Linear and Polynomial Regression Analysis

Student : A00209408

1. Linear Regression Analysis

1.1. Business Understanding

Business Objectives

The main objective is to analyse a range of quantifiable metrics to better predict the performance of a computer and give it a performance rating which helps compare it against other hardware available on the field.

Goals and success criteria

- Predict the performance metric of hardware computers
- Allow us to compare the metric against other similar ones in its bracket.

1.2. Data Understanding and Preparation

Describe Data

The data involves mostly concrete numerical values for performance in areas such as cycle time, min and max : channels and memory.

The data is alphabetically ordered by the producer so I will scramble some parts for my test set.

"Estimated.performance" contains values predicted by the original model too for comparison.

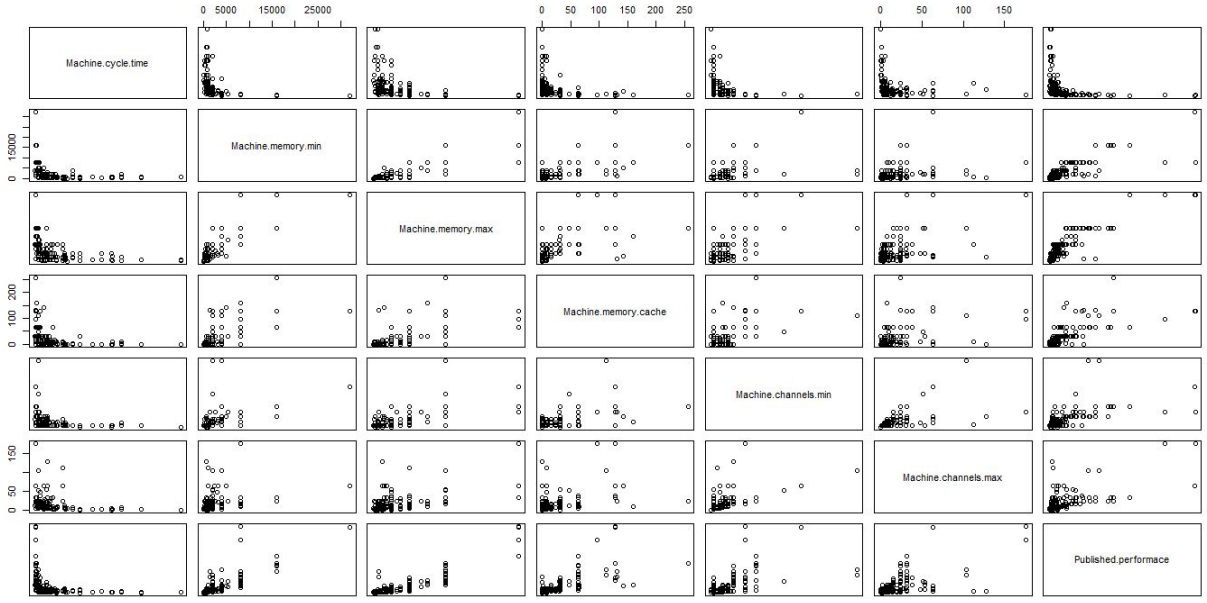
Example:

```
amdahl,470v/7,29,8000,32000,32,8,32,269,253
amdahl,470v/7a,29,8000,32000,32,8,32,220,253
amdahl,470v/7b,29,8000,32000,32,8,32,172,253
amdahl,470v/7c,29,8000,16000,32,8,16,132,132
amdahl,470v/b,26,8000,32000,64,8,32,318,290
amdahl,580-5840,23,16000,32000,64,16,32,367,381
amdahl,580-5850,23,16000,32000,64,16,32,489,381
amdahl,580-5860,23,16000,64000,64,16,32,636,749
amdahl,580-5880,23,32000,64000,128,32,64,1144,1238
```

Explore Data

While most values are at the lower side of the range with a few outliers we can see some nice linear correlation between **Published.Performance** and all the other parameters in the data set.

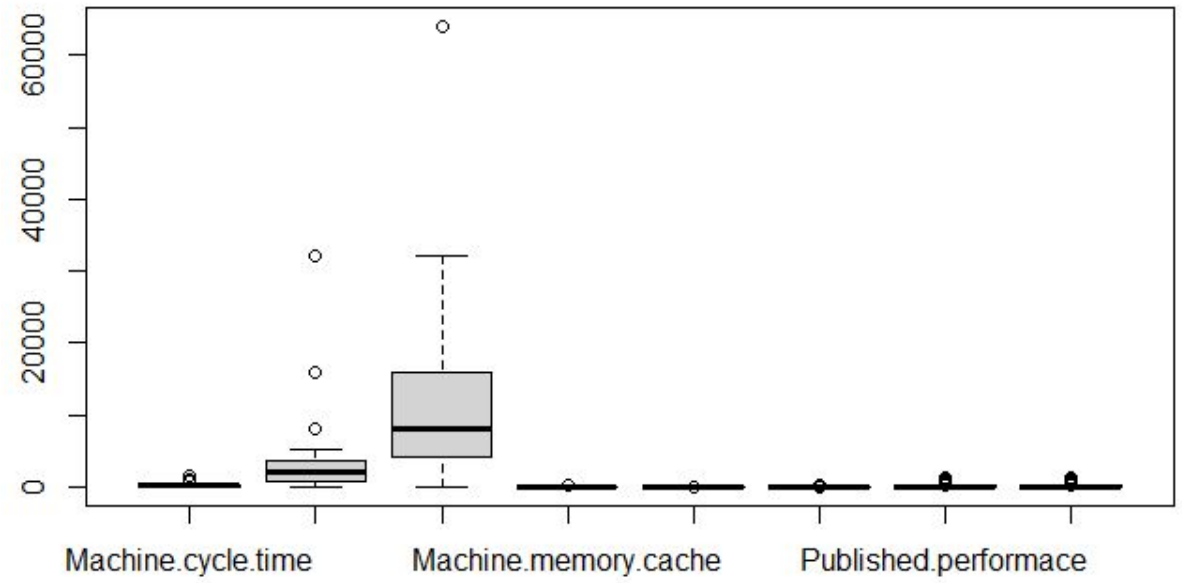
Machine.cycle.time can be ignored based on these graphs and keep the rest of the parameters



