ASML_Assessement

Gustavo Chinchayan

9/27/2021

Exercise 1:

Importing of the file data1.txt on to the Rstudio platform

```
data1 <- read.table("C:/Users/USER/Documents/R/R Exercize/data1.txt", header=FALSE, col.names =
"Values")
```

by observing the data set to make sure it has been loaded successfully

```
str(data1)
```

```
## 'data.frame':    4766 obs. of  1 variable:
##  $ Values: num  0.00601 0.14179 0.25937 0.82704 1.62859 ...
```

```
head(data1, 10)
```

```
##         Values
## 1  0.006005354
## 2  0.141786600
## 3  0.259368800
## 4  0.827037100
## 5  1.628588000
## 6  1.673124000
## 7  1.981014000
## 8  2.015298000
## 9  2.831079000
## 10 4.052456000
```

```
which(is.na(data1)) #No Null values present in the dataset1 file
```

```
## integer(0)
```

In theoretical context, the Poisson Process meets the following criteria:

1. Events and independent of each other. The occurrence of one event does not affect the probability another event will occur.
2. The average (events per time period) is constant.
3. Two events cannot occur at the same time (similar to a Bernoulli Trial.

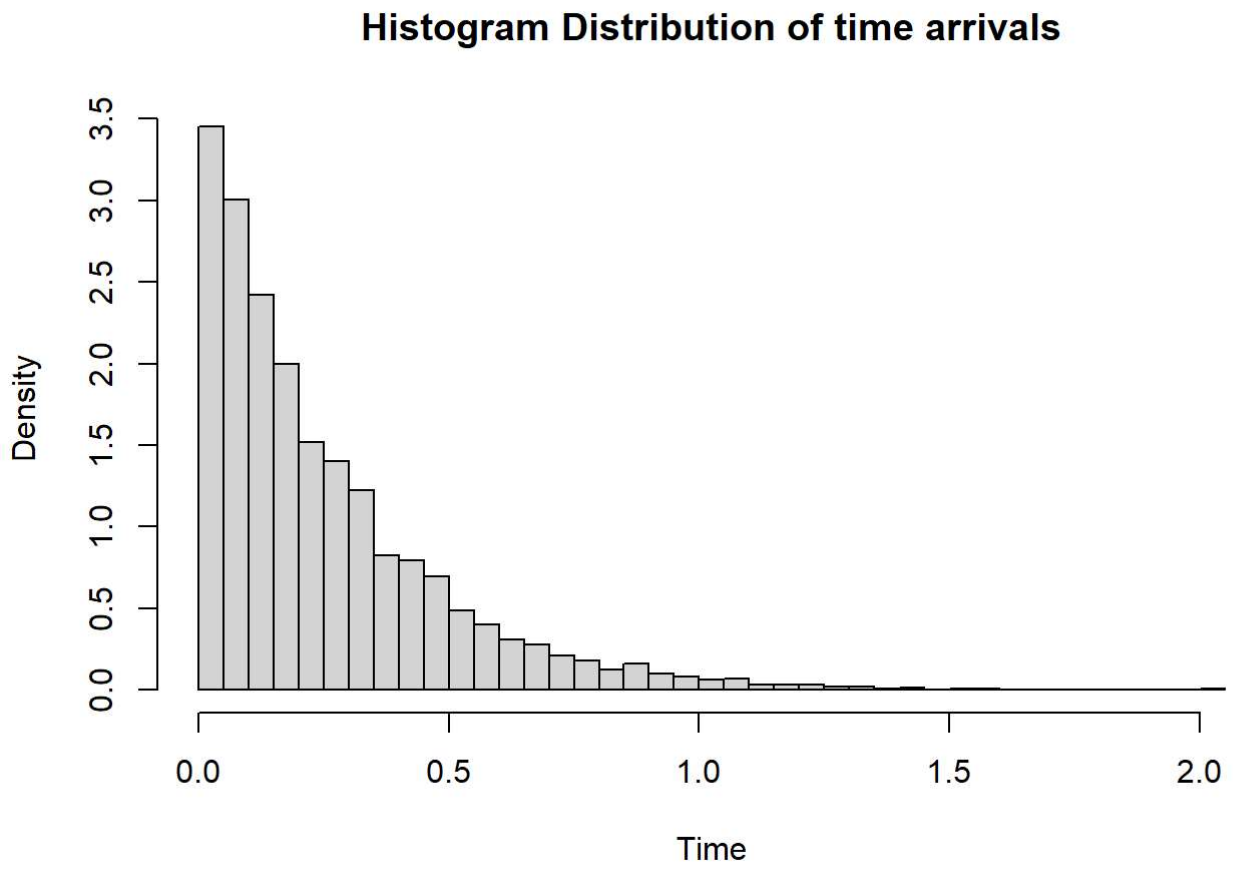
Beginning with the information, the first step is to compute the exponential random variable into a data-frame variable

```
exponentialrv <- {}

for (i in c(1:(length(data1$Values)-1)))
{
  exponentialrv[i] <- data1$Values[i+1]-data1$Values[i]
}

dfexponentialrv <- data.frame(exponentialrv)
head(dfexponentialrv,15)
```

```
##    exponentialrv
## 1      0.1357812
## 2      0.1175822
## 3      0.5676683
## 4      0.8015509
## 5      0.0445360
## 6      0.3078900
## 7      0.0342840
## 8      0.8157810
## 9      1.2213770
## 10     0.1620000
## 11     0.5480350
## 12     0.1096310
## 13     0.1833230
## 14     0.1192530
## 15     0.3528240
```

Plotting the histogram of distribution we can observe the exponential random variable

```
hist(dfexponentialrv$exponentialrv, breaks=40, xlab = 'Time', freq = FALSE, main='Histogram Dist
ribution of time arrivals')
```



A Homogenous Poisson process has a constant rate parameter λ while a Non-Homogenous Poisson process can have a variable rate parameter $\lambda(t)$ that is the function of time.$P(X) = \frac{\lambda^x e^{-\lambda}}{x!}$

In order to prove that this data set is similar to the Homogenous Poisson process, we can prove by common methods of construction to provide an estimate of λ and then make comparisons of the estimates.

Method of Moments

We can calculate λ by:

$$E(X) = \frac{1}{\lambda_n}$$

by isolating for λ_n formula changes to $\lambda_n = \frac{1}{E(X)}$

similarly its known that $E(X) = \frac{1}{n}\sum_{i=1}^{n}(X_i) = \tilde{X_n}$

thus the calculation results

```
mmlambda <- (1/mean(dfexponentialrv$exponentialrv))
mmlambda
```

```
## [1] 3.971419
```

Method: Maximum Likelihood

Maximum likelihood to estimate λ is solved with this formula:

$$l(\lambda) = \lambda^n \mathrm{e}{-\lambda} \sum_{i=1}^{n} x_i$$

As we are trying to prove Exponential function we use the following formula

$$ln(l(\lambda)) = n \ln(\lambda) - \lambda \sum_{i=1}^{n} x_i$$

To further explain the critical points the derivative of the function equates to 0 $ln(l(\lambda_n) = 0$ therefore we can solve for λ with $\hat{\lambda} = \frac{n}{\sum_{i=1} x_i} = \frac{1}{\tilde{X}}$

To verify that the critical points involve a maximum its computed: $ < 0$, in order to assure that critical point is related to the maximum

Thus,

$$\hat{\lambda_n} = \frac{1}{\tilde{X_n}}$$

is the estimator of $lambda$ by maximum likelihood, as observed this is the exact same result as the previous method of moments explained.

