# Regression for Linear Models R Notebook

Swarn Singh and Ved Nigam

## We are exploring the relationship between CPU Cores and FPS for popular video games. The goal is to understand whether there is a linear relationship between both items

### Our dataset is from Kaggle: https://www.kaggle.com/datasets/kritikseth/achieved-frames-per-second-fps-in-video-games (https://www.kaggle.com/datasets/kritikseth/achieved-frames-per-second-fps-in-video-games)

Dividing the data into 80/20 for train/test

```
df <- read.csv("/Users/swarn/Machine Learning/cs4375-ml-portfolio/Assignment 3/fps-da
taset.csv");

set.seed(1234)
i <- sample(1:nrow(df), .8*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

### Data Exploration

A quick explanation about each function we are using

- `summary()` - The summary() function provides a summary of the variables in a data frame in R. It calculates descriptive statistics such as mean, median, minimum, and maximum for each numeric variable, and provides frequency counts for categorical variables. When summary() is applied to a data frame train, it returns a summary of each variable in the data frame
- `colSums()` - calculates the sum of missing values for each column in the data frame. This is a useful way to identify which columns have missing data in a data set.
- `str()` - The str() function provides information about the structure of an R object. When applied to a data frame train, it returns a compact display of the internal structure of the data frame, including the data types of each variable, the number of observations, and the first few observations of each variable.
- `names()` - The names() function returns the variable names of a data frame or a list. When applied to a data frame train, it returns a character vector containing the names of each variable in the data frame.
- `head()` - The head() function returns the first few rows of a data frame or a matrix. When applied to a data frame train, it returns the first six rows of the data frame. This is a useful way to preview the data and check if it has been read in correctly.

```
summary(train)
```

```
##        id           CpuName          CpuNumberOfCores CpuNumberOfThreads
##  Min.   :     1   Length:340666      Min.   : 1.00    Min.   : 1.000
##  1st Qu.:106592   Class :character   1st Qu.: 4.00    1st Qu.: 4.000
##  Median :212990   Mode  :character   Median : 4.00    Median : 8.000
##  Mean   :212947                      Mean   : 4.88    Mean   : 7.893
##  3rd Qu.:319454                      3rd Qu.: 6.00    3rd Qu.:12.000
##  Max.   :425833                      Max.   :32.00    Max.   :64.000
##   CpuBaseClock     CpuCacheL1       CpuCacheL2     CpuCacheL3
##  Min.   :100.0   Min.   :  64.0   Min.   :  512   Length:340666
##  1st Qu.:100.0   1st Qu.: 256.0   1st Qu.: 1024   Class :character
##  Median :100.0   Median : 256.0   Median : 1024   Mode  :character
##  Mean   :108.6   Mean   : 348.1   Mean   : 1990
##  3rd Qu.:100.0   3rd Qu.: 384.0   3rd Qu.: 2048
##  Max.   :800.0   Max.   :3072.0   Max.   :18432
##   CpuDieSize          CpuFrequency  CpuMultiplier   CpuMultiplierUnlocked
##  Length:340666      Min.   :1500   Min.   : 8.00   Min.   :0.0000
##  Class :character   1st Qu.:3300   1st Qu.:32.00   1st Qu.:0.0000
##  Mode  :character   Median :3500   Median :35.00   Median :1.0000
##                     Mean   :3496   Mean   :33.42   Mean   :0.5688
##                     3rd Qu.:3700   3rd Qu.:37.00   3rd Qu.:1.0000
##                     Max.   :4700   Max.   :43.00   Max.   :1.0000
##  CpuProcessSize     CpuTDP      CpuNumberOfTransistors CpuTurboClock
##  Min.   : 7.0   Min.   : 10.0   Length:340666          Min.   :1600
##  1st Qu.:14.0   1st Qu.: 65.0   Class :character       1st Qu.:3700
##  Median :14.0   Median : 88.0   Mode  :character       Median :4000
##  Mean   :17.3   Mean   : 81.4                          Mean   :3999
##  3rd Qu.:22.0   3rd Qu.: 95.0                          3rd Qu.:4350
##  Max.   :90.0   Max.   :250.0                          Max.   :5000
##    GpuName           GpuArchitecture    GpuBandwidth        GpuBaseClock
##  Length:340666      Length:340666      Length:340666      Min.   : 100
##  Class :character   Class :character   Class :character   1st Qu.:1050
##  Mode  :character   Mode  :character   Mode  :character   Median :1410
##                                                           Mean   :1284
##                                                           3rd Qu.:1506
##                                                           Max.   :1680
##  GpuBoostClock    X.GpuBus         GpuNumberOfComputeUnits  GpuDieSize
##  Min.   : 350   Length:340666      Length:340666           Length:340666
##  1st Qu.:1178   Class :character   Class :character        Class :character
##  Median :1582   Mode  :character   Mode  :character        Mode  :character
##  Mean   :1443
##  3rd Qu.:1709
##  Max.   :1980
##   GpuDirectX        GpuNumberOfExecutionUnits GpuFP32Performance
##  Length:340666      Length:340666             Length:340666
##  Class :character   Class :character          Class :character
##  Mode  :character   Mode  :character          Mode  :character
##
##
##
##  GpuMemoryBus       GpuMemorySize      GpuMemoryType       GpuOpenCL
```