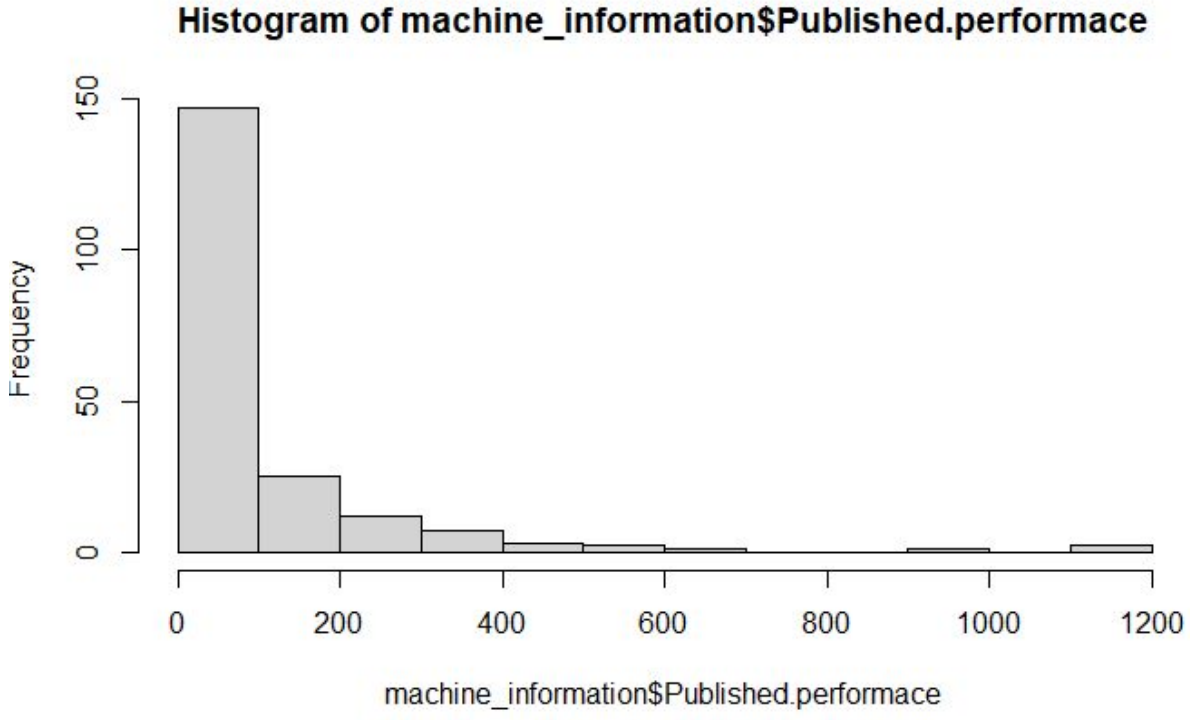
Strongest correlation towards the **Published.performance** can be seen with the **Machine.memory** values, the **Machine.cycle.time** "processing time" has a negative effect as it means it takes longer to complete a task.

```
Machine.cycle.time Machine.memory.min Machine.memory.max Machine.memory.cache
        -0.3056586          0.7966033          0.8702406           0.6965588
Machine.channels.min Machine.channels.max Published.performace Estimated.performance
           0.5971207            0.6225496                    1             0.9667149
```

There are also a few outliers in their respective columns which can be a little worrying and might need to be taken out.



There is also clearly a positive skew within the performance metric.



Both Skewness and Kurtosis values are extremely high compared to normal ranges (-1 : 1), this is not necessarily bad but does tell us we are dealing with non normally distributed data.

```
> skewness(machine_information$Published.performace)
[1] 3.916726
> kurtosis(machine_information$Published.performace)
[1] 21.96862
>
```

1.3. Modelling

Overview

The model should not be overly complicated as the total computation power is usually some amalgamation of the underlying performance metrics.

Published.performance is the Y variable and I have been playing around with different combinations of values to find the best fitting model.

Overall taking out any parameters whatsoever has proven to be detrimental to the end result.

```
machine_information_model <- lm(machine_information$Published.performace ~
                      #.
                      +Machine.cycle.time
                      #+Machine.memory.max
                      #+Machine.memory.min
                      +Machine.memory.max*Machine.memory.min
                      #+Machine.memory.max:Machine.memory.min
                      #+Machine.memory.cache
                      #+Machine.channels.max
                      #+Machine.channels.min
                      +Machine.channels.max*Machine.channels.min
                      #+Machine.channels.max:Machine.channels.min
                      -Estimated.performance #ignore the hardcoded predictions
                      ,data=machine_information)
```

Cycle time had to be taken out as it was having a negative effect and did not make sense.

```
Machine.cycle.time
        -0.3056586
```

Test Design

I have extracted a few scrambled holdout data rows to test for overfitting after the model is created. This data along with the previous **Published.performance** and **Estimated.performance** should provide enough for comparison.

1.4. Evaluation

Asses models:

Description of two of the main models with the highest R squared and lowest residual errors, the rest were scoring too low

I have noticed that the more parameters I take out the lower the prediction so I have left all of the parameters in the end.