Kolmogorov-Smirnov test

By computing the Kolmogorov-Smirnov test [ks.test() function in R], this is anon-parametric test of the equality of discontinuous and continuous of a one dimension probability distribution that is used to compare the sample with the reference probability test. By setting $\alpha = 0.05$, At this instance, the H_0 (null hypothesis) states that there is no difference between two distributions.

```
ks.test(dfexponentialrv$exponentialrv, "pexp", mmlambda)
```

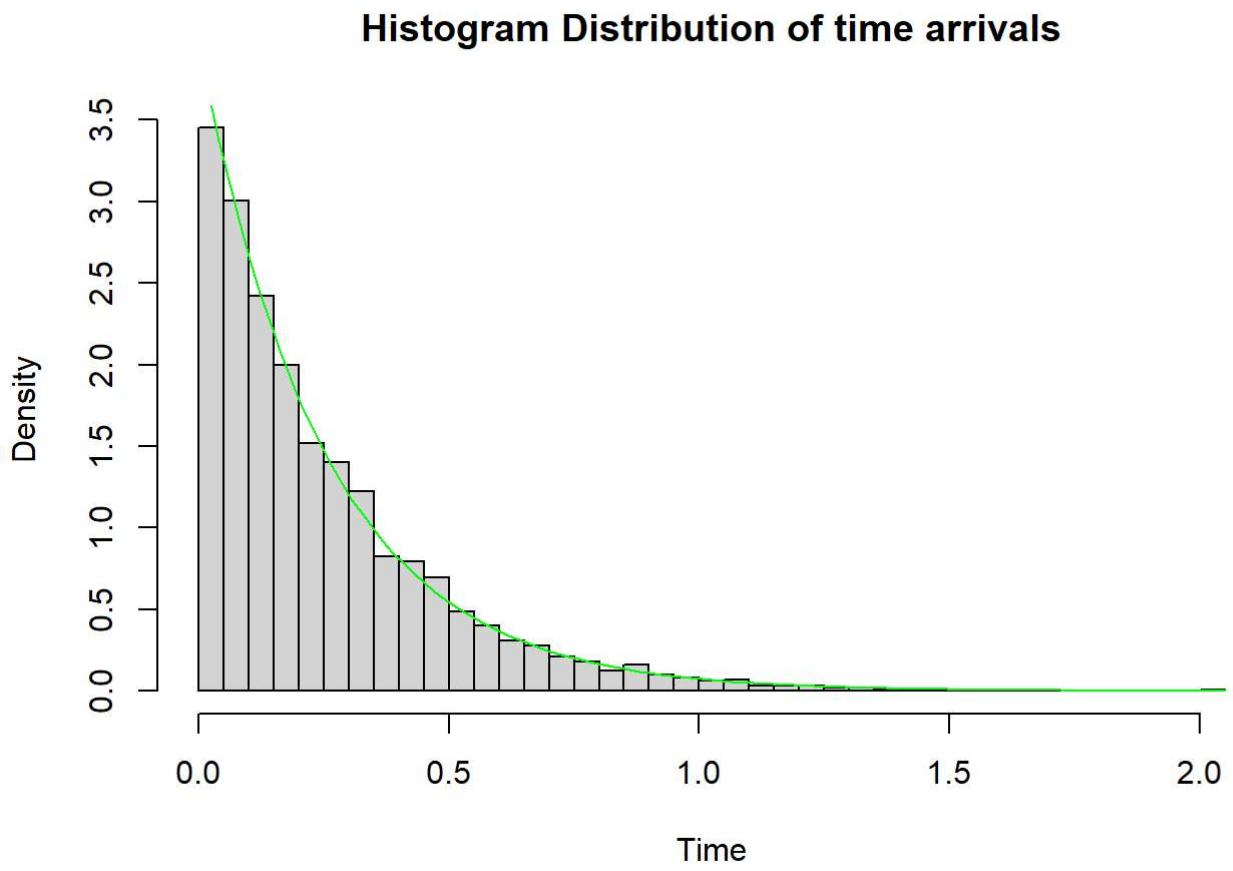
```
## Warning in ks.test(dfexponentialrv$exponentialrv, "pexp", mmlambda): ties should
## not be present for the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  dfexponentialrv$exponentialrv
## D = 0.011477, p-value = 0.5568
## alternative hypothesis: two-sided
```

Observing that that the P-value results to 0.5568, it can be concluded to accept the Null hypothesis (> 0.05), in this instance exponential distribution is present.

To visualize this a curve with rate of λ obtained in the previous tests is added to the previous histogram.

```
hist(dfexponentialrv$exponentialrv, breaks=40, xlab = 'Time', freq = FALSE, main='Histogram Dist
ribution of time arrivals')
curve(dexp(x, rate = mmlambda), from = 0, col = "green", add = TRUE)
```



Exercise 2:

Importing of the files ukcomp1_r.dat and ukcomp2_r.daton to the Rstudio platform. It is noted that ukcomp1_r.dat is the training set and ukcomp2_r.dat is the test set.

```
trainukcomp <- read.table("ukcomp1_r.dat", sep="",dec=".", header = TRUE, check.names = FALSE)
testukcomp <- read.table("ukcomp2_r.dat", sep="",dec=".", header = TRUE, check.names = FALSE)
```

Observing the data set to make sure it has been loaded successfully

Training Data Set

```
str(trainukcomp)
```

```
## 'data.frame':    40 obs. of  13 variables:
##  $ RETCAP : num  0.26 0.57 0.09 0.32 0.17 0.24 0.53 0.26 0.13 0.16 ...
##  $ GEARRAT: num  0.46 0 0.24 0.45 0.91 0.26 0.52 0.24 0.19 0.29 ...
##  $ CAPINT : num  0.64 1.79 0.36 1.86 1.26 1.54 3.34 1.38 0.91 1.7 ...
##  $ WCFTDT : num  0.25 0.33 0.2 0.21 0.12 0.25 0.4 0.37 0.21 0.18 ...
##  $ LOGSALE: num  4.11 4.25 4.44 4.71 4.85 5.61 4.83 4.49 4.13 4.4 ...
##  $ LOGASST: num  4.3 4 4.88 4.44 4.75 5.42 4.3 4.35 4.17 4.17 ...
##  $ CURRAT : num  1.53 1.73 0.44 1.23 1.76 1.44 0.83 1.45 2.89 2.13 ...
##  $ QUIKRAT: num  0.18 1.26 0.39 0.69 0.9 1.23 0.83 0.58 1.95 0.56 ...
##  $ NFATAST: num  0.1 0.12 0.94 0.29 0.26 0.42 0.14 0.4 0.06 0.21 ...
##  $ INVTAST: num  0.74 0.27 0.01 0.29 0.33 0.06 0 0.36 0.29 0.58 ...
##  $ FATTOT : num  0.12 0.15 0.97 0.52 0.54 0.57 0.21 1.04 0.11 0.4 ...
##  $ PAYOUT : num  0.07 0.3 0.57 0 0.31 0.15 0.21 0.16 0.39 0.46 ...
##  $ WCFTCL : num  0.25 0.33 0.5 0.23 0.21 0.37 0.59 0.44 0.21 0.21 ...
```

```
head(trainukcomp, 10)
```

```
##    RETCAP GEARRAT CAPINT WCFTDT LOGSALE LOGASST CURRAT QUIKRAT NFATAST INVTAST
## 1    0.26    0.46   0.64   0.25    4.11    4.30   1.53    0.18    0.10    0.74
## 2    0.57    0.00   1.79   0.33    4.25    4.00   1.73    1.26    0.12    0.27
## 3    0.09    0.24   0.36   0.20    4.44    4.88   0.44    0.39    0.94    0.01
## 4    0.32    0.45   1.86   0.21    4.71    4.44   1.23    0.69    0.29    0.29
## 5    0.17    0.91   1.26   0.12    4.85    4.75   1.76    0.90    0.26    0.33
## 6    0.24    0.26   1.54   0.25    5.61    5.42   1.44    1.23    0.42    0.06
## 7    0.53    0.52   3.34   0.40    4.83    4.30   0.83    0.83    0.14    0.00
## 8    0.26    0.24   1.38   0.37    4.49    4.35   1.45    0.58    0.40    0.36
## 9    0.13    0.19   0.91   0.21    4.13    4.17   2.89    1.95    0.06    0.29
## 10   0.16    0.29   1.70   0.18    4.40    4.17   2.13    0.56    0.21    0.58
##    FATTOT PAYOUT WCFTCL
## 1    0.12   0.07   0.25
## 2    0.15   0.30   0.33
## 3    0.97   0.57   0.50
## 4    0.52   0.00   0.23
## 5    0.54   0.31   0.21
## 6    0.57   0.15   0.37
## 7    0.21   0.21   0.59
## 8    1.04   0.16   0.44
## 9    0.11   0.39   0.21
## 10   0.40   0.46   0.21
```

```
which(is.na(trainukcomp))#No Null values present in the dataset1 file
```

```
## integer(0)
```