```
##  Length:340666      Length:340666      Length:340666      Length:340666
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    GpuOpenGL     GpuPixelRate    GpuProcessSize   GpuNumberOfROPs
##  Min.   :2.00   Min.   :   425   Min.   :  7.00   Min.   : 1.00
##  1st Qu.:4.60   1st Qu.: 40510   1st Qu.: 14.00   1st Qu.:32.00
##  Median :4.60   Median : 82030   Median : 16.00   Median :48.00
##  Mean   :4.58   Mean   : 73315   Mean   : 18.52   Mean   :48.48
##  3rd Qu.:4.60   3rd Qu.:107700   3rd Qu.: 16.00   3rd Qu.:64.00
##  Max.   :4.60   Max.   :169900   Max.   :110.00   Max.   :96.00
##  GpuShaderModel     GpuNumberOfShadingUnits GpuNumberOfTMUs GpuTextureRate
##  Length:340666      Length:340666           Min.   :  2.0   Min.   :   850
##  Class :character   Class :character        1st Qu.: 80.0   1st Qu.: 87840
##  Mode  :character   Mode  :character        Median :120.0   Median :154400
##                                             Mean   :111.1   Mean   :166668
##                                             3rd Qu.:144.0   3rd Qu.:202000
##                                             Max.   :320.0   Max.   :509800
##  GpuNumberOfTransistors  GpuVulkan           GameName         GameResolution
##  Length:340666          Length:340666      Length:340666      Min.   : 720
##  Class :character       Class :character   Class :character   1st Qu.:1080
##  Mode  :character       Mode  :character   Mode  :character   Median :1080
##                                                               Mean   :1065
##                                                               3rd Qu.:1080
##                                                               Max.   :1440
##  GameSetting          Dataset               FPS
##  Length:340666      Length:340666      Min.   :   0.0
##  Class :character   Class :character   1st Qu.:  80.0
##  Mode  :character   Mode  :character   Median : 120.0
##                                        Mean   : 138.6
##                                        3rd Qu.: 180.0
##                                        Max.   :1000.0
```

```
colSums(is.na(train))
```

```
##                         id                    CpuName          CpuNumberOfCores
##                          0                          0                         0
##        CpuNumberOfThreads               CpuBaseClock                CpuCacheL1
##                          0                          0                         0
##                CpuCacheL2                CpuCacheL3                CpuDieSize
##                          0                          0                         0
##              CpuFrequency             CpuMultiplier     CpuMultiplierUnlocked
##                          0                          0                         0
##            CpuProcessSize                     CpuTDP    CpuNumberOfTransistors
##                          0                          0                         0
##             CpuTurboClock                    GpuName          GpuArchitecture
##                          0                          0                         0
##              GpuBandwidth               GpuBaseClock             GpuBoostClock
##                          0                          0                         0
##                  X.GpuBus   GpuNumberOfComputeUnits                GpuDieSize
##                          0                          0                         0
##               GpuDirectX GpuNumberOfExecutionUnits        GpuFP32Performance
##                          0                          0                         0
##              GpuMemoryBus              GpuMemorySize             GpuMemoryType
##                          0                          0                         0
##                GpuOpenCL                 GpuOpenGL              GpuPixelRate
##                          0                          0                         0
##            GpuProcessSize           GpuNumberOfROPs             GpuShaderModel
##                          0                          0                         0
##   GpuNumberOfShadingUnits           GpuNumberOfTMUs             GpuTextureRate
##                          0                          0                         0
##    GpuNumberOfTransistors                 GpuVulkan                   GameName
##                          0                          0                         0
##            GameResolution               GameSetting                    Dataset
##                          0                          0                         0
##                       FPS
##                          0
```

```
str(train)
```

```
## 'data.frame':    340666 obs. of  46 variables:
##  $ id                       : int  237392 106390 304108 408457 295846 126055 38255
4 345167 342900 347518 ...
##  $ CpuName                  : chr  "Intel Core i7-2600K" "Intel Core i3-2100" "AMD
Ryzen 5 1600" "Intel Core i5-7500" ...
##  $ CpuNumberOfCores         : int  4 2 6 4 6 4 4 4 2 8 ...
##  $ CpuNumberOfThreads       : int  8 4 12 4 6 8 4 8 2 16 ...
##  $ CpuBaseClock             : int  100 100 100 100 200 100 100 100 200 100 ...
##  $ CpuCacheL1               : int  256 128 576 256 288 256 256 256 256 768 ...
##  $ CpuCacheL2               : int  1024 512 3072 1024 6144 1024 1024 1024 2048 409
6 ...
##  $ CpuCacheL3               : chr  "8" "3" "16" "6" ...
##  $ CpuDieSize               : chr  "0.000216" "0.000131" "0.000192" "?" ...
```