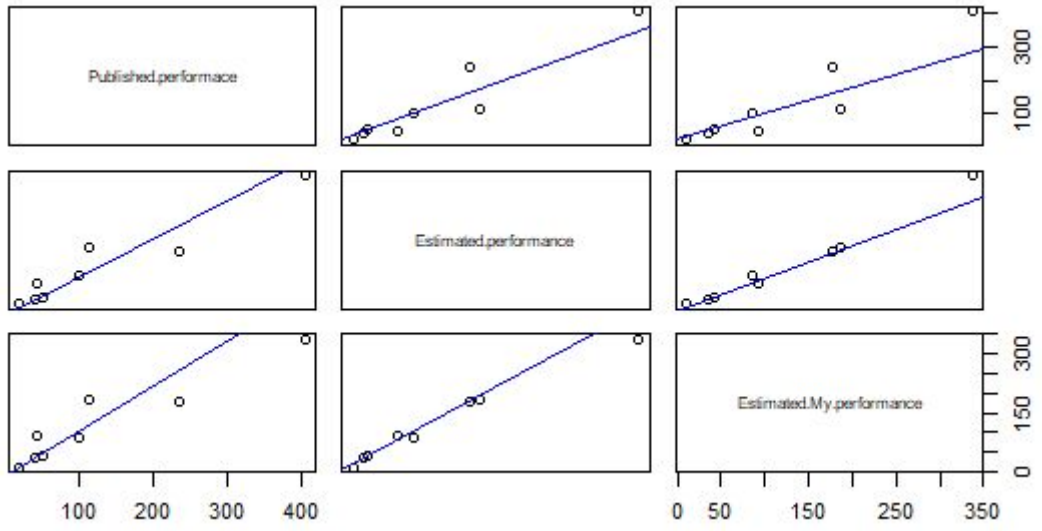
```
machine_information_model <- lm(machine_information$Published.performace ~
                      #.
                      +Machine.cycle.time
                      #+Machine.memory.max
                      #+Machine.memory.min
                      +Machine.memory.max*Machine.memory.min
                      #+Machine.memory.max:Machine.memory.min
                      +Machine.memory.cache
                      #+Machine.channels.max
                      #+Machine.channels.min
                      +Machine.channels.max*Machine.channels.min
                      #+Machine.channels.max:Machine.channels.min
                      -Estimated.performance #ignore the hardcoded predictions
                      ,data=machine_information)
```

Final model:

```
machine_information_model <- lm(machine_information$Published.performace ~
                      +Machine.memory.max*Machine.memory.min
                      -Machine.memory.min
                      +Machine.memory.cache
                      +Machine.channels.max*Machine.channels.min
                      -Estimated.performance #ignore the hardcoded predictions
                      ,data=machine_information)
```

Multiplication model:

This one might not work well for really large values as we see a sharp trend upwards in the line of best fit.



```
Residuals:
    Min      1Q  Median      3Q     Max
-184.96  -18.62    1.83   15.46  322.87

Coefficients:
                                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                                -4.007e+00  6.537e+00  -0.613  0.54058
Machine.memory.max                          3.789e-03  5.742e-04   6.600 3.88e-10 ***
Machine.memory.cache                        1.024e+00  1.307e-01   7.832 3.13e-13 ***
Machine.channels.max                        1.252e+00  2.165e-01   5.784 2.91e-08 ***
Machine.channels.min                       -3.508e+00  1.324e+00  -2.650  0.00872 **
Machine.memory.max:Machine.memory.min       4.015e-07  3.110e-08  12.909  < 2e-16 ***
Machine.channels.max:Machine.channels.min   3.491e-02  1.489e-02   2.344  0.02008 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 50.7 on 193 degrees of freedom
Multiple R-squared:  0.9055,    Adjusted R-squared:  0.9025
F-statistic: 308.1 on 6 and 193 DF,  p-value: < 2.2e-16
```

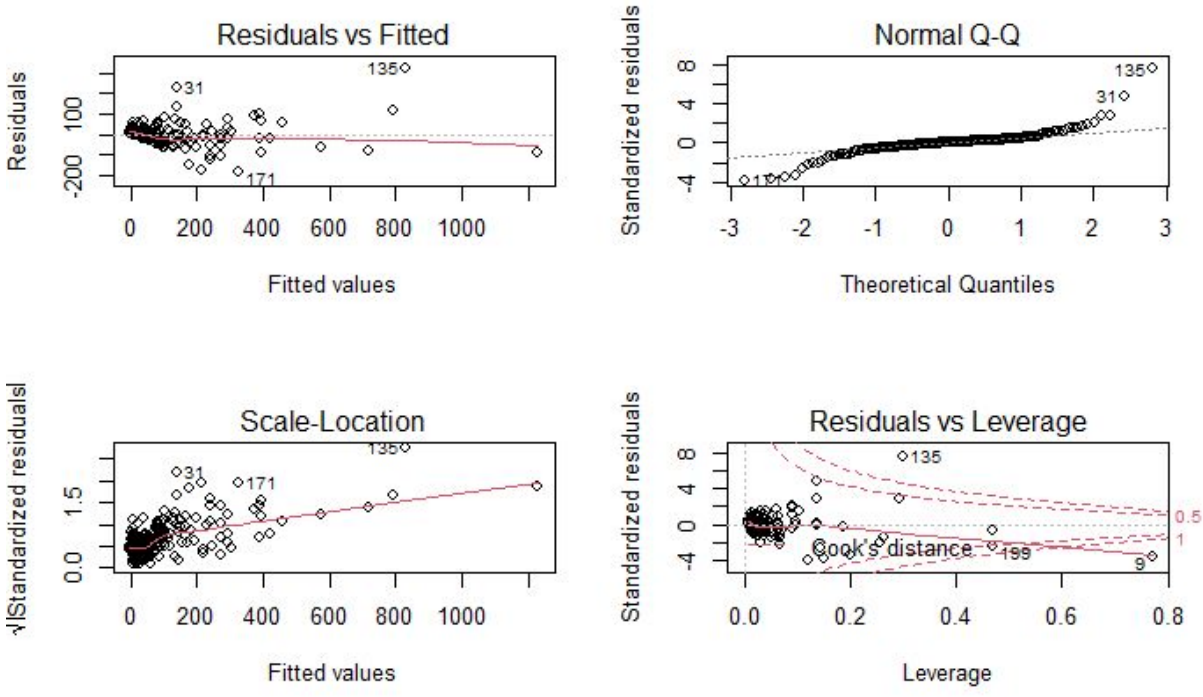
The residual error is the lowest here (50.92 where max performance value is 1150)

