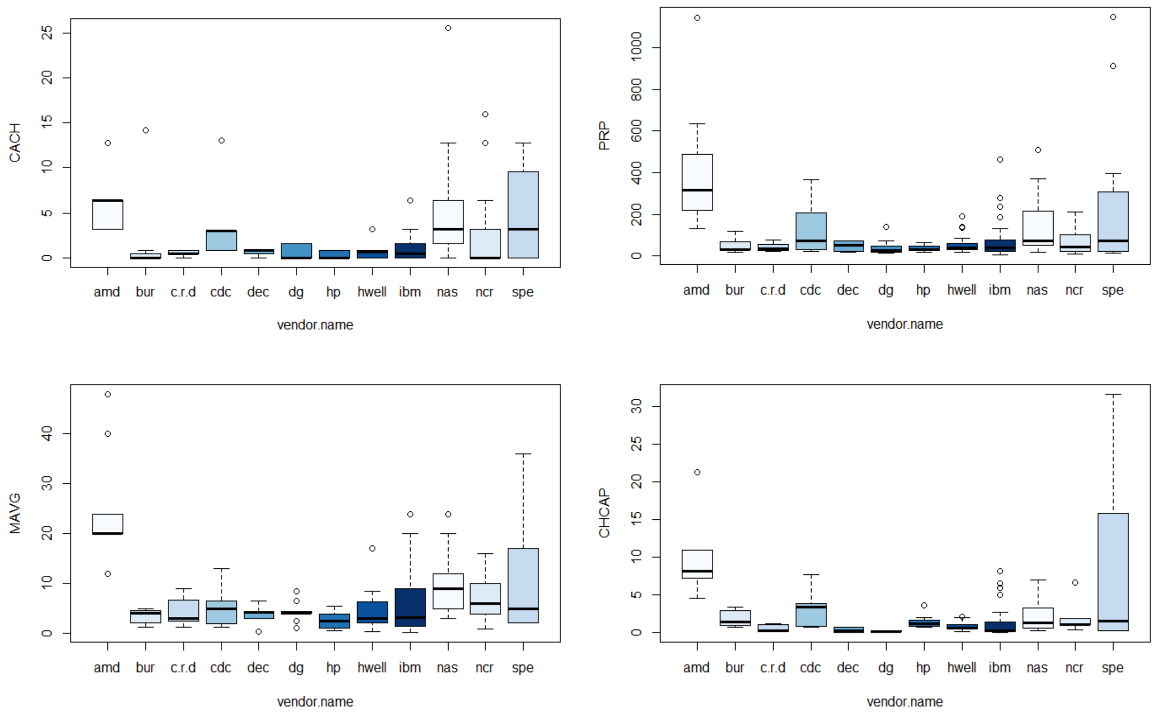

Figure 3.2.1: Box Plot for CACH, PRP, MAVG and CHCAP

3.3. Pairs Plot

A pairs plot allows us to see both distribution of single variables and relationships between two variables.

Pair plots are a great method to identify trends for follow-up analysis as well. A pairs plot is a matrix of scatterplots that lets you understand the pairwise relationship between different variables in a dataset. In RStudio, pair plots can also be used to determine the pairwise correlation coefficients of the variables.

3.3.1 Pairs Plot Model:

We will now draw the pairs plot to demonstrate the variables of each CPU in our data set:

```
library(ggplot2)
library(GGally)
data <- data.frame(cpu$CACH, cpu$MAVG, cpu$CHCAP, cpu$PRP)
ggpairs(data = data, lower=list(continuous="smooth",
wrap=c(colour="blue")),
upper=list(wrap=list(corSize=6)), axisLabels='show')
```

Result:

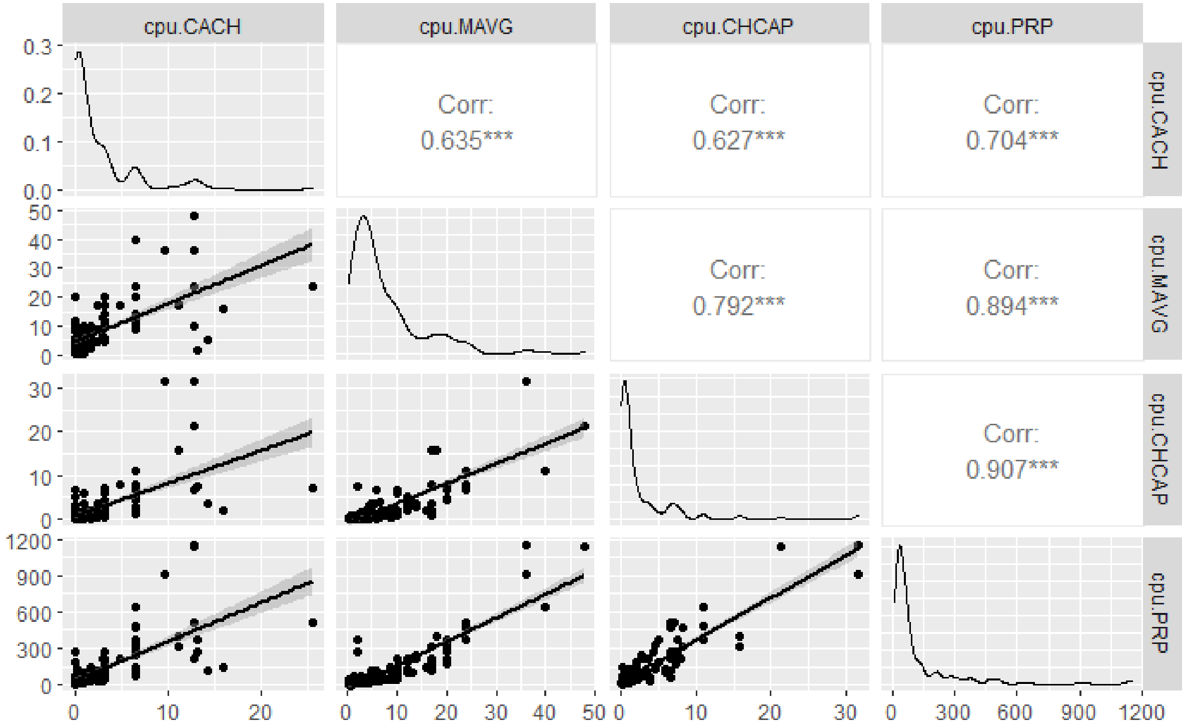

Figure 3.3.1: Pairs Plot and Correlation Coefficients

Note: *** means the p-value is in the range [0, 0.001]. These significance codes are displayed as a series of stars or a decimal point if the variables are statistically significant.

From the correlation coefficients, we can conclude that CACH, MAVG, CHCAP and PRP are strongly pairwise related to one another, as all of them are close to 1 or -1. This subject will be discussed further in the linear regression section.

4. Statistical Methods

4.1 ANOVA Test

In this experiment data, ANOVA is used to understand whether there is a statistically significant difference in the population mean …. resulted from many types of CPU. Researchers can conduct a one-way ANOVA using "Name of CPU" as the factor and the remaining 4 variables as the response.

```
CACH.aov= aov(CACH~vendor.name, data = cpu)
summary(CACH.aov)
```

```
             Df Sum Sq Mean Sq F value   Pr(>F)    
vendor.name  11  561.4   51.03   4.243 1.61e-05 ***
Residuals   158 1900.6   12.03                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MAVG.aov= aov(MAVG~vendor.name, data = cpu)
summary(MAVG.aov)
```

```
             Df Sum Sq Mean Sq F value   Pr(>F)
vendor.name  11   4173   379.4   9.533 4.61e-13 ***
Residuals   158   6288    39.8
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
CHCAP.aov= aov(CHCAP~vendor.name, data = cpu)
summary(CHCAP.aov)
```

```
             Df Sum Sq Mean Sq F value   Pr(>F)
vendor.name  11   1198  108.87   7.524 2.49e-10 ***
Residuals   158   2286   14.47
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
PRP.aov= aov(PRP~vendor.name, data = cpu)
summary(PRP.aov)
```

```
             Df  Sum Sq Mean Sq F value   Pr(>F)
vendor.name  11 1467582  133417   5.811 7.28e-08 ***
Residuals   158 3627857   22961
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Taking the F-value of the 4 ANOVA tests with a significance level of $\alpha =$

$$F_{CACH} = 4.243 > F_{0.02,11,158} = 2.15$$

$$F_{MAVG} = 4.243 > F_{0.02,11,158} = 2.15$$

$$F_{CHCAP} = 4.243 > F_{0.02,11,158} = 2.15$$

$$F_{PRP} = 4.243 > F_{0.02,11,158} = 2.15$$

We can safely conclude with a 98% confidence level that there is a significant difference between each treatment when it comes to CACH, MAVG, CHCAP, PRP.

4.2 Tukey's Test

Multiple Comparisons Using Tukey's Range Test (Tukey Honest Significant Difference)

After knowing that each treatment differs one another, we want to perform multiple comparisons on them by using Tukey's range test. First , we will observe these 6 types of cpu in terms of CACH, MAVG, CHCAP, PRP.

4.2.1 For CACH