Test Data set

```
str(testukcomp)
```

```
## 'data.frame':    40 obs. of  13 variables:
##  $ RETCAP : num  0.19 0.22 0.17 0.12 0.21 0.12 0.15 0.1 0.08 0.31 ...
##  $ WCFTCL : num  0.16 0.26 0.26 0.08 0.34 0.25 0.25 0.12 0.04 0.12 ...
##  $ WCFTDT : num  0.16 0.16 0.2 0.08 0.34 0.25 0.16 0.09 0.04 0.11 ...
##  $ GEARRAT: num  0.15 0.54 0.49 0.39 0.11 0.19 0.35 0.39 0.5 0.41 ...
##  $ LOGSALE: num  5.23 4.15 5.38 4.12 4.78 ...
##  $ LOGASST: num  4.84 4.34 4.88 3.93 4.59 ...
##  $ NFATAST: num  0.28 0.13 0.43 0.23 0.3 0.34 0.48 0.26 0.25 0.17 ...
##  $ CAPINT : num  2.47 0.64 3.18 1.55 1.56 1.74 1.39 1.6 1.58 1.88 ...
##  $ FATTOT : num  0.36 0.16 0.74 0.5 0.5 0.38 0.62 0.42 0.33 0.25 ...
##  $ INVTAST: num  0.42 0.04 0.13 0.37 0.2 0.31 0.22 0.3 0.31 0.31 ...
##  $ PAYOUT : num  0.31 0.45 0.5 0.65 0.25 0.8 0.46 1.03 0 0.25 ...
##  $ QUIKRAT: num  0.54 0.83 0.84 0.5 1.1 1 0.73 0.94 0.74 0.66 ...
##  $ CURRAT : num  1.33 0.93 1.09 1.09 1.74 1.89 1.38 1.57 1.28 1.1 ...
```

```
head(testukcomp, 10)
```

```
##    RETCAP WCFTCL WCFTDT GEARRAT LOGSALE LOGASST NFATAST CAPINT FATTOT INVTAST
## 1    0.19   0.16   0.16    0.15  5.2297  4.8375    0.28   2.47   0.36    0.42
## 2    0.22   0.26   0.16    0.54  4.1495  4.3402    0.13   0.64   0.16    0.04
## 3    0.17   0.26   0.20    0.49  5.3831  4.8811    0.43   3.18   0.74    0.13
## 4    0.12   0.08   0.08    0.39  4.1225  3.9333    0.23   1.55   0.50    0.37
## 5    0.21   0.34   0.34    0.11  4.7795  4.5877    0.30   1.56   0.50    0.20
## 6    0.12   0.25   0.25    0.19  4.1503  3.9086    0.34   1.74   0.38    0.31
## 7    0.15   0.25   0.16    0.35  5.6998  5.5577    0.48   1.39   0.62    0.22
## 8    0.10   0.12   0.09    0.39  4.4162  4.2128    0.26   1.60   0.42    0.30
## 9    0.08   0.04   0.04    0.50  4.7108  4.5126    0.25   1.58   0.33    0.31
## 10   0.31   0.12   0.11    0.41  4.4678  4.1928    0.17   1.88   0.25    0.31
##    PAYOUT QUIKRAT CURRAT
## 1    0.31    0.54   1.33
## 2    0.45    0.83   0.93
## 3    0.50    0.84   1.09
## 4    0.65    0.50   1.09
## 5    0.25    1.10   1.74
## 6    0.80    1.00   1.89
## 7    0.46    0.73   1.38
## 8    1.03    0.94   1.57
## 9    0.00    0.74   1.28
## 10   0.25    0.66   1.10
```

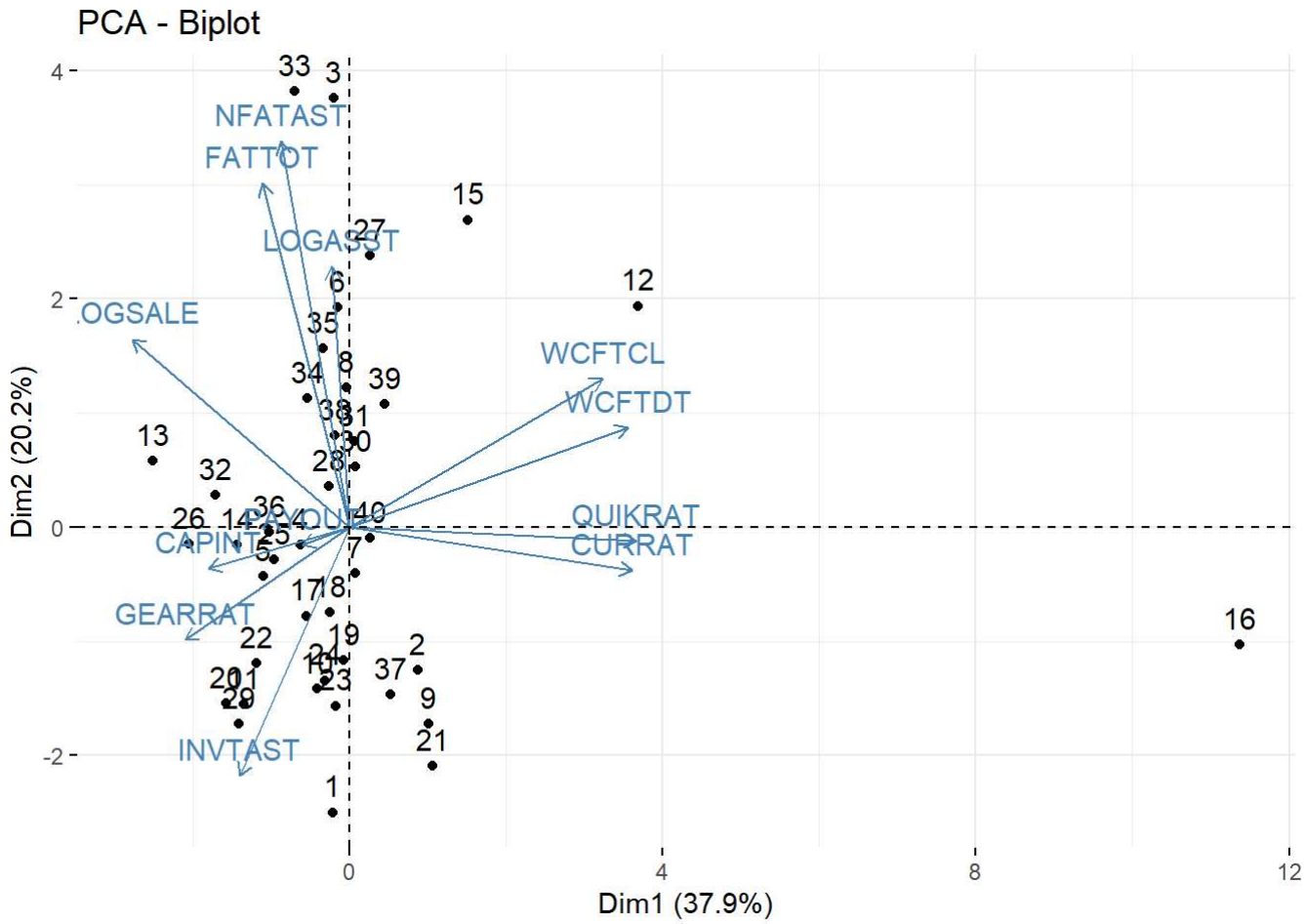
```
which(is.na(testukcomp))#No Null values present in the dataset1 file
```

```
## integer(0)
```

PCA

By looking at the Principal component analysis: Summarizes and visualizes all the most information contained in a multivariate data set. Using the factoextra library.

```
my_data <- trainukcomp[,-1]
res.pca <- prcomp(my_data, scale = TRUE)
fviz_pca(res.pca)
```



In the plot above: Dimensions 1 and 2 retained under 60% of the total information contained in the data set. It is observed that there are about 4 variables that are positively correlated from each other, while the rest are negatively correlated (on the opposite side of the plots)

In order to explain the target variable: *RETCAP*, the variables will be identified, its essential to begin with a Linear regression model.

Linear Regression

The data set contains 13 variables, in this particular context the target variable is a continuous value, explained by 12 other variables.