The Multiple R-squared is also the highest at 90%

Median Residual is highly promising in this case being only 1.37

The huge P value is worrying, this is not even close to allowing us to reject the null hypothesis but I assume we can ignore it for this project.

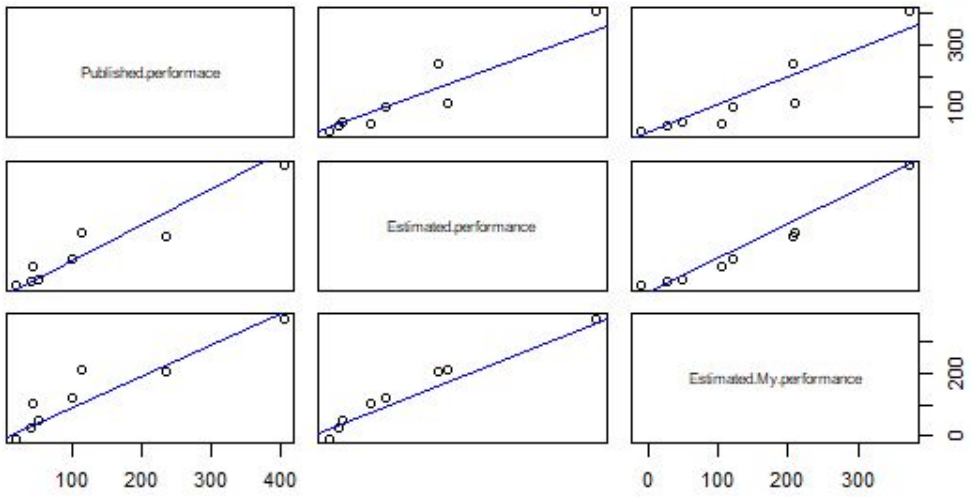
The residuals are not ideal, the extreme values are skewing them highly which is definitely an issue but being selective about the data is also going to give us a false confidence level.



Final predictions **Estimated.My.performance**

Published.performace	Estimated.performance	Estimated.My.performance
100	101	86
114	182	186
237	171	177
405	382	337
21	24	11
42	37	35
45	80	92
52	41	43

Addition model:



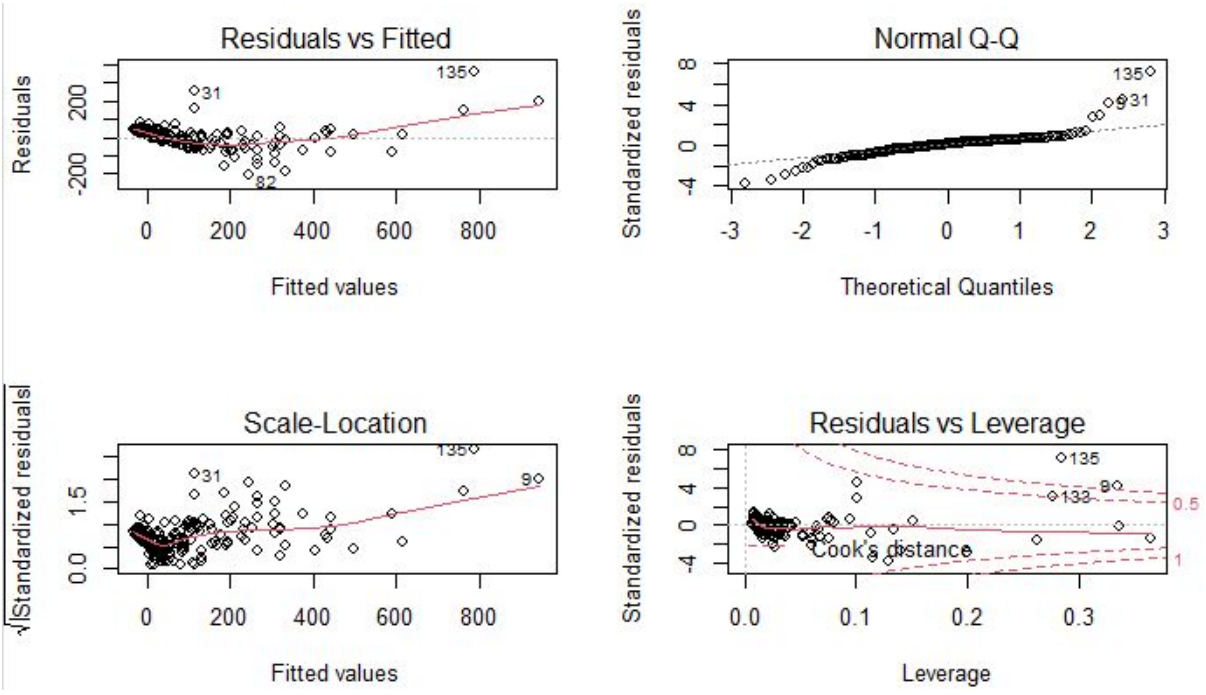
The line of best fit looks more in line here but the R squared is nearly 5% lower

```
Residuals:
    Min      1Q  Median      3Q     Max
-203.75  -25.64    9.27   27.39  371.50

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         -4.075e+01  6.094e+00  -6.686 2.38e-10 ***
Machine.memory.max   5.172e-03  6.861e-04   7.538 1.78e-12 ***
Machine.memory.min   1.437e-02  1.896e-03   7.576 1.42e-12 ***
Machine.memory.cache 7.664e-01  1.591e-01   4.817 2.93e-06 ***
Machine.channels.max 1.613e+00  2.343e-01   6.887 7.75e-11 ***
Machine.channels.min -7.289e-01  8.827e-01  -0.826     0.41
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 60.64 on 194 degrees of freedom
Multiple R-squared:  0.864,     Adjusted R-squared:  0.8605
F-statistic: 246.5 on 5 and 194 DF,  p-value: < 2.2e-16
```