```
TukeyHSD(CACH.aov)
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = CACH ~ vendor.name, data = cpu)

$vendor.name
                   diff          lwr          upr     p adj
bur-amd     -3.81388889  -9.40515001  1.777372230 0.5071430
c.r.d-amd   -5.17979798 -10.35168421 -0.007911749 0.0492665
cdc-amd     -1.17777778  -6.60209782  4.246542263 0.9998871
dec-amd     -5.05555556 -11.12012973  1.009018616 0.2052508
dg-amd      -5.15555556 -10.22954234 -0.081568768 0.0427434
hp-amd      -5.34603175 -11.14487379  0.452810297 0.1019585
hwell-amd   -4.73888889  -9.35753098 -0.120246801 0.0387222
ibm-amd     -4.58888889  -8.86558470 -0.312193076 0.0239011
nas-amd     -1.07288889  -5.54589778  3.400120006 0.9997021
ncr-amd     -2.48888889  -7.47853761  2.500759834 0.8853387
spe-amd     -1.19658120  -6.18622992  3.793067527 0.9997026
c.r.d-bur   -1.36590909  -6.71262399  3.980805805 0.9994564
cdc-bur      2.63611111  -2.95515001  8.227372230 0.9191701
dec-bur     -1.24166667  -7.45600596  4.972672629 0.9999506
dg-bur      -1.34166667  -6.59374196  3.910408629 0.9994566
hp-bur      -1.53214286  -7.48743647  4.423150757 0.9994190
hwell-bur   -0.92500000  -5.73860652  3.888606520 0.9999667
ibm-bur     -0.77500000  -5.26154434  3.711544335 0.9999887
nas-bur      2.74100000  -1.93305194  7.415051944 0.7286699
ncr-bur      1.32500000  -3.84564284  6.495642840 0.9994403
spe-bur      2.61730769  -2.55333515  7.787950532 0.8749132
cdc-c.r.d    4.00202020  -1.16986603  9.173906433 0.3074418
dec-c.r.d    0.12424242  -5.71563935  5.964124201 1.0000000
dg-c.r.d     0.02424242  -4.77893373  4.827418582 1.0000000
hp-c.r.d    -0.16623377  -5.72966152  5.397193983 1.0000000
hwell-c.r.d  0.44090909  -3.87846749  4.760285668 1.0000000
ibm-c.r.d    0.59090909  -3.36070910  4.542527286 0.9999975
nas-c.r.d    4.10690909  -0.05637902  8.270197197 0.0569121
ncr-c.r.d    2.69090909  -2.02308642  7.404904600 0.7612722
spe-c.r.d    3.98321678  -0.73077873  8.697212293 0.1888509
dec-cdc     -3.87777778  -9.94235195  2.186796394 0.6083607
dg-cdc      -3.97777778  -9.05176457  1.096209010 0.2882697
hp-cdc      -4.16825397  -9.96709601  1.630588074 0.4225349
hwell-cdc   -3.56111111  -8.17975320  1.057530976 0.3128015
```

ibm-cdc	-3.41111111	-7.68780692	0.865584702	0.2636436
nas-cdc	0.10488889	-4.36812001	4.577897784	1.0000000
ncr-cdc	-1.31111111	-6.30075983	3.678537612	0.9992914
spe-cdc	-0.01880342	-5.00845214	4.970845305	1.0000000
dg-dec	-0.10000000	-5.85336023	5.653360226	1.0000000
hp-dec	-0.29047619	-6.69222447	6.111272087	1.0000000
hwell-dec	0.31666667	-5.03942022	5.672753551	1.0000000
ibm-dec	0.46666667	-4.59751531	5.530848639	1.0000000
nas-dec	3.98266667	-1.24835814	9.213691476	0.3319034
ncr-dec	2.56666667	-3.11245354	8.245786877	0.9387707
spe-dec	3.85897436	-1.82014585	9.538094569	0.5133078
hp-dg	-0.19047619	-5.66301336	5.282060977	1.0000000
hwell-dg	0.41666667	-3.78499357	4.618326903	1.0000000
ibm-dg	0.56666667	-3.25592684	4.389260170	0.9999977
nas-dg	4.08266667	0.04163925	8.123694084	0.0452361
ncr-dg	2.66666667	-1.93970809	7.273041423	0.7446013
spe-dg	3.95897436	-0.64740040	8.565349115	0.1693279
hwell-hp	0.60714286	-4.44609134	5.660377055	0.9999997
ibm-hp	0.75714286	-3.98558296	5.499868677	0.9999950
nas-hp	4.27314286	-0.64733778	9.193623495	0.1579827
ncr-hp	2.85714286	-2.53729090	8.251576610	0.8381386
spe-hp	4.14945055	-1.24498320	9.543884302	0.3163480
ibm-hwell	0.15000000	-3.04354566	3.343545663	1.0000000
nas-hwell	3.66600000	0.21398386	7.118016136	0.0268281
ncr-hwell	2.25000000	-1.84941474	6.349414740	0.8041068
spe-hwell	3.54230769	-0.55710705	7.641722432	0.1633893
nas-ibm	3.51600000	0.53695859	6.495041414	0.0072151
ncr-ibm	2.10000000	-1.60991557	5.809915572	0.7709238
spe-ibm	3.39230769	-0.31760788	7.102223265	0.1087153
ncr-nas	-1.41600000	-5.35060990	2.518609898	0.9887648
spe-nas	-0.12369231	-4.05830221	3.810917591	1.0000000
spe-ncr	1.29230769	-3.22099939	5.805614779	0.9984345

We see that μ_{nas} - μ_{ibm}= 3.51 > 0 and p-value =0.007 < 0.05 so we can say that

$$\mu_{nas} > \mu_{ibm}$$

with μ_{spe} - μ_{hwell}= 3.542 > 0 and p-value = 0.1633 > 0.05, so we conclude that

$$\mu_{spe} = \mu_{hwell}$$

Overall we have the CACH order

Group A includes bur, hp, c.r.d, cdc, nas

Group B includes dg, hwell, ibm, ncr, spe, dec

$$\mu_{GroupA} > \mu_{GroupB}$$

We can apply this calculating method for MAVG, CHCAP, PRP

4.2.2 For MAVG