```
##  $ CpuFrequency             : int  3400 3100 3200 3400 3500 4200 3800 4200 3000 30
00 ...
##  $ CpuMultiplier            : num  34 31 32 34 17.5 42 38 42 15 30 ...
##  $ CpuMultiplierUnlocked    : int  1 0 1 0 1 1 1 1 0 1 ...
##  $ CpuProcessSize           : int  32 32 14 14 32 14 14 14 45 14 ...
##  $ CpuTDP                   : int  95 65 65 65 95 91 91 91 65 65 ...
##  $ CpuNumberOfTransistors   : chr  "1160" "504" "4800" "?" ...
##  $ CpuTurboClock            : int  3800 3100 3600 3800 4100 4500 4200 4500 3000 37
00 ...
##  $ GpuName                  : chr  "AMD Radeon RX 580 OEM" "AMD Radeon R7 250 OEM"
"NVIDIA GeForce GTX 1050 Ti" "NVIDIA GeForce GTX TITAN X" ...
##  $ GpuArchitecture          : chr  "GCN 4.0" "GCN 1.0" "Pascal" "Maxwell 2.0" ...
##  $ GpuBandwidth             : chr  "256000" "32000" "112100" "336600" ...
##  $ GpuBaseClock             : int  1120 1000 1291 1000 1000 1481 1506 1506 600 148
1 ...
##  $ GpuBoostClock            : int  1266 1050 1392 1089 1050 1582 1709 1683 600 158
2 ...
##  $ X.GpuBus                 : chr  "PCIe 3.0 x16" "PCIe 3.0 x8" "PCIe 3.0 x16" "PC
Ie 3.0 x16" ...
##  $ GpuNumberOfComputeUnits  : chr  "36" "6" "?" "?" ...
##  $ GpuDieSize               : chr  "0.000232" "0.000077" "0.000132" "0.000601" ...
##  $ GpuDirectX               : chr  "12" "12" "12" "12" ...
##  $ GpuNumberOfExecutionUnits: chr  "?" "?" "?" "?" ...
##  $ GpuFP32Performance       : chr  "5834000" "806400" "2138000" "6691000" ...
##  $ GpuMemoryBus             : chr  "256" "128" "128" "384" ...
##  $ GpuMemorySize            : chr  "8000" "2000" "4000" "12000" ...
##  $ GpuMemoryType            : chr  "GDDR5" "DDR3" "GDDR5" "GDDR5" ...
##  $ GpuOpenCL                : chr  "2" "1.2" "1.2" "1.2" ...
##  $ GpuOpenGL                : num  4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 3.3 4.6 ...
##  $ GpuPixelRate             : int  40510 8400 44540 104500 8400 139200 82030 10770
0 4800 139200 ...
##  $ GpuProcessSize           : int  14 28 14 28 28 16 16 16 65 16 ...
##  $ GpuNumberOfROPs          : int  32 8 32 96 8 88 48 64 8 88 ...
##  $ GpuShaderModel           : chr  "6.4" "5.1" "6.4" "6.4" ...
##  $ GpuNumberOfShadingUnits  : chr  "2304" "384" "768" "3072" ...
##  $ GpuNumberOfTMUs          : int  144 24 48 192 24 224 80 120 16 224 ...
##  $ GpuTextureRate           : int  182300 25200 66820 209100 25200 354400 136700 2
02000 9600 354400 ...
##  $ GpuNumberOfTransistors   : chr  "5700" "950" "3300" "8000" ...
##  $ GpuVulkan                : chr  "1.2.131" "1.2.131" "1.2.131" "1.1.126" ...
##  $ GameName                 : chr  "playerUnknownsBattlegrounds" "grandTheftAuto5"
"fortnite" "counterStrikeGlobalOffensive" ...
##  $ GameResolution           : int  1080 720 1080 1080 1080 1080 1080 1080 720 1080
...
##  $ GameSetting              : chr  "max" "med" "high" "med" ...
##  $ Dataset                  : chr  "userbenchmark" "userbenchmark" "userbenchmark"
"fpsbenchmark" ...
##  $ FPS                      : num  70 20 70 287 30 ...
```