A full multiple linear regression can be used as a primary step of the variable selection process.

```
linearm_model=lm(RETCAP~., data = trainukcomp)
summary(linearm_model)$coef
```

```
##               Estimate Std. Error    t value    Pr(>|t|)
## (Intercept)  0.18807204 0.13391661  1.4043967 0.171600538
## GEARRAT     -0.04043567 0.07677092 -0.5267056 0.602698806
## CAPINT      -0.01413791 0.02338316 -0.6046193 0.550479704
## WCFTDT       0.30555746 0.29736579  1.0275475 0.313280788
## LOGSALE      0.11843959 0.03611612  3.2794103 0.002865998
## LOGASST     -0.07695988 0.04517414 -1.7036268 0.099935934
## CURRAT      -0.22328066 0.08773480 -2.5449499 0.016956876
## QUIKRAT      0.17670888 0.09162882  1.9285293 0.064365189
## NFATAST     -0.36997685 0.13739742 -2.6927496 0.012024938
## INVTAST      0.25056172 0.18586858  1.3480585 0.188841487
## FATTOT      -0.10098591 0.08764238 -1.1522498 0.259318906
## PAYOUT      -0.01883896 0.01769456 -1.0646753 0.296452093
## WCFTCL       0.21512978 0.19788455  1.0871479 0.286581721
```

This coefficient table shows the beta coefficient estimates and their significance level.

```
summary(linearm_model)
```

```
##
## Call:
## lm(formula = RETCAP ~ ., data = trainukcomp)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.126501 -0.043091 -0.002002  0.036908  0.201047
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.18807    0.13392   1.404  0.17160
## GEARRAT     -0.04044    0.07677  -0.527  0.60270
## CAPINT      -0.01414    0.02338  -0.605  0.55048
## WCFTDT       0.30556    0.29737   1.028  0.31328
## LOGSALE      0.11844    0.03612   3.279  0.00287 **
## LOGASST     -0.07696    0.04517  -1.704  0.09994 .
## CURRAT      -0.22328    0.08773  -2.545  0.01696 *
## QUIKRAT      0.17671    0.09163   1.929  0.06437 .
## NFATAST     -0.36998    0.13740  -2.693  0.01202 *
## INVTAST      0.25056    0.18587   1.348  0.18884
## FATTOT      -0.10099    0.08764  -1.152  0.25932
## PAYOUT      -0.01884    0.01769  -1.065  0.29645
## WCFTCL       0.21513    0.19788   1.087  0.28658
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07441 on 27 degrees of freedom
## Multiple R-squared:  0.7889, Adjusted R-squared:  0.6951
## F-statistic: 8.408 on 12 and 27 DF,  p-value: 2.555e-06
```

As we notice in this Fisher test, the p-value for the entire model is less than $.05$, which indicates that we can reject the H_0 Null hypothesis and make a note that there are some variables present in the model that will explain RETCAP variable. This is because as we look at all the coefficients in this test, the most significant variables

(those that have a p-value which $< .05$) are LOGSALE, CURRAT, and NFATAST.

For observation purposes, its possible to isolate these individual variables and observe their results

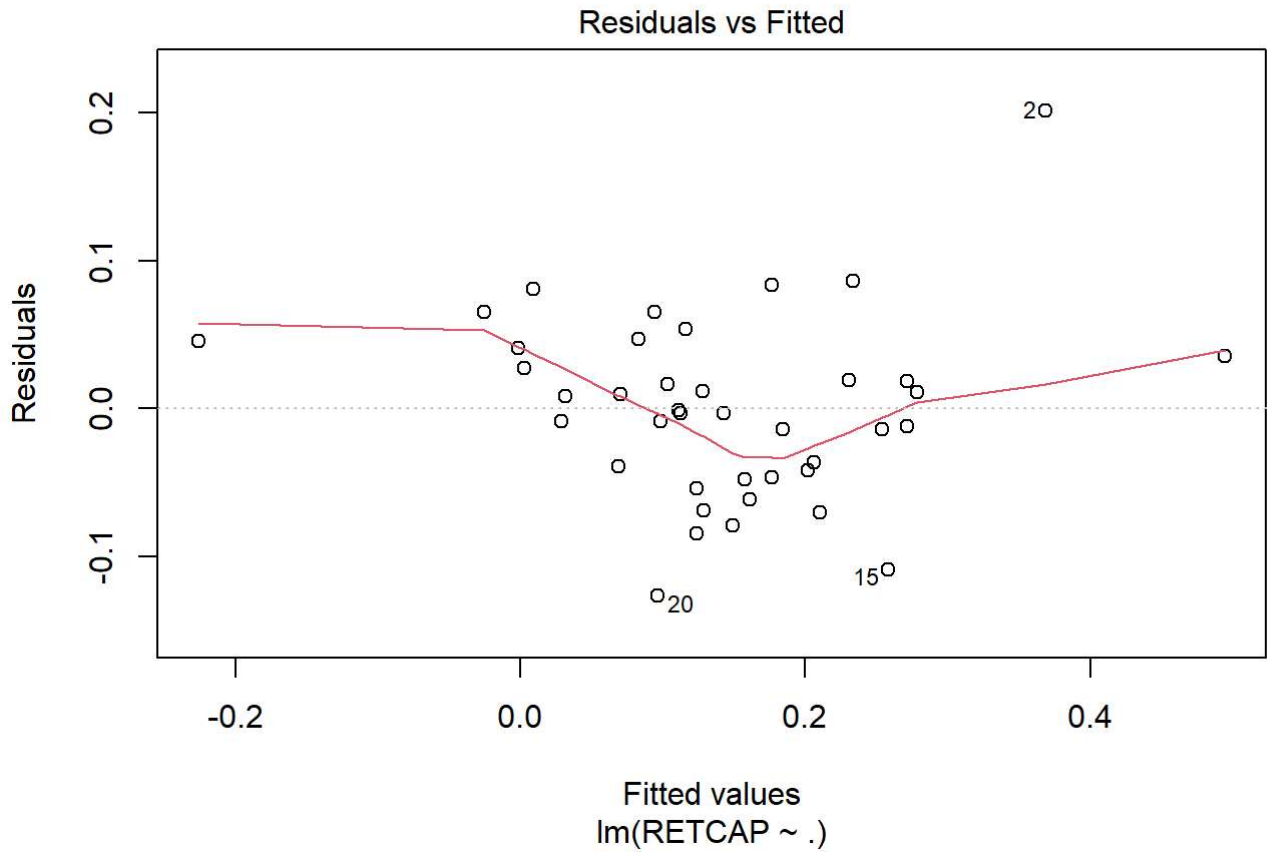
```
lmmodel <- lm(RETCAP ~ LOGSALE + CURRAT + NFATAST, data=trainukcomp)
summary(lmmodel)
```

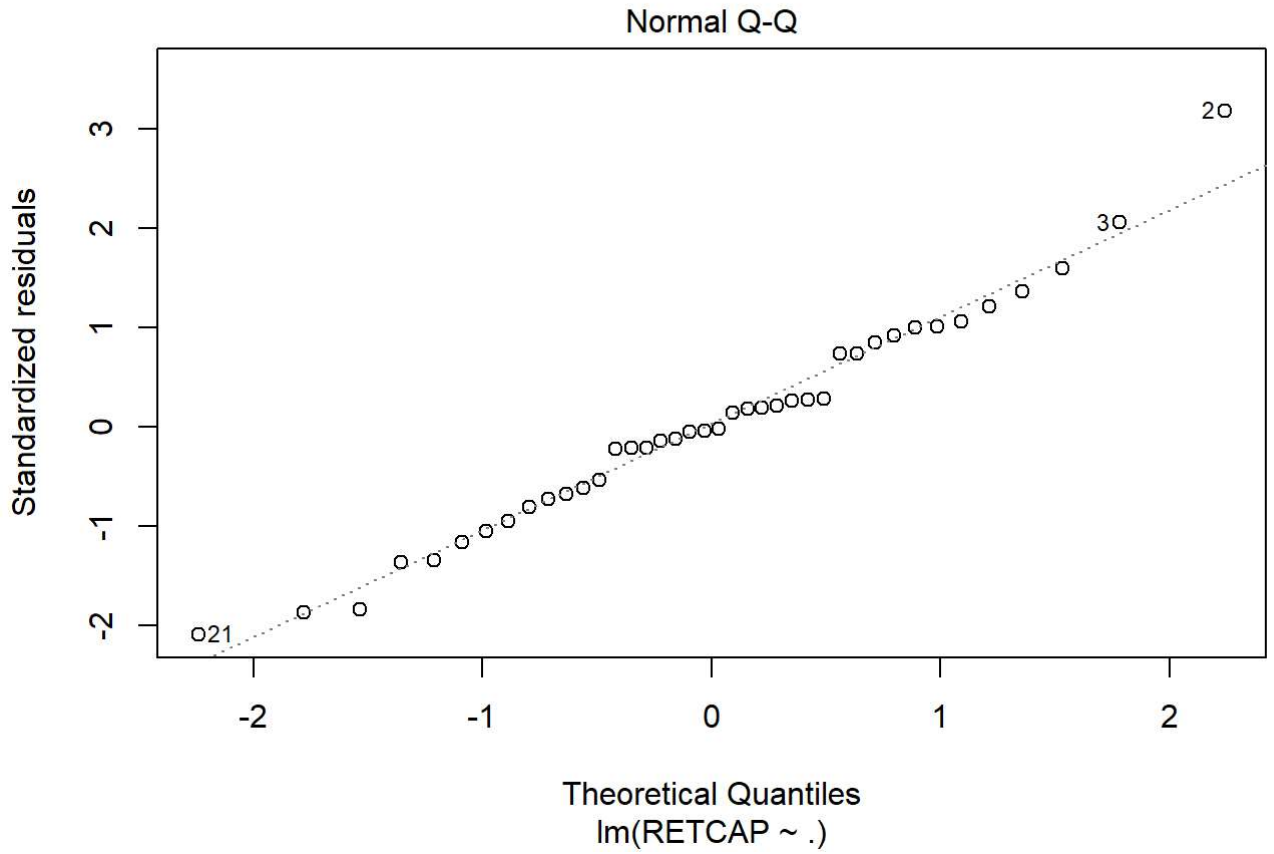
```
##
## Call:
## lm(formula = RETCAP ~ LOGSALE + CURRAT + NFATAST, data = trainukcomp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23100 -0.06266 -0.02689  0.06336  0.36545
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.076780   0.124115  -0.619  0.54006
## LOGSALE      0.071704   0.025291   2.835  0.00747 **
## CURRAT       0.008612   0.012740   0.676  0.50338
## NFATAST     -0.319272   0.105816  -3.017  0.00466 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1188 on 36 degrees of freedom
## Multiple R-squared:  0.2826, Adjusted R-squared:  0.2228
## F-statistic: 4.727 on 3 and 36 DF,  p-value: 0.006998
```