```
TukeyHSD(MAVG.aov)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = MAVG ~ vendor.name, data = cpu)

$vendor.name
                   diff         lwr         upr     p adj
bur-amd     -21.83733333 -32.0070981 -11.667569 0.0000000
c.r.d-amd   -20.92787879 -30.3348561 -11.520901 0.0000000
cdc-amd     -19.55666667 -29.4227877  -9.690546 0.0000000
dec-amd     -21.49600000 -32.5266586 -10.465341 0.0000001
dg-amd      -20.96166667 -30.1905778 -11.732756 0.0000000
hp-amd      -22.63561905 -33.1829461 -12.088292 0.0000000
hwell-amd   -19.79123333 -28.1919327 -11.390534 0.0000000
ibm-amd     -19.07003604 -26.8487803 -11.291292 0.0000000
nas-amd     -14.87333333 -23.0091452  -6.737522 0.0000006
ncr-amd     -18.56410256 -27.6396139  -9.488591 0.0000000
spe-amd     -12.73533333 -21.8108447  -3.659822 0.0004131
c.r.d-bur     0.90945455  -8.8155130  10.634422 1.0000000
cdc-bur       2.28066667  -7.8890981  12.450431 0.9998449
dec-bur       0.34133333 -10.9617283  11.644395 1.0000000
dg-bur        0.87566667  -8.6771639  10.428497 1.0000000
hp-bur       -0.79828571 -11.6301775  10.033606 1.0000000
hwell-bur     2.04610000  -6.7092139  10.801414 0.9997683
ibm-bur       2.76729730  -5.3931337  10.927728 0.9931185
nas-bur       6.96400000  -1.5374825  15.465483 0.2275462
ncr-bur       3.27323077  -6.1314850  12.677946 0.9914555
spe-bur       9.10200000  -0.3027157  18.506716 0.0677491
cdc-c.r.d     1.37121212  -8.0357652  10.778189 0.9999981
dec-c.r.d    -0.56812121 -11.1900941  10.053852 1.0000000
dg-c.r.d     -0.03378788  -8.7701303   8.702555 1.0000000
hp-c.r.d     -1.70774026 -11.8268798   8.411399 0.9999911
hwell-c.r.d   1.13664545  -6.7197295   8.993020 0.9999982
ibm-c.r.d     1.85784275  -5.3296283   9.045314 0.9993927
nas-c.r.d     6.05454545  -1.5179252  13.627016 0.2602485
ncr-c.r.d     2.36377622  -6.2103584  10.937911 0.9988931
spe-c.r.d     8.19254545  -0.3815892  16.766680 0.0758210
dec-cdc      -1.93933333 -12.9699920   9.091325 0.9999865
dg-cdc       -1.40500000 -10.6339111   7.823911 0.9999970
hp-cdc       -3.07895238 -13.6262794   7.468375 0.9981355
```