```
names(train)
```

```
##  [1] "id"                        "CpuName"
##  [3] "CpuNumberOfCores"          "CpuNumberOfThreads"
##  [5] "CpuBaseClock"              "CpuCacheL1"
##  [7] "CpuCacheL2"                "CpuCacheL3"
##  [9] "CpuDieSize"                "CpuFrequency"
## [11] "CpuMultiplier"             "CpuMultiplierUnlocked"
## [13] "CpuProcessSize"            "CpuTDP"
## [15] "CpuNumberOfTransistors"    "CpuTurboClock"
## [17] "GpuName"                   "GpuArchitecture"
## [19] "GpuBandwidth"              "GpuBaseClock"
## [21] "GpuBoostClock"             "X.GpuBus"
## [23] "GpuNumberOfComputeUnits"   "GpuDieSize"
## [25] "GpuDirectX"                "GpuNumberOfExecutionUnits"
## [27] "GpuFP32Performance"        "GpuMemoryBus"
## [29] "GpuMemorySize"             "GpuMemoryType"
## [31] "GpuOpenCL"                 "GpuOpenGL"
## [33] "GpuPixelRate"              "GpuProcessSize"
## [35] "GpuNumberOfROPs"           "GpuShaderModel"
## [37] "GpuNumberOfShadingUnits"   "GpuNumberOfTMUs"
## [39] "GpuTextureRate"            "GpuNumberOfTransistors"
## [41] "GpuVulkan"                 "GameName"
## [43] "GameResolution"            "GameSetting"
## [45] "Dataset"                   "FPS"
```

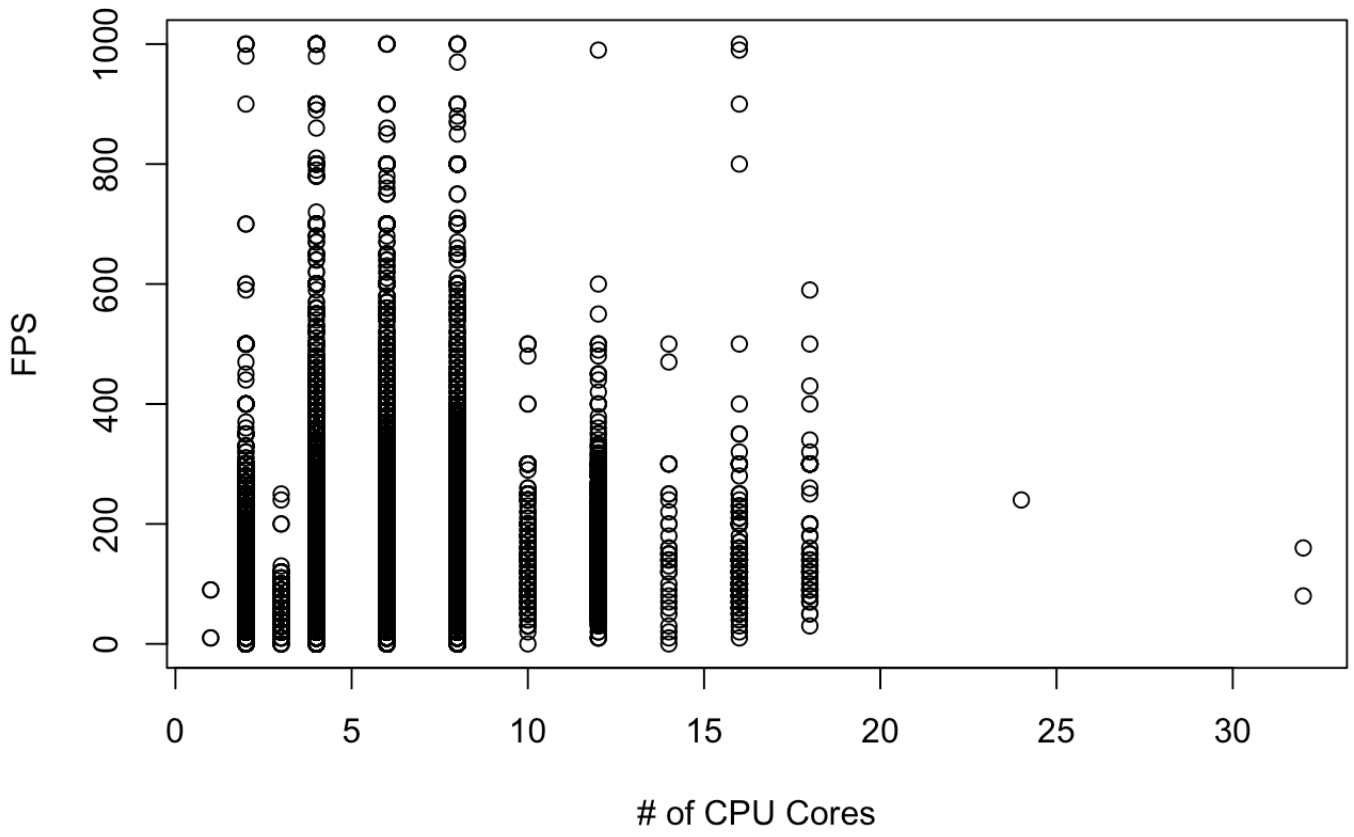
```
head(train)
```

| | id | CpuName | CpuNumberOfCo… | CpuNumberOfThreads | CpuBaseClo… |
|---|---|---|---|---|---|
| | <int> | <chr> | <int> | <int> | <int |
| 237392 | 237392 | Intel Core i7-2600K | 4 | 8 | 10 |
| 106390 | 106390 | Intel Core i3-2100 | 2 | 4 | 10 |
| 304108 | 304108 | AMD Ryzen 5 1600 | 6 | 12 | 10 |
| 408457 | 408457 | Intel Core i5-7500 | 4 | 4 | 10 |
| 295846 | 295846 | AMD FX-6300 | 6 | 6 | 20 |
| 126055 | 126055 | Intel Core i7-7700K | 4 | 8 | 10 |

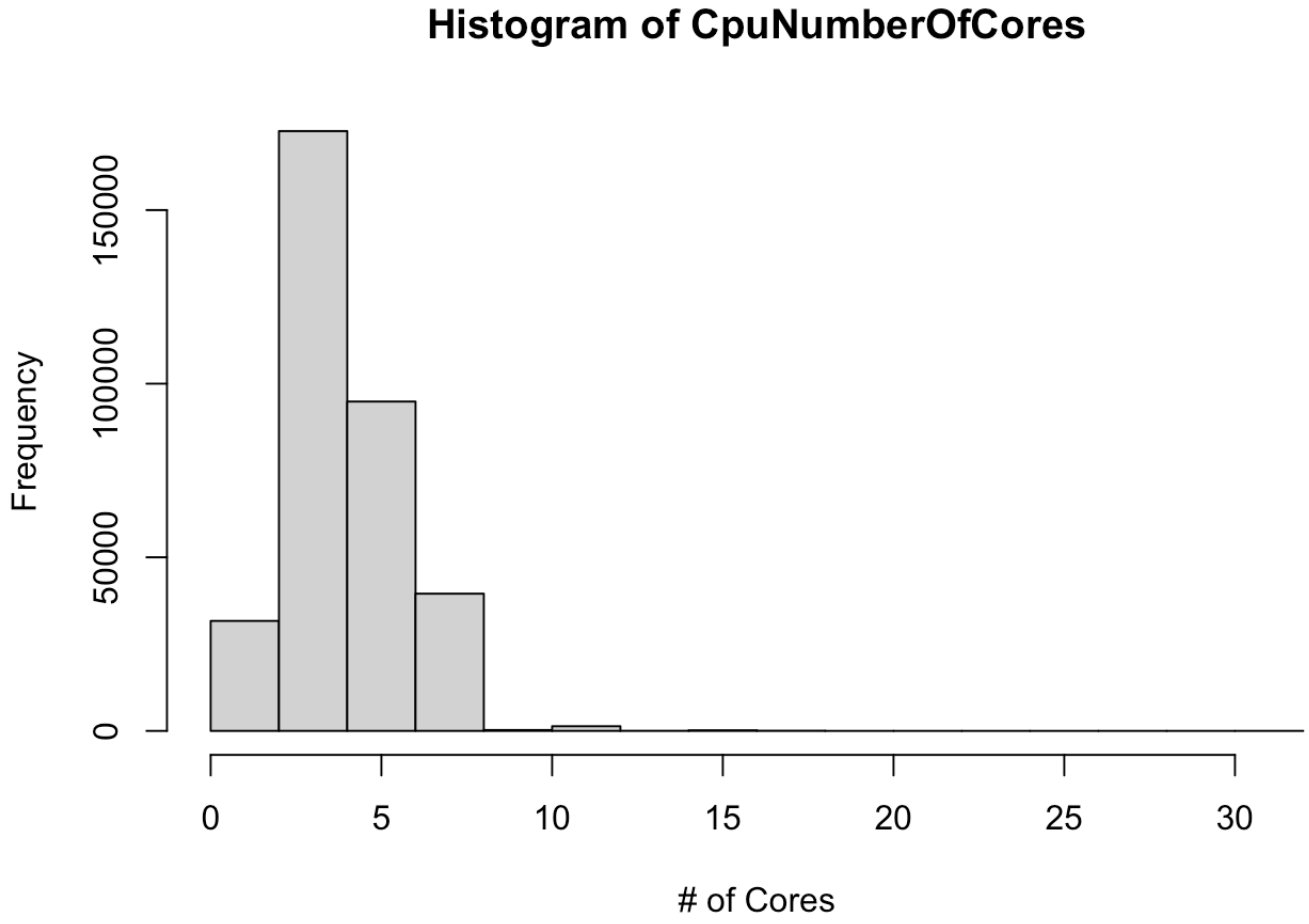
6 rows | 1-7 of 47 columns

## Informative Graphs