This has a R-squared value that is much lower so while still being a potential good candidate it is not as good as the final model above. The Residual error is also slightly higher



Published.performace	Estimated.performance	Estimated.My.performance
100	101	111
114	182	206
237	171	205
405	382	376
21	24	-14
42	37	20
45	80	95
52	41	44

1. Polynomial Regression Analysis

1.1. Business Understanding (Same as Linear)

Business Objectives

The main objective is to analyse a range of quantifiable metrics to better predict the performance of a computer and give it a performance rating which helps compare it against other hardware available on the field.

Goals and success criteria

- Predict the performance metric of hardware computers
- Allow us to compare the metric against other similar ones in its bracket.

1.2. Data Understanding and Preparation (Same as Linear)

Describe Data

The data involves mostly concrete numerical values for performance in areas such as cycle time, min and max : channels and memory.

The data is alphabetically ordered by the producer so I will scramble some parts for my test set.

"Estimated.performance" contains values predicted by the original model too for comparison.

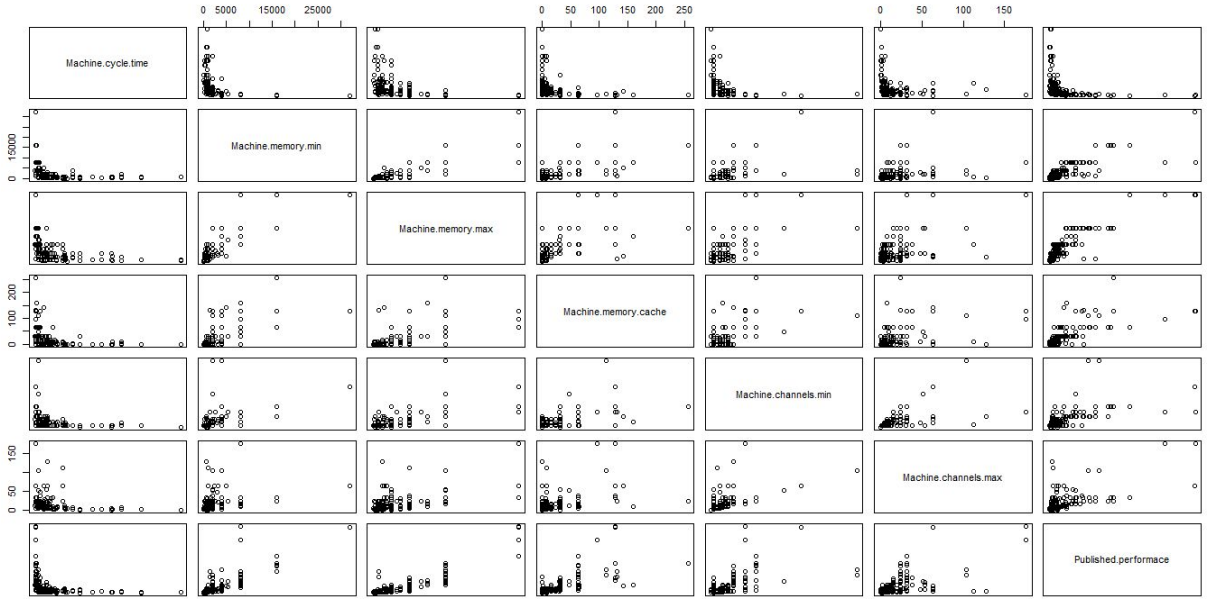
Example:

```
amdahl,470v/7,29,8000,32000,32,8,32,269,253
amdahl,470v/7a,29,8000,32000,32,8,32,220,253
amdahl,470v/7b,29,8000,32000,32,8,32,172,253
amdahl,470v/7c,29,8000,16000,32,8,16,132,132
amdahl,470v/b,26,8000,32000,64,8,32,318,290
amdahl,580-5840,23,16000,32000,64,16,32,367,381
amdahl,580-5850,23,16000,32000,64,16,32,489,381
amdahl,580-5860,23,16000,64000,64,16,32,636,749
amdahl,580-5880,23,32000,64000,128,32,64,1144,1238
```

Explore Data

While most values are at the lower side of the range with a few outliers we can see some nice linear correlation between **Published.Performance** and all the other parameters in the data set.