```
hwell-cdc    -0.23456667  -8.6352660   8.166133 1.0000000
ibm-cdc       0.48663063  -7.2921136   8.265375 1.0000000
nas-cdc       4.68333333  -3.4524785  12.819145 0.7513958
ncr-cdc       0.99256410  -8.0829473  10.068075 0.9999999
spe-cdc       6.82133333  -2.2541780  15.896845 0.3517307
dg-dec        0.53433333  -9.9302683  10.998935 1.0000000
hp-dec       -1.13961905 -12.7835528  10.504315 1.0000000
hwell-dec     1.70476667  -8.0372473  11.446781 0.9999871
ibm-dec       2.42596396  -6.7851135  11.637041 0.9992759
nas-dec       6.62266667  -2.8918759  16.137209 0.4745077
ncr-dec       2.93189744  -7.3976714  13.261466 0.9985544
spe-dec       8.76066667  -1.5689022  19.090236 0.1844875
hp-dg        -1.67395238 -11.6277740   8.279869 0.9999915
hwell-dg      1.17043333  -6.4718311   8.812698 0.9999968
ibm-dg        1.89163063  -5.0611616   8.844423 0.9990203
nas-dg        6.08833333  -1.2617613  13.438428 0.2132150
ncr-dg        2.39756410  -5.9808226  10.775951 0.9984430
spe-dg        8.22633333  -0.1520534  16.604720 0.0594420
hwell-hp      2.84438571  -6.3467792  12.035551 0.9968645
ibm-hp        3.56558301  -5.0608082  12.191974 0.9672899
nas-hp        7.76228571  -1.1874180  16.711989 0.1593615
ncr-hp        4.07151648  -5.7402453  13.883278 0.9663375
spe-hp        9.90028571   0.0885239  19.712048 0.0458066
ibm-hwell     0.72119730  -5.0874401   6.529835 0.9999996
nas-hwell     4.91790000  -1.3608610  11.196661 0.2895601
ncr-hwell     1.22713077  -6.2291627   8.683424 0.9999932
spe-hwell     7.05590000  -0.4003934  14.512193 0.0823550
nas-ibm       4.19670270  -1.2217798   9.615185 0.3060713
ncr-ibm       0.50593347  -6.2419125   7.253779 1.0000000
spe-ibm       6.33470270  -0.4131432  13.082549 0.0880443
ncr-nas      -3.69076923 -10.8473045   3.465766 0.8606249
spe-nas       2.13800000  -5.0185352   9.294535 0.9977057
spe-ncr       5.82876923  -2.3803397  14.037878 0.4423440
```
We can have the MAVG order
Group A includes ncr, nas, spe
Group B includes hp, bur, ibm, cdc, c.r.d, dec, hwell, hp

$$\mu'_{GroupA} = \mu'_{GroupB}$$

4.2.3 For CHCAP

```
TukeyHSD(CHCAP.aov)
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = CHCAP ~ vendor.name, data = cpu)