```
plot(train$CpuNumberOfCores, train$FPS, main="Scatter plot of # of CPU Cores and FPS"
, xlab="# of CPU Cores", ylab="FPS")
```




```
hist(train$CpuNumberOfCores, main="Histogram of CpuNumberOfCores", xlab="# of Cores",
ylab="Frequency")
```




The scatter plot shows us a quick visual that the FPS in the dataset can go up to a 1000 and it's achievable with just 5 CPU Cores The histogram shows us that there are alot more instances on 5-10 cores within the dataset

### Linear Regression

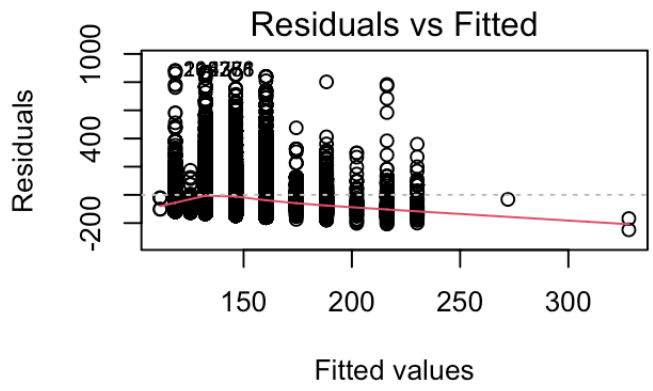
```
lm1 <- lm(formula = FPS ~ CpuNumberOfCores, data = df)
summary(lm1)
```

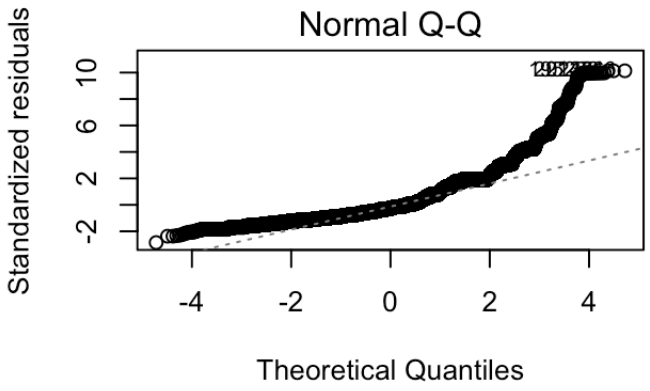
```
##
## Call:
## lm(formula = FPS ~ CpuNumberOfCores, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -247.93  -62.41  -22.41   39.66  881.56
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      104.4759     0.4005  260.88   <2e-16 ***
## CpuNumberOfCores   6.9829     0.0774   90.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 86.99 on 425831 degrees of freedom
## Multiple R-squared:  0.01875,    Adjusted R-squared:  0.01875
## F-statistic:  8139 on 1 and 425831 DF,  p-value: < 2.2e-16
```

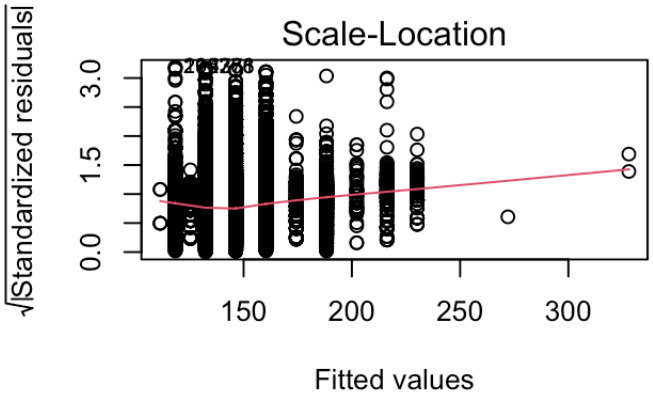
- Based on this output, we can conclude that the number of CPU cores is a significant predictor of FPS, and that the relationship is positive and moderately strong, but explains only a small amount of the variance in the dependent variable. *The p-value is <2.2e-16 for both the intercept and the CpuNumberOfCores variable, indicating that they are both highly statistically significant predictors of the dependent variable (FPS).
- The standard error is 0.0774 which is pretty low

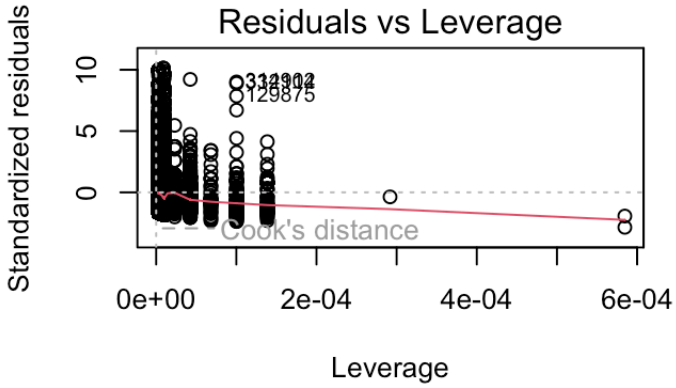
### Plot Residuals