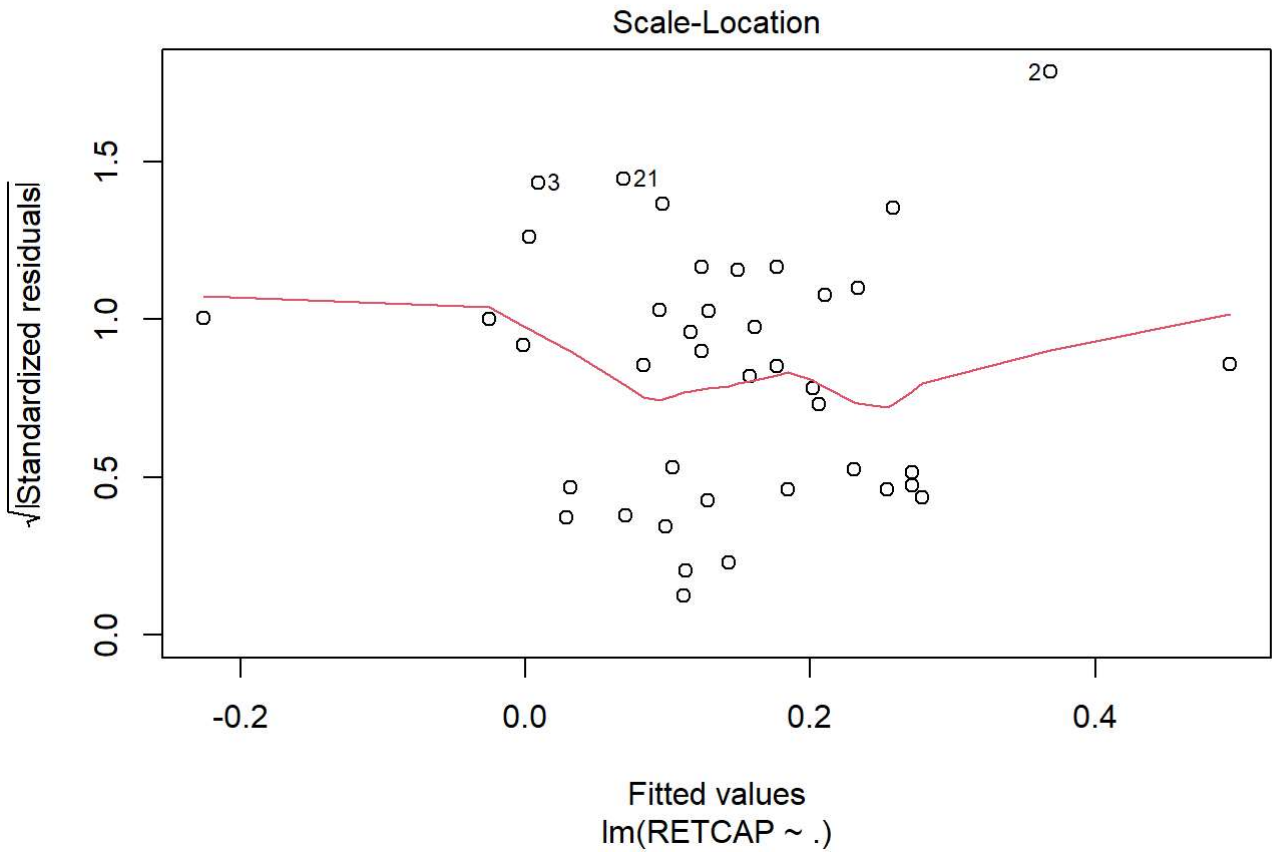
As seen here, although the p-value for this model is low, the adjusted R2 is close to 0, which indicates that the regression model did not explain much of the variability in the outcome.

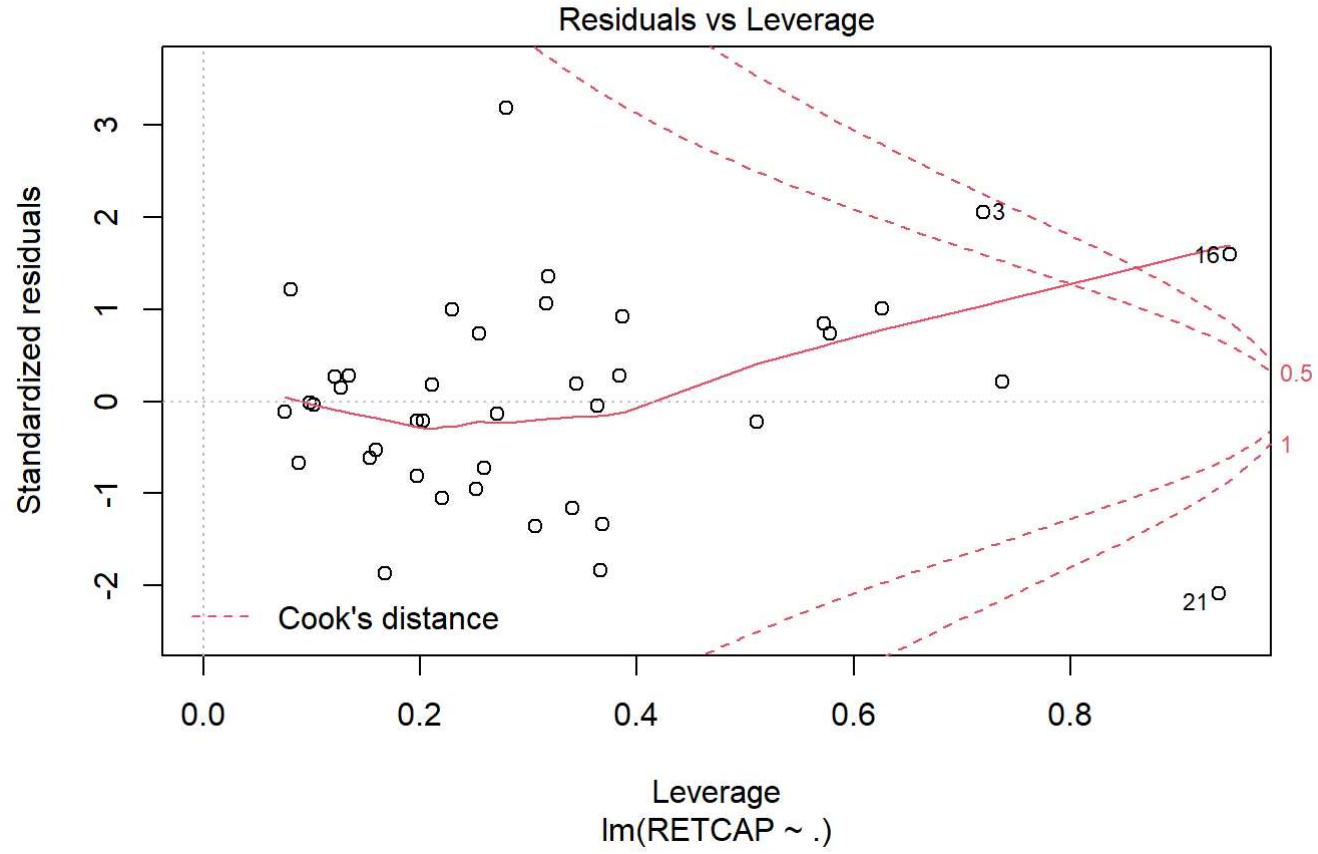
It is essential to make sure that our model agrees with the conditions of the gaussianity of the noise and that variance of noise is constant, so we use the plot() function in order to visualize the QQplots and Residiuals vs fitted values.

```
plot(linearm_model)
```


Residuals vs Fitted
Residuals
0.2
0.1
0.0
-0.1
-0.2
0.0
0.2
0.4
2
15
20
Fitted values
lm(RETCAP ~ .)