$vendor.name
                  diff         lwr        upr     p adj
bur-amd     -8.00864606 -14.1407728 -1.8765193 0.0015209
c.r.d-amd   -9.25164009 -14.9238241 -3.5794561 0.0000146
cdc-amd     -6.30391733 -12.2529541 -0.3548805 0.0274883
dec-amd     -9.51172176 -16.1629471 -2.8604964 0.0002844
dg-amd      -9.70103809 -15.2658525 -4.1362237 0.0000025
hp-amd      -8.31869908 -14.6784869 -1.9589112 0.0014819
hwell-amd   -8.95519980 -14.0206221 -3.8897775 0.0000016
ibm-amd     -8.36508764 -13.0554858 -3.6746895 0.0000013
nas-amd     -7.68188367 -12.5875851 -2.7761822 0.0000394
ncr-amd     -7.49322433 -12.9655423 -2.0209064 0.0006573
spe-amd     -1.37161961  -6.8439376  4.1006983 0.9995462
c.r.d-bur   -1.24299404  -7.1069186  4.6209306 0.9999110
cdc-bur      1.70472873  -4.4273981  7.8368555 0.9988047
dec-bur     -1.50307570  -8.3185536  5.3124022 0.9998684
dg-bur      -1.69239204  -7.4525222  4.0677381 0.9980238
hp-bur      -0.31005302  -6.8414267  6.2213206 1.0000000
hwell-bur   -0.94655374  -6.2258002  4.3326927 0.9999835
ibm-bur     -0.35644158  -5.2769878  4.5641046 1.0000000
nas-bur      0.32676239  -4.7994298  5.4529546 1.0000000
ncr-bur      0.51542172  -5.1553986  6.1862421 1.0000000
spe-bur      6.63702644   0.9662061 12.3078468 0.0080978
cdc-c.r.d    2.94772276  -2.7244613  8.6199068 0.8544235
dec-c.r.d   -0.26008166  -6.6648792  6.1447159 1.0000000
dg-c.r.d    -0.44939800  -5.7172051  4.8184091 1.0000000
hp-c.r.d     0.93294102  -5.1686600  7.0345420 0.9999968
hwell-c.r.d  0.29644030  -4.4407673  5.0336479 1.0000000
ibm-c.r.d    0.88655246  -3.4473220  5.2204269 0.9999375
nas-c.r.d    1.56975643  -2.9962636  6.1357765 0.9922835
ncr-c.r.d    1.75841576  -3.4115839  6.9284154 0.9929453
spe-c.r.d    7.88002048   2.7100208 13.0500202 0.0000734
dec-cdc     -3.20780443  -9.8590298  3.4434209 0.9066882
dg-cdc      -3.39712076  -8.9619351  2.1676936 0.6755443
hp-cdc      -2.01478175  -8.3745696  4.3450061 0.9961526
hwell-cdc   -2.65128247  -7.7167048  2.4141399 0.8484454
```

ibm-cdc	-2.06117031	-6.7515685	2.6292279	0.9494268
nas-cdc	-1.37796634	-6.2836678	3.5277351	0.9986858
ncr-cdc	-1.18930700	-6.6616250	4.2830109	0.9998861
spe-cdc	4.93229772	-0.5400202	10.4046157	0.1212980
dg-dec	-0.18931633	-6.4992227	6.1205901	1.0000000
hp-dec	1.19302268	-5.8279930	8.2140384	0.9999905
hwell-dec	0.55652196	-5.3176812	6.4307251	1.0000000
ibm-dec	1.14663412	-4.4074270	6.7006952	0.9999315
nas-dec	1.82983809	-3.9072052	7.5668814	0.9959199
ncr-dec	2.01849742	-4.2099874	8.2469823	0.9953184
spe-dec	8.14010214	1.9116173	14.3685870	0.0015032
hp-dg	1.38233902	-4.6195792	7.3842572	0.9997989
hwell-dg	0.74583829	-3.8622658	5.3539424	0.9999942
ibm-dg	1.33595046	-2.8564182	5.5283191	0.9959518
nas-dg	2.01915442	-2.4127782	6.4510870	0.9353910
ncr-dg	2.20781376	-2.8441546	7.2597821	0.9513288
spe-dg	8.32941848	3.2774502	13.3813868	0.0000111
hwell-hp	-0.63650072	-6.1785550	4.9055536	0.9999998
ibm-hp	-0.04638856	-5.2478977	5.1551206	1.0000000
nas-hp	0.63681541	-4.7596435	6.0332744	0.9999998
ncr-hp	0.82547474	-5.0907848	6.7417342	0.9999988
spe-hp	6.94707946	1.0308200	12.8633390	0.0077417
ibm-hwell	0.59011216	-2.9123583	4.0925826	0.9999913
nas-hwell	1.27331613	-2.5126277	5.0592600	0.9935794
ncr-hwell	1.46197547	-3.0339925	5.9579434	0.9951802
spe-hwell	7.58358019	3.0876122	12.0795482	0.0000061
nas-ibm	0.68320397	-2.5840124	3.9504203	0.9999222
ncr-ibm	0.87186330	-3.1969276	4.9406543	0.9999009
spe-ibm	6.99346802	2.9246771	11.0622590	0.0000036
ncr-nas	0.18865934	-4.1265615	4.5038802	1.0000000
spe-nas	6.31026406	1.9950432	10.6254849	0.0001801
spe-ncr	6.12160472	1.1717069	11.0715025	0.0036555

We have the CHCAP order :
Group A includes cdc, spe
Group B includes nas, ncr, bur, dg, cdc, c.r.d, hwell, ibm, hp

$$\mu''_{GroupA} = \mu''_{GroupB}$$

4.2.4 For PRP