```
par(mfrow = c(2,2))
plot(lm1)
```



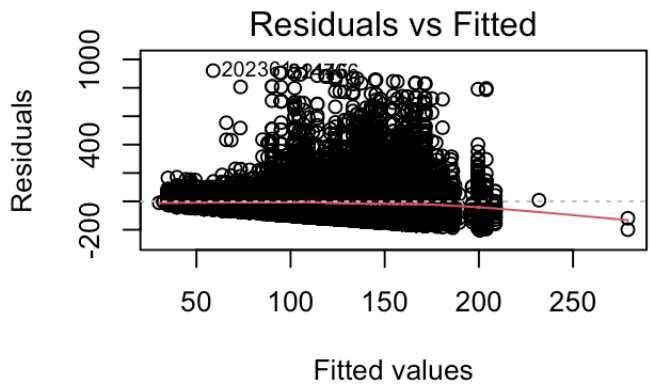

- In the Residuals vs Fitted graph the line is kind of linear but dips after 150
- In the Q-Q graph, the line is running along the data from [-2,2], unclear as to why it goes so far apart after 2 though
- The Scale-location graph is similar to the Residuals vs Fitted but inverted. Constant slope after 150 on the x-axis
- Couldn't really understand the last graph and the zooming is unclear. Scaling is something that we ran out of time for

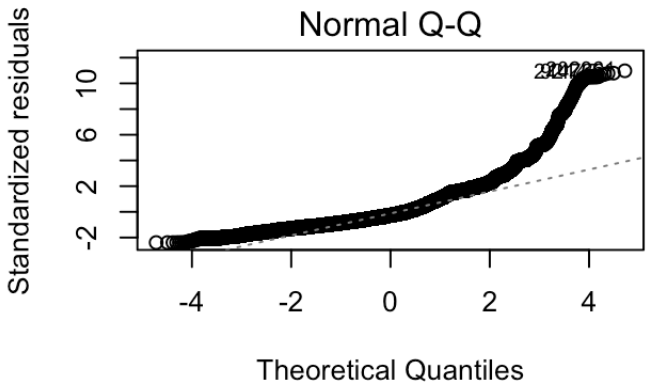
### Multiple Linear Regression

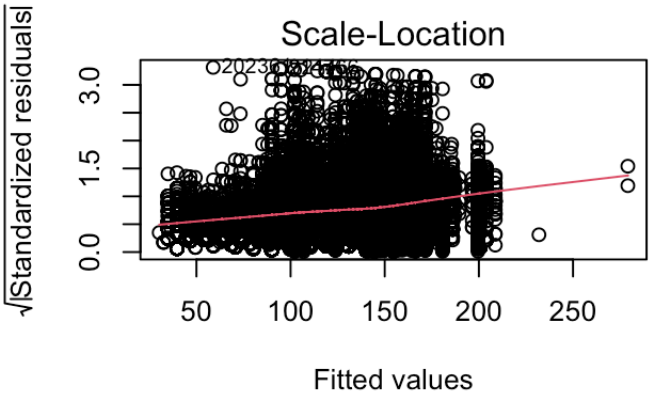
```
lm2 <- lm(FPS ~ CpuNumberOfCores + CpuNumberOfThreads +  CpuFrequency, data = df)

summary(lm2)
```

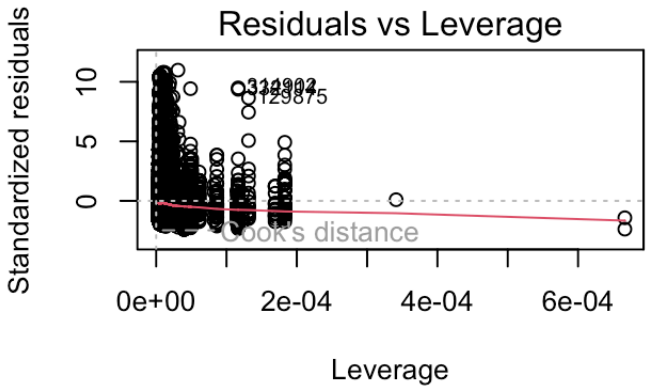
```
##
## Call:
## lm(formula = FPS ~ CpuNumberOfCores + CpuNumberOfThreads + CpuFrequency,
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -200.48  -58.58  -22.10   38.09  921.12
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -5.376e+01  1.058e+00  -50.80   <2e-16 ***
## CpuNumberOfCores    1.237e+00  1.147e-01   10.79   <2e-16 ***
## CpuNumberOfThreads  2.332e+00  5.229e-02   44.59   <2e-16 ***
## CpuFrequency        4.802e-02  2.964e-04  162.03   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 83.9 on 425829 degrees of freedom
## Multiple R-squared:  0.08723,    Adjusted R-squared:  0.08723
## F-statistic: 1.357e+04 on 3 and 425829 DF,  p-value: < 2.2e-16
```