Normal Q-Q
Standardized residuals
3
2
1
0
-1
-2
-2
-1
0
1
2
2
3
21
Theoretical Quantiles
lm(RETCAP ~ .)





As observed with the graphs above, they all meet the conditions set in this model.

After linear models computed it is essential to go through a variable selection process in order to get an idea which variables are important that can explain RETCAP

Variable Selection

Stepwise regression can be used by removing predictors in the predictive model, in order to find a subset of variables in the data resulting as the best best performing model that yields the lowest prediction error in this model.

In this context we will use 2 methods, the Backward Selection (removes the least contributive predictors, and stops when all predictors are statistically signficant) and Stepwise Selection (combination of forward and backward selection)

Backward

```
backwarduk1 <- stepAIC(linearm_model,~., direction="backward", tests="F")
```

```
## Start:  AIC=-197.57
## RETCAP ~ GEARRAT + CAPINT + WCFTDT + LOGSALE + LOGASST + CURRAT +
##     QUIKRAT + NFATAST + INVTAST + FATTOT + PAYOUT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC
## - GEARRAT  1  0.001536 0.15105 -199.16
## - CAPINT   1  0.002024 0.15154 -199.03
## - WCFTDT   1  0.005847 0.15536 -198.03
## - PAYOUT   1  0.006277 0.15579 -197.93
## - WCFTCL   1  0.006545 0.15606 -197.86
## - FATTOT   1  0.007352 0.15687 -197.65
## <none>                 0.14951 -197.57
## - INVTAST  1  0.010063 0.15958 -196.96
## - LOGASST  1  0.016072 0.16559 -195.49
## - QUIKRAT  1  0.020595 0.17011 -194.41
## - CURRAT   1  0.035865 0.18538 -190.97
## - NFATAST  1  0.040152 0.18967 -190.06
## - LOGSALE  1  0.059554 0.20907 -186.16
##
## Step:  AIC=-199.16
## RETCAP ~ CAPINT + WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT +
##     NFATAST + INVTAST + FATTOT + PAYOUT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC
## - CAPINT   1  0.002588 0.15364 -200.48
## - PAYOUT   1  0.005092 0.15614 -199.84
## - WCFTCL   1  0.005567 0.15662 -199.71
## - FATTOT   1  0.007599 0.15865 -199.20
## <none>                 0.15105 -199.16
## - INVTAST  1  0.008574 0.15962 -198.95
## - WCFTDT   1  0.009823 0.16087 -198.64
## - LOGASST  1  0.016709 0.16776 -196.96
## - QUIKRAT  1  0.019187 0.17024 -196.38
## - CURRAT   1  0.034549 0.18560 -192.92
## - NFATAST  1  0.040106 0.19116 -191.74
## - LOGSALE  1  0.058563 0.20961 -188.06
##
## Step:  AIC=-200.48
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     INVTAST + FATTOT + PAYOUT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC
## - PAYOUT   1  0.004966 0.15860 -201.21
## - WCFTCL   1  0.006079 0.15972 -200.93
## - INVTAST  1  0.007156 0.16079 -200.66
## <none>                 0.15364 -200.48
## - FATTOT   1  0.008379 0.16202 -200.36
## - WCFTDT   1  0.008907 0.16254 -200.23
## - LOGASST  1  0.015302 0.16894 -198.68
## - QUIKRAT  1  0.016604 0.17024 -198.38
## - CURRAT   1  0.032326 0.18596 -194.84
## - NFATAST  1  0.037518 0.19116 -193.74
## - LOGSALE  1  0.085310 0.23895 -184.82
```

```
##
## Step:  AIC=-201.21
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     INVTAST + FATTOT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC
## - WCFTCL   1  0.004794 0.16340 -202.02
## - FATTOT   1  0.006933 0.16554 -201.50
## - INVTAST  1  0.007420 0.16602 -201.38
## <none>                 0.15860 -201.21
## - WCFTDT   1  0.011401 0.17000 -200.43
## - LOGASST  1  0.013470 0.17207 -199.95
## - QUIKRAT  1  0.018358 0.17696 -198.83
## - CURRAT   1  0.035686 0.19429 -195.09
## - NFATAST  1  0.043176 0.20178 -193.58
## - LOGSALE  1  0.083618 0.24222 -186.27
##
## Step:  AIC=-202.02
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     INVTAST + FATTOT
##
##           Df Sum of Sq     RSS     AIC
## - INVTAST  1  0.006559 0.16996 -202.44
## <none>                 0.16340 -202.02
## - FATTOT   1  0.008397 0.17179 -202.01
## - LOGASST  1  0.010104 0.17350 -201.62
## - QUIKRAT  1  0.015986 0.17938 -200.28
## - CURRAT   1  0.034396 0.19779 -196.38
## - NFATAST  1  0.040969 0.20437 -195.07
## - LOGSALE  1  0.082492 0.24589 -187.67
## - WCFTDT   1  0.310122 0.47352 -161.46
##
## Step:  AIC=-202.44
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     FATTOT
##
##           Df Sum of Sq     RSS     AIC
## <none>                 0.16996 -202.44
## - FATTOT   1  0.009059 0.17902 -202.37
## - QUIKRAT  1  0.011787 0.18174 -201.76
## - LOGASST  1  0.015575 0.18553 -200.94
## - NFATAST  1  0.045986 0.21594 -194.87
## - CURRAT   1  0.048877 0.21883 -194.33
## - LOGSALE  1  0.084438 0.25439 -188.31
## - WCFTDT   1  0.303564 0.47352 -163.46
```

```
summary(backwarduk1)
```

```
##
## Call:
## lm(formula = RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT +
##     NFATAST + FATTOT, data = trainukcomp)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.130281 -0.044026  0.002847  0.029266  0.216228
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.14802    0.10075   1.469 0.151548
## WCFTDT       0.61237    0.08100   7.560  1.3e-08 ***
## LOGSALE      0.09899    0.02483   3.987 0.000363 ***
## LOGASST     -0.05548    0.03240  -1.712 0.096484 .
## CURRAT      -0.12169    0.04011  -3.034 0.004767 **
## QUIKRAT      0.06089    0.04087   1.490 0.146087
## NFATAST     -0.37365    0.12698  -2.943 0.006012 **
## FATTOT      -0.10996    0.08420  -1.306 0.200866
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07288 on 32 degrees of freedom
## Multiple R-squared:   0.76,  Adjusted R-squared:  0.7075
## F-statistic: 14.48 on 7 and 32 DF,  p-value: 2.546e-08
```