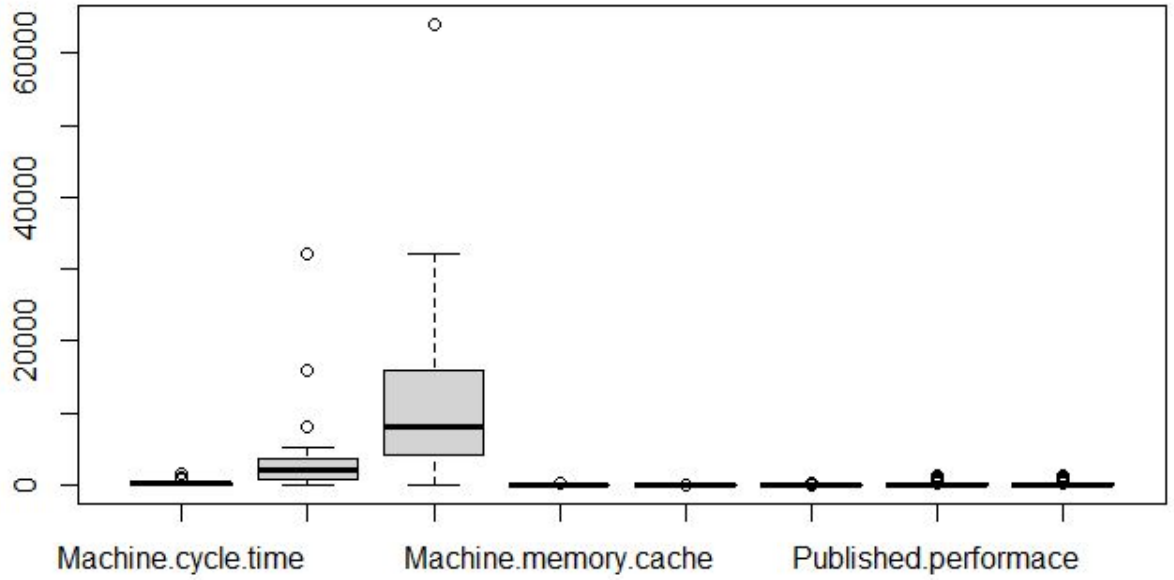
Machine.cycle.time can be ignored based on these graphs and keep the rest of the parameter



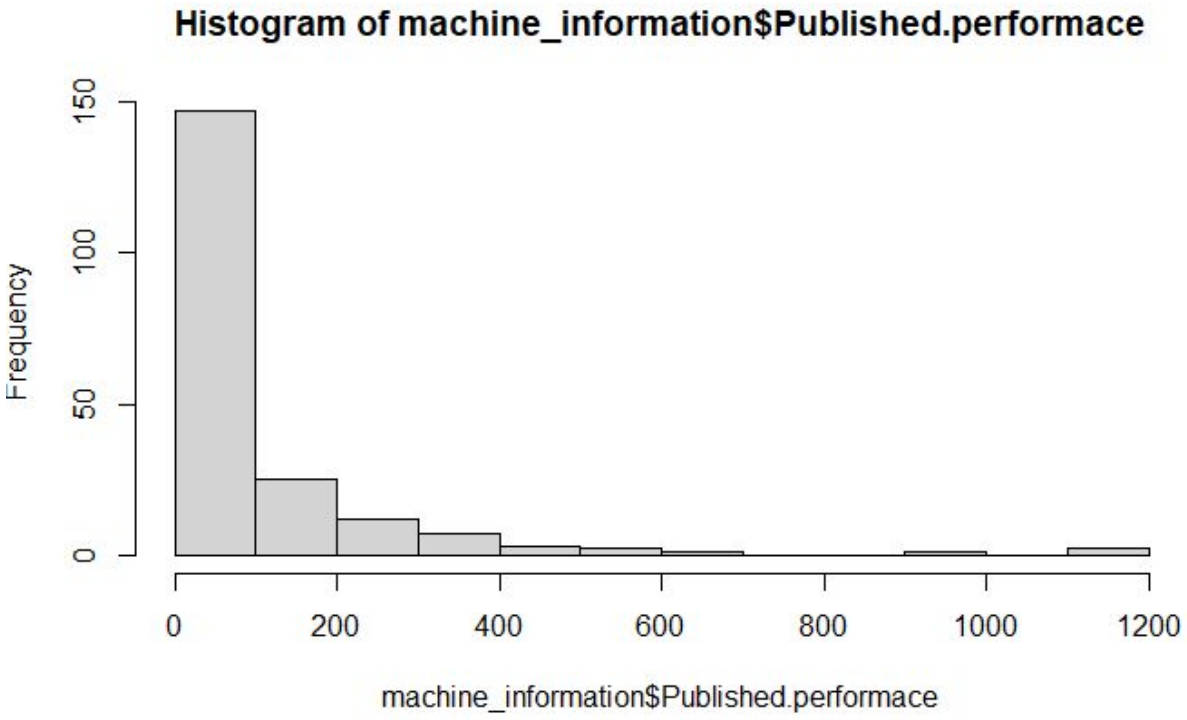
Strongest correlation towards the **Published.performance** can be seen with the **Machine.memory** values, the **Machine.cycle.time** "processing time" has a negative effect as it means it takes longer to complete a task.

```
Machine.cycle.time Machine.memory.min Machine.memory.max Machine.memory.cache
        -0.3056586          0.7966033          0.8702406           0.6965588
Machine.channels.min Machine.channels.max Published.performace Estimated.performance
           0.5971207            0.6225496                    1             0.9667149
```

There are also a few outliers in their respective columns which can be a little worrying and might need to be taken out.



There is also clearly a positive skew within the performance metric.