```
#residuals
par(mfrow = c(2,2))
plot(lm2)
```



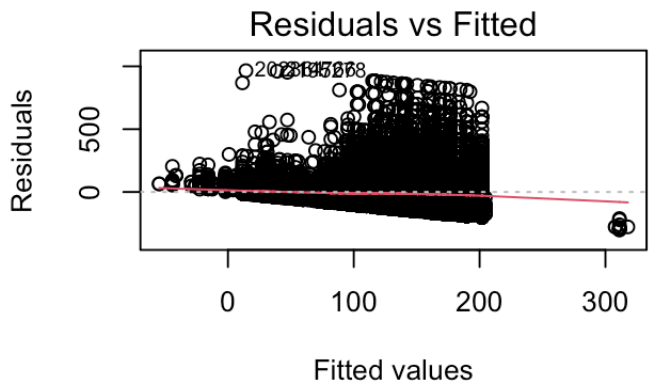

```
#summary
summary(lm2)
```

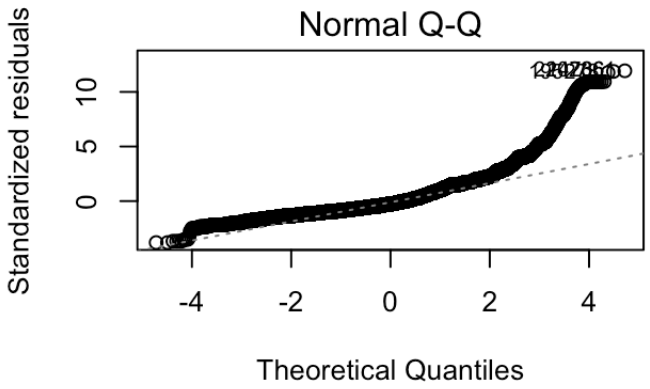
```
##
## Call:
## lm(formula = FPS ~ CpuNumberOfCores + CpuNumberOfThreads + CpuFrequency,
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -200.48  -58.58  -22.10   38.09  921.12
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -5.376e+01  1.058e+00  -50.80   <2e-16 ***
## CpuNumberOfCores    1.237e+00  1.147e-01   10.79   <2e-16 ***
## CpuNumberOfThreads  2.332e+00  5.229e-02   44.59   <2e-16 ***
## CpuFrequency        4.802e-02  2.964e-04  162.03   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 83.9 on 425829 degrees of freedom
## Multiple R-squared:  0.08723,    Adjusted R-squared:  0.08723
## F-statistic: 1.357e+04 on 3 and 425829 DF,  p-value: < 2.2e-16
```

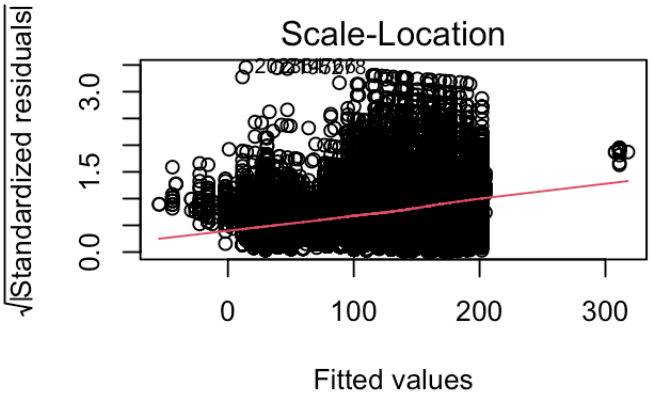
```
lm3 <- lm(FPS ~ CpuNumberOfCores + CpuNumberOfThreads +  CpuFrequency + CpuMultiplier
+ CpuProcessSize + CpuBaseClock + CpuNumberOfThreads + CpuTurboClock + CpuBaseClock,
data = df)
summary(lm3)
```

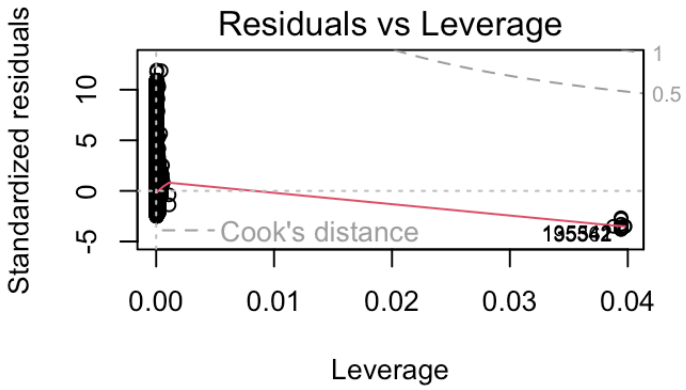
```
##
## Call:
## lm(formula = FPS ~ CpuNumberOfCores + CpuNumberOfThreads + CpuFrequency +
##     CpuMultiplier + CpuProcessSize + CpuBaseClock + CpuNumberOfThreads +
##     CpuTurboClock + CpuBaseClock, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -301.13  -56.54  -20.46   39.05  965.72
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -1.449e+02  2.946e+00  -49.19   <2e-16 ***
## CpuNumberOfCores    5.877e+00  1.597e-01   36.80   <2e-16 ***
## CpuNumberOfThreads -1.990e+00  5.986e-02  -33.24   <2e-16 ***
## CpuFrequency       -3.500e-02  1.423e-03  -24.60   <2e-16 ***
## CpuMultiplier       5.807e+00  1.445e-01   40.17   <2e-16 ***
## CpuProcessSize     -8.659e-01  2.870e-02  -30.16   <2e-16 ***
## CpuBaseClock        5.216e-01  2.526e-02   20.65   <2e-16 ***
## CpuTurboClock       3.930e-02  5.351e-04   73.45   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 80.92 on 425825 degrees of freedom
## Multiple R-squared:  0.1509, Adjusted R-squared:  0.1508
## F-statistic: 1.081e+04 on 7 and 425825 DF,  p-value: < 2.2e-16
```