```
TukeyHSD(PRP.aov)
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = PRP ~ vendor.name, data = cpu)

$vendor.name
                   diff          lwr        upr     p adj
bur-amd     -366.583333 -610.867316 -122.29935 0.0001036
c.r.d-amd   -373.606061 -599.567421 -147.64470 0.0000103
cdc-amd     -286.222222 -523.212496  -49.23195 0.0052026
dec-amd     -369.000000 -633.963181 -104.03682 0.0004781
dg-amd      -377.750000 -599.434102 -156.06590 0.0000046
hp-amd      -379.904762 -633.258021 -126.55150 0.0001053
hwell-amd   -359.633333 -561.423280 -157.84339 0.0000013
ibm-amd     -335.225225 -522.075430 -148.37502 0.0000011
nas-amd     -272.373333 -467.800519  -76.94615 0.0004710
ncr-amd     -353.025641 -571.024986 -135.02630 0.0000174
spe-amd     -161.410256 -379.409602   56.58909 0.3752556
c.r.d-bur     -7.022727 -240.622407  226.57695 1.0000000
cdc-bur       80.361111 -163.922872  324.64509 0.9946779
dec-bur       -2.416667 -273.923134  269.08980 1.0000000
dg-bur       -11.166667 -240.631512  218.29818 1.0000000
hp-bur       -13.321429 -273.510107  246.86725 1.0000000
hwell-bur      6.950000 -203.358005  217.25801 1.0000000
ibm-bur       31.358108 -164.660446  227.37666 0.9999949
nas-bur       94.210000 -110.000821  298.42082 0.9298058
ncr-bur       13.557692 -212.349344  239.46473 1.0000000
spe-bur      205.173077  -20.733959  431.08011 0.1145190
cdc-c.r.d     87.383838 -138.577522  313.34520 0.9801600
dec-c.r.d      4.606061 -250.540238  259.75236 1.0000000
dg-c.r.d      -4.143939 -213.996239  205.70836 1.0000000
hp-c.r.d      -6.298701 -249.366635  236.76923 1.0000000
hwell-c.r.d   13.972727 -174.742214  202.68767 1.0000000
ibm-c.r.d     38.380835 -134.266625  211.02830 0.9998576
nas-c.r.d    101.232727  -80.662659  283.12811 0.7895754
ncr-c.r.d     20.580420 -185.375549  226.53639 1.0000000
spe-c.r.d    212.195804    6.239836  418.15177 0.0371049
dec-cdc      -82.777778 -347.740959  182.18540 0.9965932
dg-cdc       -91.527778 -313.211880  130.15632 0.9675549
```

```
hp-cdc       -93.682540 -347.035799  159.67072 0.9860342
hwell-cdc    -73.411111 -275.201058  128.37884 0.9877393
ibm-cdc      -49.003003 -235.853208  137.84720 0.9993041
nas-cdc       13.848889 -181.578297  209.27608 1.0000000
ncr-cdc      -66.803419 -284.802764  151.19593 0.9971249
spe-cdc      124.811966  -93.187379  342.81131 0.7578017
dg-dec        -8.750000 -260.116144  242.61614 1.0000000
hp-dec       -10.904762 -290.599187  268.78966 1.0000000
hwell-dec      9.366667 -224.642479  243.37581 1.0000000
ibm-dec       33.774775 -187.480952  255.03050 0.9999969
nas-dec       96.626667 -131.918477  325.17181 0.9614704
ncr-dec       15.974359 -232.148215  264.09693 1.0000000
spe-dec      207.589744  -40.532831  455.71232 0.2007939
hp-dg         -2.154762 -241.251657  236.94213 1.0000000
hwell-dg      18.116667 -165.455210  201.68854 1.0000000
ibm-dg        42.524775 -124.485555  209.53510 0.9994731
nas-dg       105.376667  -71.177111  281.93044 0.7060100
ncr-dg        24.724359 -176.529624  225.97834 0.9999997
spe-dg       216.339744   15.085760  417.59373 0.0234110
hwell-hp      20.271429 -200.505986  241.04884 1.0000000
ibm-hp        44.679537 -162.531668  251.89074 0.9998946
nas-hp       107.531429 -107.445941  322.50880 0.8834183
ncr-hp        26.879121 -208.805411  262.56365 0.9999999
spe-hp       218.494505  -17.190027  454.17904 0.0975320
ibm-hwell     24.408108 -115.118921  163.93514 0.9999872
nas-hwell     87.260000  -63.559687  238.07969 0.7453046
ncr-hwell      6.607692 -172.497046  185.71243 1.0000000
spe-hwell    198.223077   19.118339  377.32782 0.0166260
nas-ibm       62.851892  -67.303378  193.00716 0.9059473
ncr-ibm      -17.800416 -179.887811  144.28698 0.9999999
spe-ibm      173.814969   11.727573  335.90236 0.0240644
ncr-nas      -80.652308 -252.556670   91.25205 0.9216733
spe-nas      110.963077  -60.941285  282.86744 0.5939936
spe-ncr      191.615385   -5.572442  388.80321 0.0653318
```

We can have the PRP order

Group A includes nas, ncr, spe

Group B includes c.r.d, dg, hwell, ibm, hp, cdc, bur, dec

$$\mu'''_{GroupA} = \mu'''_{GroupB}$$

Depending on the foundation of PC we want to build, we will choose the model that fits our purpose.

4.3 Multiple Linear Regression

We has construct a linear model for Relative Performance by predicting the square root of that

$$SQRPERF = A_0 + A_1(MAVG) + A_2(CACH) + A_3(CHCAP)$$

Where

$SQRPERF$: square root of relative performance

$MAVG$: Average memory size, calculated by $(MMIN + MMAX)/2 \times 10^{-3}$

$CACH$: Cache memory size, calculated by $CACHE \times 10^{-1}$

$CHCAP$: Channel capacity = $[\, INT\, [\frac{CHMIN+CHMAX}{2} + 1] \times MCYT^{-1}$

4.3.1 Calculating A_0, A_1, A_2 and A_3