Both Skewness and Kurtosis values are extremely high compared to normal ranges (-1 : 1), this is not necessarily bad but does tell us we are dealing with non normally distributed data.

```
> skewness(machine_information$Published.performace)
[1] 3.916726
> kurtosis(machine_information$Published.performace)
[1] 21.96862
>
```

1.3. Modelling

Overview

Polynomial regression is slightly different but I have reused my two best performing models from linear and built from there.

Having taken some parameters in and out it still seems like leaving most of the parameters produces the best R values.

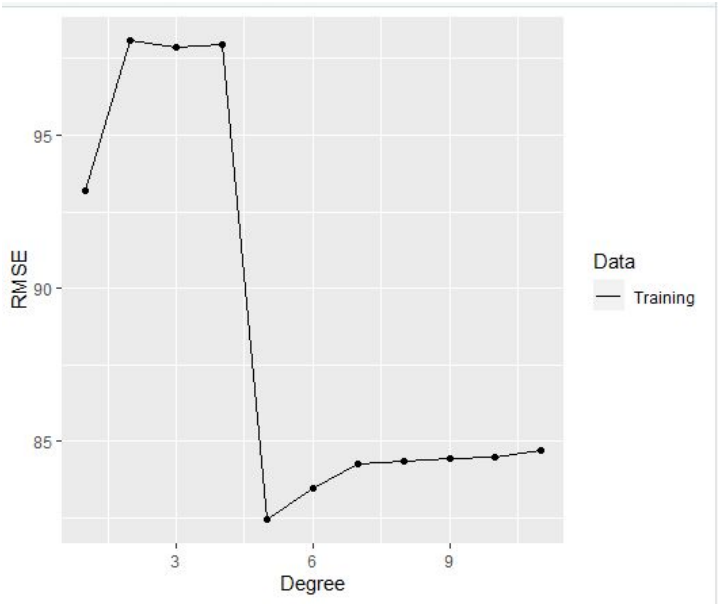
Since I already split the data into training and test I can Cross Validate the model after.

ANOVA of the models:

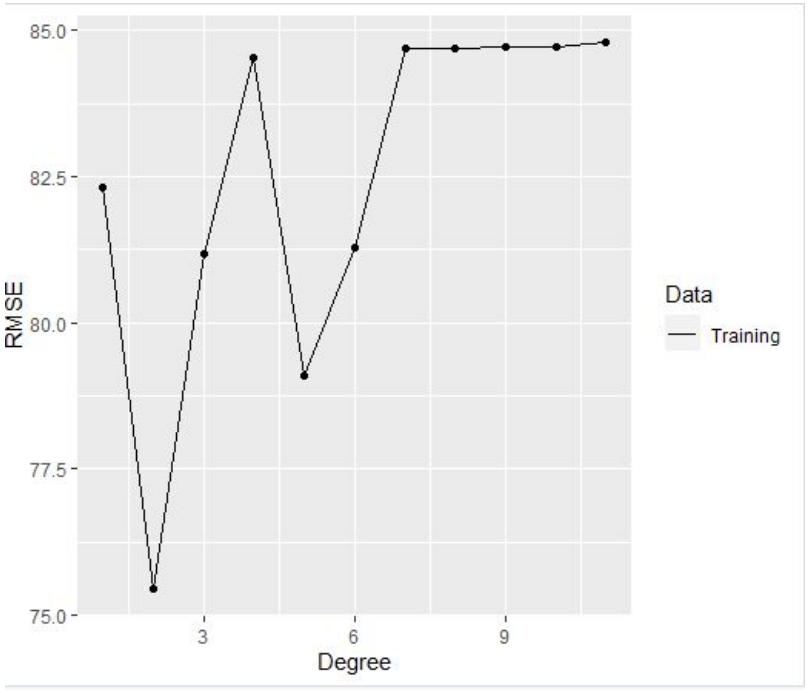
```
    Machine.channels.min - Estimated.performanc
  Res.Df    RSS Df Sum of Sq F Pr(>F)
1    194 963263
2    195 641385 -1    321878
> |
```

Exploration for ideal Degree values for models:

Multiplication model

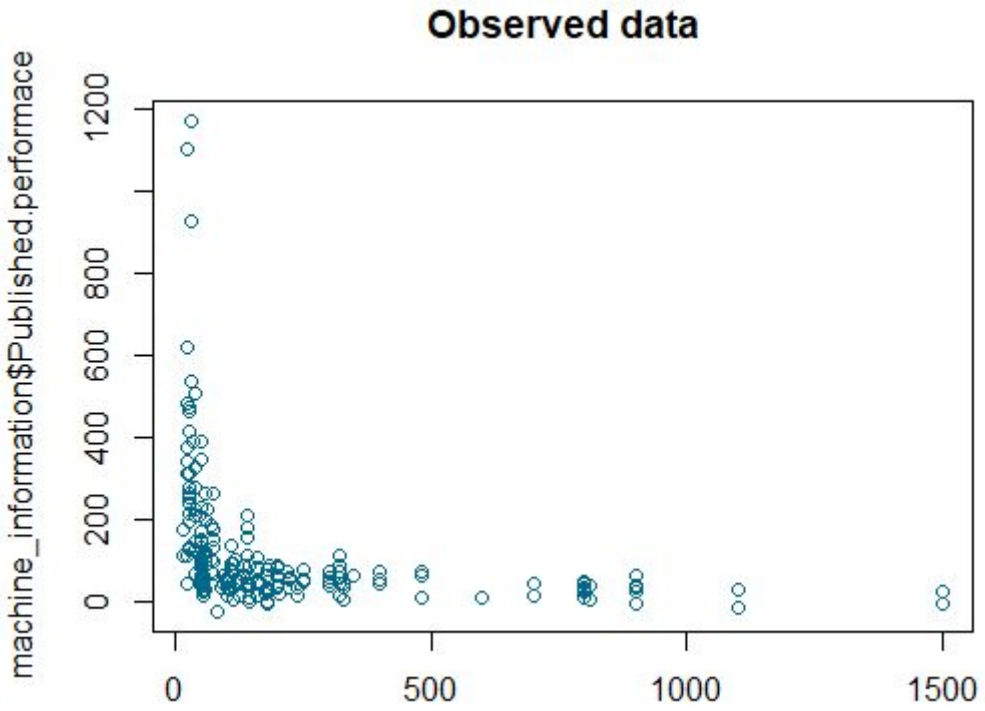


Addition model


85.0
82.5
80.0
77.5
75.0
RMSE
3
6
9
Degree
Data
Training

1.4. Evaluation

Overall I feel like polynomial regression is not the best choice of modelling for this specific data set.
Even after adding some noise it seems like the data does not follow a polynomial pattern.



The output of the top scoring test data is also very unsatisfactory compared to the Y that has been provided in the set. The variation of **Estimated.My.performance** is not even close to what it should be.

Published.performace	Estimated.performance	Estimated.My.performance
100	101	-259
114	182	1098
237	171	459
405	382	1494
21	24	-503
42	37	-539
45	80	-372
52	41	-543

I have still completed the necessary modelling.
This time the addition model was the clear winner but still underperformed compared to the linear regression for this data set.

I did not find another combination that would perform better than this with polynomial modelling.

Addition model with degree of 4:

```
machine_information_model2 <- lm(machine_information$Published.performace ~
                                 polym(
                                  +Machine.memory.max
                                  +Machine.memory.min
                                  +Machine.memory.cache
                                  +Machine.channels.max
                                  -Machine.channels.min
                                  -Estimated.performance #ignore the hardcoded predictions
                                  ,degree=4),data=machine_information)
machine_information_model2$coefficients
```

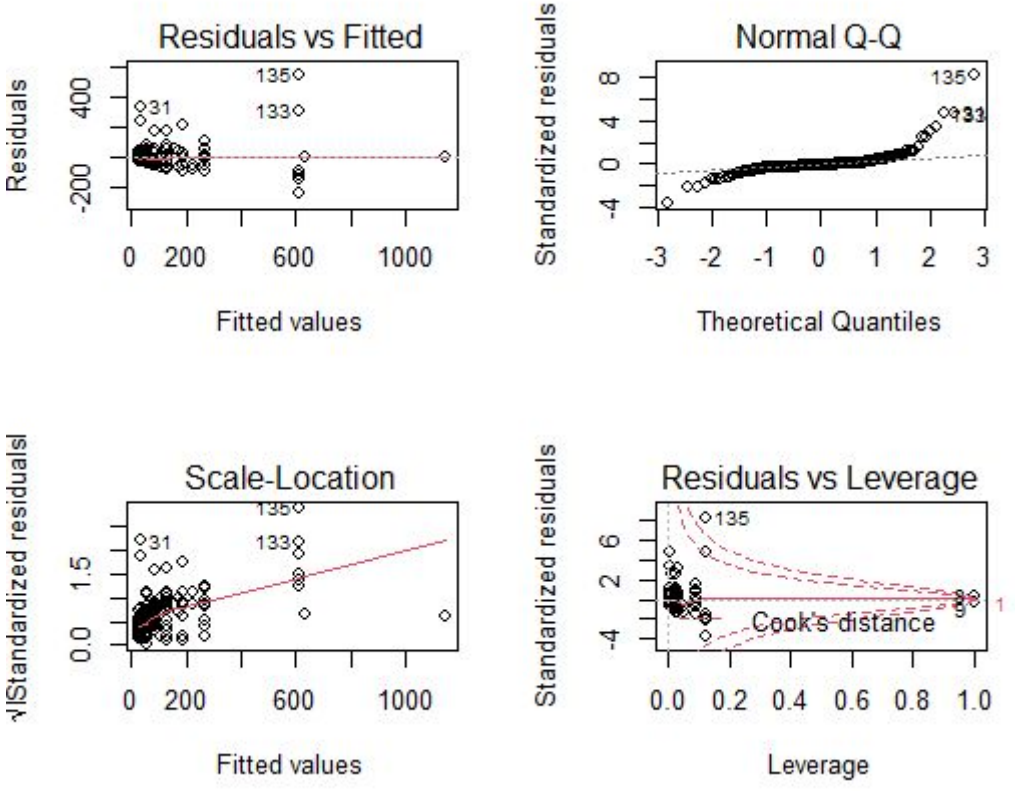
```
Residual standard error: 57.35 on 195 degrees of freedom
Multiple R-squared:  0.8778,    Adjusted R-squared:  0.8752
F-statistic:   350 on 4 and 195 DF,  p-value: < 2.2e-16
```

Multiplication model, degree of 5:

```
machine_information_model <- lm(machine_information$Published.performace ~
                                  polym(
                                  +Machine.memory.max*Machine.memory.min
                                  -Machine.memory.min
                                  +Machine.memory.cache
                                  +Machine.channels.max*Machine.channels.min
                                  ,degree=5)
                                  ,data=machine_information)
```

```
Residual standard error: 70.46 on 194 degrees of freedom
Multiple R-squared:  0.8164,    Adjusted R-squared:  0.8117
F-statistic: 172.5 on 5 and 194 DF,  p-value: < 2.2e-16
```

Residuals:



References:

1. https://rpubs.com/iabrady/residual-analysis
2. https://www.simplypsychology.org/p-value.html
3. http://www.learnbymarketing.com/tutorials/linear-regression-in-r/
4. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/lm