```
#residuals
par(mfrow = c(2,2))
plot(lm3)
```




```
#summary
summary(lm3)
```

```
##
## Call:
## lm(formula = FPS ~ CpuNumberOfCores + CpuNumberOfThreads + CpuFrequency +
##     CpuMultiplier + CpuProcessSize + CpuBaseClock + CpuNumberOfThreads +
##     CpuTurboClock + CpuBaseClock, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -301.13  -56.54  -20.46   39.05  965.72
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -1.449e+02  2.946e+00  -49.19   <2e-16 ***
## CpuNumberOfCores    5.877e+00  1.597e-01   36.80   <2e-16 ***
## CpuNumberOfThreads -1.990e+00  5.986e-02  -33.24   <2e-16 ***
## CpuFrequency       -3.500e-02  1.423e-03  -24.60   <2e-16 ***
## CpuMultiplier       5.807e+00  1.445e-01   40.17   <2e-16 ***
## CpuProcessSize     -8.659e-01  2.870e-02  -30.16   <2e-16 ***
## CpuBaseClock        5.216e-01  2.526e-02   20.65   <2e-16 ***
## CpuTurboClock       3.930e-02  5.351e-04   73.45   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 80.92 on 425825 degrees of freedom
## Multiple R-squared:  0.1509, Adjusted R-squared:  0.1508
## F-statistic: 1.081e+04 on 7 and 425825 DF,  p-value: < 2.2e-16
```

## Predictions

### Model 1

```
pred1 <- predict(lm1,newdata = test)
cor1 <- cor(pred1,test$FPS)
mse1 <- mean((pred1 - test$FPS)^2)
rmse1 <- sqrt(mse1)
head(pred1)
```

```
##        3        5        7       18       33       36
## 132.4074 132.4074 132.4074 132.4074 132.4074 132.4074
```

```
cor1
```

```
## [1] 0.1411365
```

```
mse1
```

```
## [1] 7546.688
```

```
rmse1
```

```
## [1] 86.87168
```

Model 2

```
pred2 <- predict(lm2,newdata = test)
cor2 <- cor(pred2,test$FPS)
mse2 <- mean((pred2 - test$FPS)^2)
rmse2 <- sqrt(mse2)
head(pred2)
```

```
##        3        5        7       18       33       36
## 123.7773 123.7773 123.7773 123.7773 123.7773 123.7773
```

```
cor2
```

```
## [1] 0.2983036
```

```
mse2
```

```
## [1] 7014.775
```

```
rmse2
```

```
## [1] 83.75425
```

Model 3

```
pred3 <- predict(lm3,newdata = test)
cor3 <- cor(pred3,test$FPS)
mse3 <- mean((pred3 - test$FPS)^2)
rmse3 <- sqrt(mse3)
head(pred3)
```

```
##        3        5        7       18       33       36
## 131.4905 131.4905 131.4905 131.4905 131.4905 131.4905
```

```
cor3
```

```
## [1] 0.392428
```

```
mse3
```

```
## [1] 6514.242
```

```
rmse3
```

```
## [1] 80.71085
```

- The code provided fits three linear regression models using different independent variables (NumberOfCPUCores, Log_NumberOfCPUCores, and Squared_NumberOfCPUCores) to predict the dependent variable (FPS). Each model makes different assumptions about the relationship between the variables, and the output displays performance metrics for each model when applied to the test data.

*Based on the results of the three models, it seems that Model 3 (Squared_NumberOfCPUCores) is the best model, as it has the highest correlation and the lowest MSE and RMSE values. This suggests that the relationship between NumberOfCPUCores and FPS may not be linear, but instead may follow a quadratic relationship. However, further analysis and exploration of the data is necessary to confirm this.

- Using the three models, the output displays the performance metrics for each model when applied to the test data. The correlation coefficients range from 0.14 to 0.39, indicating that there is a weak to moderate relationship between the predicted and actual values of FPS. The MSE values range from 6514 to 7546, while the RMSE values range from 80.7 to 86.9, with lower values indicating better model performance.
- Comparing the results of the three models, it seems that Model 3 (Squared_NumberOfCPUCores) performs the best, with the highest correlation and the lowest MSE and RMSE values. This indicates that the squared term of NumberOfCPUCores may be an important predictor of FPS. Model 2 (Log_NumberOfCPUCores) also performs well, with a higher correlation than Model 1 (NumberOfCPUCores) and similar MSE and RMSE values to Model 3. This suggests that the log transformation of NumberOfCPUCores may improve model performance. Overall, the results suggest that a more complex model may be necessary to accurately predict FPS based on the available independent variables.