```
model <- lm(SQRPERF ~ MAVG + CACH + CHCAP,data = cpu)
summary(model)
```

Result:

```
Call:
lm(formula = SQRPERF ~ MAVG + CACH + CHCAP, data = cpu)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.0887 -0.9953 -0.2823  0.7313  7.0308 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.02042    0.18286  21.987  < 2e-16 ***
MAVG         0.40140    0.02742  14.641  < 2e-16 ***
CACH         0.36634    0.04428   8.273 4.05e-14 ***
CHCAP        0.33478    0.04712   7.105 3.38e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.638 on 166 degrees of freedom
Multiple R-squared:  0.9196,	Adjusted R-squared:  0.9181 
F-statistic: 632.6 on 3 and 166 DF,  p-value: < 2.2e-16
```

From the figure, we all have low p-values for the slope coefficient ($< 10^{-7}$), which means that the coefficients of variables are meaningful or significantly affected to each other. Moreover, Intercept A_0 are also low.

Thus,

$$SQRPERF = 4.0204 + 0.4015 \times (MAVG) + 0.3663 \times (CACH) + 0.3348 \times (CHCAP)$$

Moreover, the median of Residual values is -0.2832, IQR = 0.8633. It shows that it is a good approximation for the equation for the Relative Performance.

Hence, we can conclude that the linear regression our group used is fit with the dataset.

5. Conclusion

As can be seen from our report by using models and statistic methods, we can conclude that there is a strictly ralation between elements of the CPU . We try to find the similarities for the variety of factor which will affect the relative performance, having the purpose to find for its alternatives when it comes to the CPU.

Using the observation, and analysis used in our report, it is not possible to reduce any of factors (cache, mavg, ...) as its fundamental factor to the performance of the CPU, being a heart of the computer hardware. With the use of statistics, this has helped us to have further in depth of the differences in CPU and how we should not easily decrease any factors of it.

6. References

[1]: Phillip Ein-Dor and Jacob Feldmesser. 1987. Attributes of the performance of central processing units: a relative performance prediction model. Commun. ACM 30, 4 (April 1987), 308–317. https://doi.org/10.1145/32232.32234

[2]: Galarnyk, M. (2020, July 6). Understanding Boxplots. Medium. Retrieved November 14, 2021, from https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51.

[3]: Koehrsen, W. (2018, April 6). Visualizing data with pairs plots in Python. Medium. Retrieved at November 14, 2021, from https://towardsdatascience.com/visualizing-data-with-pair-plots-in-python-f228cf529166.

[4]: Zach. (2020, August 11). How to create and interpret pairs plots in R. Statology. Retrieved November 14, 2021, from https://www.statology.org/pairs-plots-r/.

[5]: Histogram. Corporate Finance Institute. (2019, November 22). Retrieved November 14, 2021, from https://corporatefinanceinstitute.com/resources/excel/study/histogram/