the most significant variables are WCFTDT, LOGSALE, CURRAT, NFATAST

STEPWISE SELECTION

```
stepwiseuk1 <- stepAIC(linearm_model,~., direction="both", tests="F")
```

```
## Start:  AIC=-197.57
## RETCAP ~ GEARRAT + CAPINT + WCFTDT + LOGSALE + LOGASST + CURRAT +
##     QUIKRAT + NFATAST + INVTAST + FATTOT + PAYOUT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC
## - GEARRAT  1  0.001536 0.15105 -199.16
## - CAPINT   1  0.002024 0.15154 -199.03
## - WCFTDT   1  0.005847 0.15536 -198.03
## - PAYOUT   1  0.006277 0.15579 -197.93
## - WCFTCL   1  0.006545 0.15606 -197.86
## - FATTOT   1  0.007352 0.15687 -197.65
## <none>                 0.14951 -197.57
## - INVTAST  1  0.010063 0.15958 -196.96
## - LOGASST  1  0.016072 0.16559 -195.49
## - QUIKRAT  1  0.020595 0.17011 -194.41
## - CURRAT   1  0.035865 0.18538 -190.97
## - NFATAST  1  0.040152 0.18967 -190.06
## - LOGSALE  1  0.059554 0.20907 -186.16
##
## Step:  AIC=-199.16
## RETCAP ~ CAPINT + WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT +
##     NFATAST + INVTAST + FATTOT + PAYOUT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC
## - CAPINT   1  0.002588 0.15364 -200.48
## - PAYOUT   1  0.005092 0.15614 -199.84
## - WCFTCL   1  0.005567 0.15662 -199.71
## - FATTOT   1  0.007599 0.15865 -199.20
## <none>                 0.15105 -199.16
## - INVTAST  1  0.008574 0.15962 -198.95
## - WCFTDT   1  0.009823 0.16087 -198.64
## + GEARRAT  1  0.001536 0.14951 -197.57
## - LOGASST  1  0.016709 0.16776 -196.96
## - QUIKRAT  1  0.019187 0.17024 -196.38
## - CURRAT   1  0.034549 0.18560 -192.92
## - NFATAST  1  0.040106 0.19116 -191.74
## - LOGSALE  1  0.058563 0.20961 -188.06
##
## Step:  AIC=-200.48
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     INVTAST + FATTOT + PAYOUT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC
## - PAYOUT   1  0.004966 0.15860 -201.21
## - WCFTCL   1  0.006079 0.15972 -200.93
## - INVTAST  1  0.007156 0.16079 -200.66
## <none>                 0.15364 -200.48
## - FATTOT   1  0.008379 0.16202 -200.36
## - WCFTDT   1  0.008907 0.16254 -200.23
## + CAPINT   1  0.002588 0.15105 -199.16
## + GEARRAT  1  0.002100 0.15154 -199.03
## - LOGASST  1  0.015302 0.16894 -198.68
## - QUIKRAT  1  0.016604 0.17024 -198.38
```

```
## - CURRAT   1  0.032326 0.18596 -194.84
## - NFATAST  1  0.037518 0.19116 -193.74
## - LOGSALE  1  0.085310 0.23895 -184.82
##
## Step:  AIC=-201.21
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     INVTAST + FATTOT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC
## - WCFTCL   1  0.004794 0.16340 -202.02
## - FATTOT   1  0.006933 0.16554 -201.50
## - INVTAST  1  0.007420 0.16602 -201.38
## <none>                 0.15860 -201.21
## + PAYOUT   1  0.004966 0.15364 -200.48
## - WCFTDT   1  0.011401 0.17000 -200.43
## - LOGASST  1  0.013470 0.17207 -199.95
## + CAPINT   1  0.002462 0.15614 -199.84
## + GEARRAT  1  0.000643 0.15796 -199.37
## - QUIKRAT  1  0.018358 0.17696 -198.83
## - CURRAT   1  0.035686 0.19429 -195.09
## - NFATAST  1  0.043176 0.20178 -193.58
## - LOGSALE  1  0.083618 0.24222 -186.27
##
## Step:  AIC=-202.02
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     INVTAST + FATTOT
##
##           Df Sum of Sq     RSS     AIC
## - INVTAST  1  0.006559 0.16996 -202.44
## <none>                 0.16340 -202.02
## - FATTOT   1  0.008397 0.17179 -202.01
## - LOGASST  1  0.010104 0.17350 -201.62
## + WCFTCL   1  0.004794 0.15860 -201.21
## + PAYOUT   1  0.003681 0.15972 -200.93
## + CAPINT   1  0.002924 0.16047 -200.74
## - QUIKRAT  1  0.015986 0.17938 -200.28
## + GEARRAT  1  0.000217 0.16318 -200.07
## - CURRAT   1  0.034396 0.19779 -196.38
## - NFATAST  1  0.040969 0.20437 -195.07
## - LOGSALE  1  0.082492 0.24589 -187.67
## - WCFTDT   1  0.310122 0.47352 -161.46
##
## Step:  AIC=-202.44
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     FATTOT
##
##           Df Sum of Sq     RSS     AIC
## <none>                 0.16996 -202.44
## - FATTOT   1  0.009059 0.17902 -202.37
## + INVTAST  1  0.006559 0.16340 -202.02
## - QUIKRAT  1  0.011787 0.18174 -201.76
## + PAYOUT   1  0.003997 0.16596 -201.40
## + WCFTCL   1  0.003932 0.16602 -201.38
## - LOGASST  1  0.015575 0.18553 -200.94
```

```
## + CAPINT   1  0.001408 0.16855 -200.78
## + GEARRAT  1  0.000167 0.16979 -200.48
## - NFATAST  1  0.045986 0.21594 -194.87
## - CURRAT   1  0.048877 0.21883 -194.33
## - LOGSALE  1  0.084438 0.25439 -188.31
## - WCFTDT   1  0.303564 0.47352 -163.46
```

```
summary(stepwiseuk1)
```

```
##
## Call:
## lm(formula = RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT +
##     NFATAST + FATTOT, data = trainukcomp)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.130281 -0.044026  0.002847  0.029266  0.216228
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.14802    0.10075   1.469 0.151548
## WCFTDT       0.61237    0.08100   7.560  1.3e-08 ***
## LOGSALE      0.09899    0.02483   3.987 0.000363 ***
## LOGASST     -0.05548    0.03240  -1.712 0.096484 .
## CURRAT      -0.12169    0.04011  -3.034 0.004767 **
## QUIKRAT      0.06089    0.04087   1.490 0.146087
## NFATAST     -0.37365    0.12698  -2.943 0.006012 **
## FATTOT      -0.10996    0.08420  -1.306 0.200866
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07288 on 32 degrees of freedom
## Multiple R-squared:   0.76,  Adjusted R-squared:  0.7075
## F-statistic: 14.48 on 7 and 32 DF,  p-value: 2.546e-08
```

The stepwise function as well computes that the most influential variables in this model are WCFTDT, LOGSALE, CURRAT, NFATAST.

For comparative purposes we can also use the Penalized regression models for further variable selection

In this Instance, the Least Absolute Shrinkage, and Selection Operator (also known as Lasso) can be used, this will shrink the regression coefficiants toward zero by penalizing the regression model with a penalty called L-1 norm (this is the sum of th absolute coefficients).

In order to begin computing Lasso, first must find the best lambda value using cross validation

```
x<- model.matrix(RETCAP~., trainukcomp)[,-1] #Predictor Variables
y <- trainukcomp$RETCAP #Explained Variable

cv <- cv.glmnet(x, y, alpha=1, nfolds = 10)
cv$lambda.min
```

```
## [1] 0.007380247
```

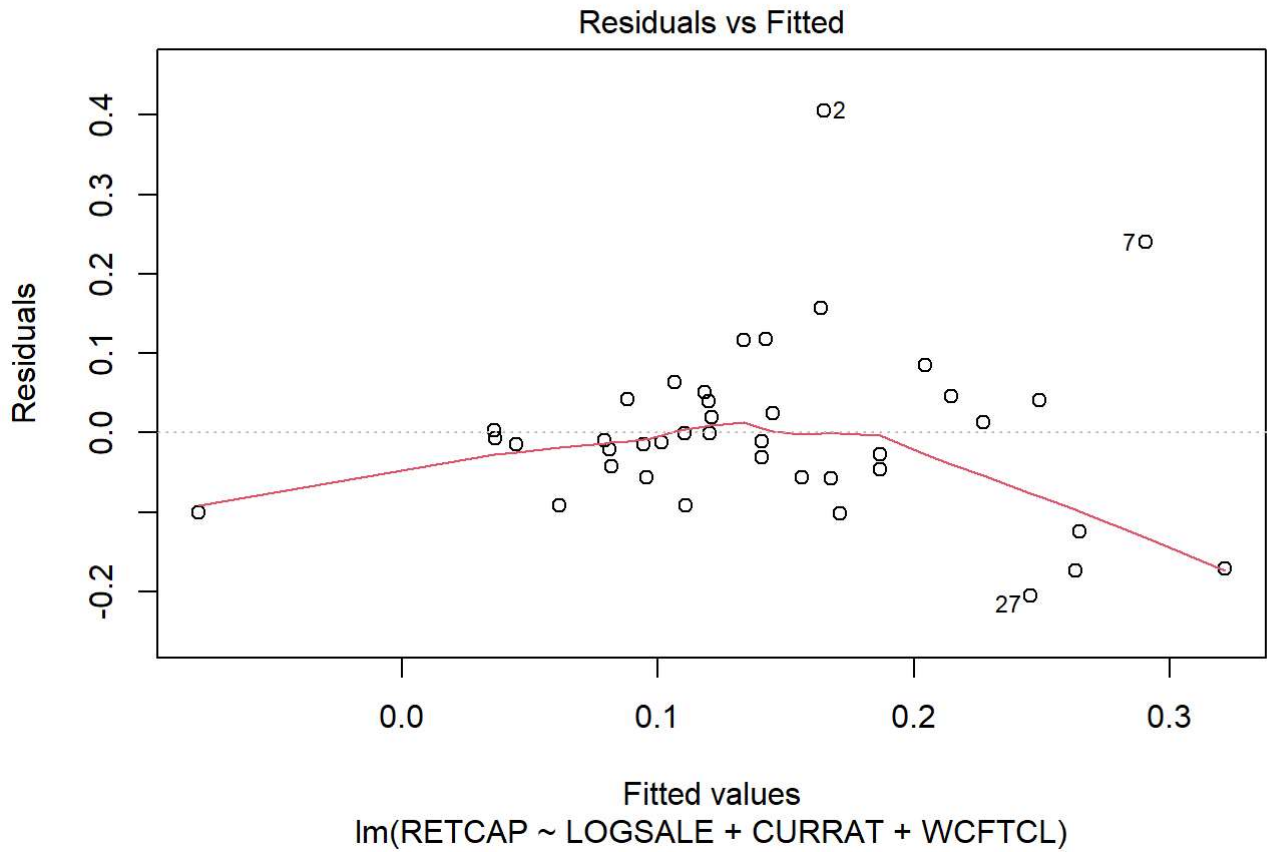
```
lassomodel <- glmnet(x, y, alpha = 1, lambda = cv$lambda.min)
coef(lassomodel)
```

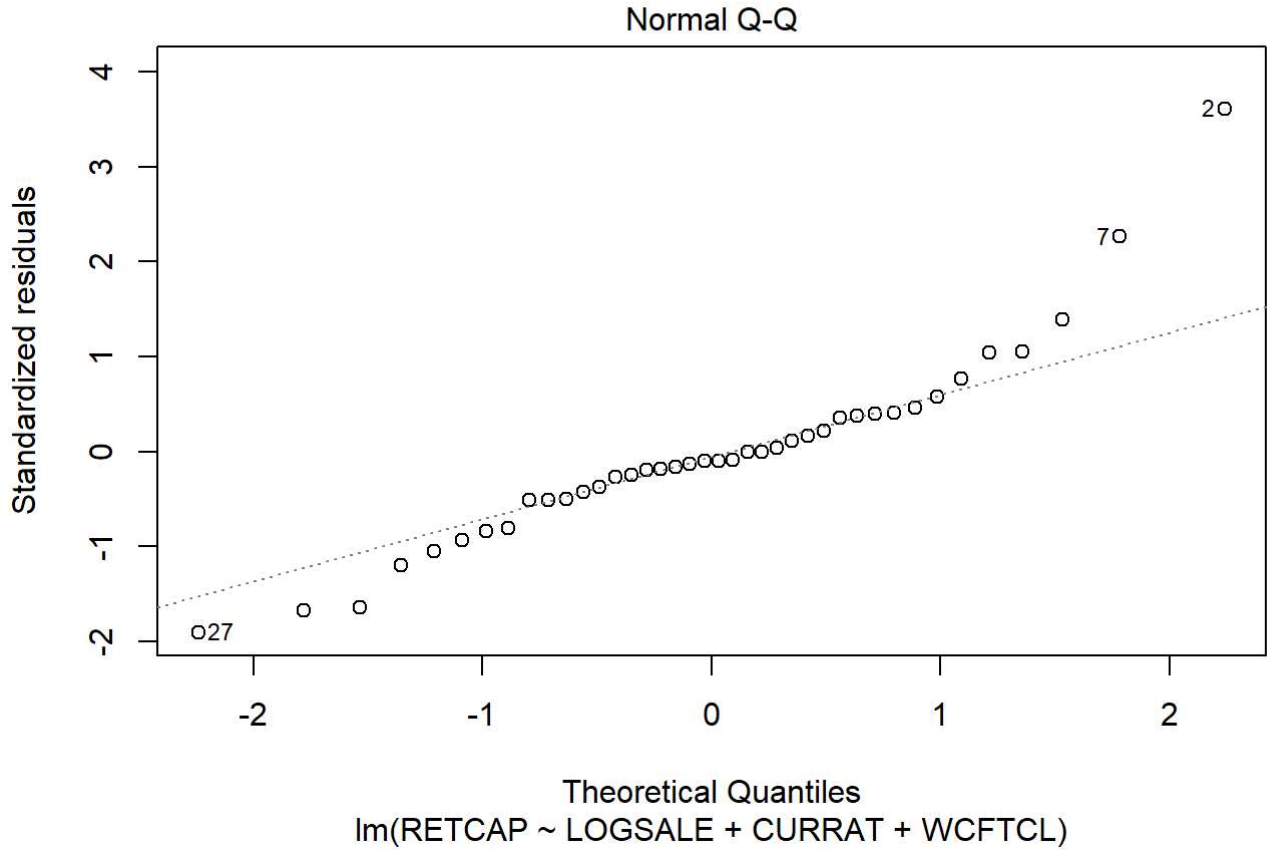
```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept)  0.03946597
## GEARRAT     -0.01249442
## CAPINT       0.02217321
## WCFTDT       .
## LOGSALE      0.04101398
## LOGASST      .
## CURRAT      -0.03070340
## QUIKRAT      .
## NFATAST     -0.27761455
## INVTAST      .
## FATTOT      -0.05518533
## PAYOUT      -0.01094732
## WCFTCL       0.30413698
```

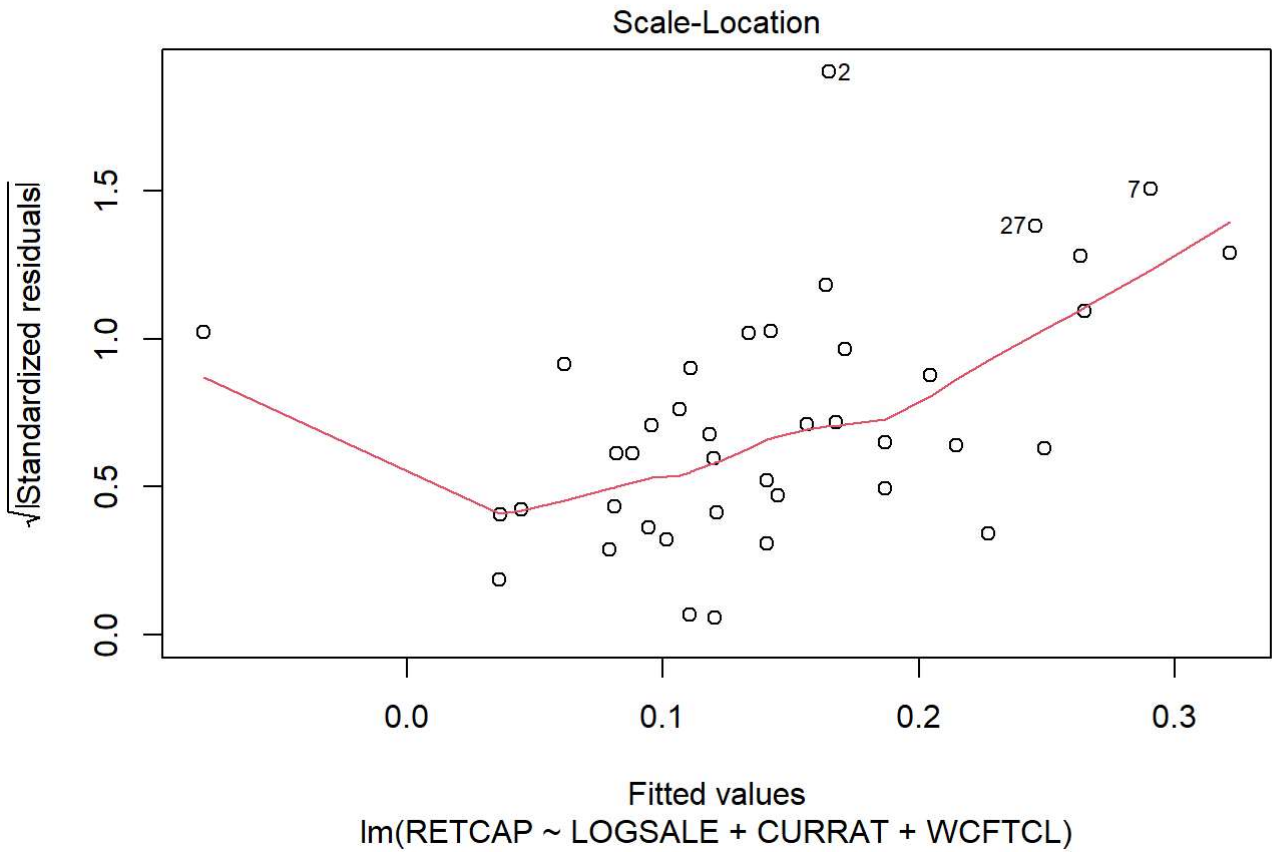
```
lassomodelm <- lm(RETCAP ~ LOGSALE + CURRAT + NFATAST + GEARRAT + CAPINT + PAYOUT + WCFTCL + FAT
TOT, data=trainukcomp)
summary(lassomodelm)
```

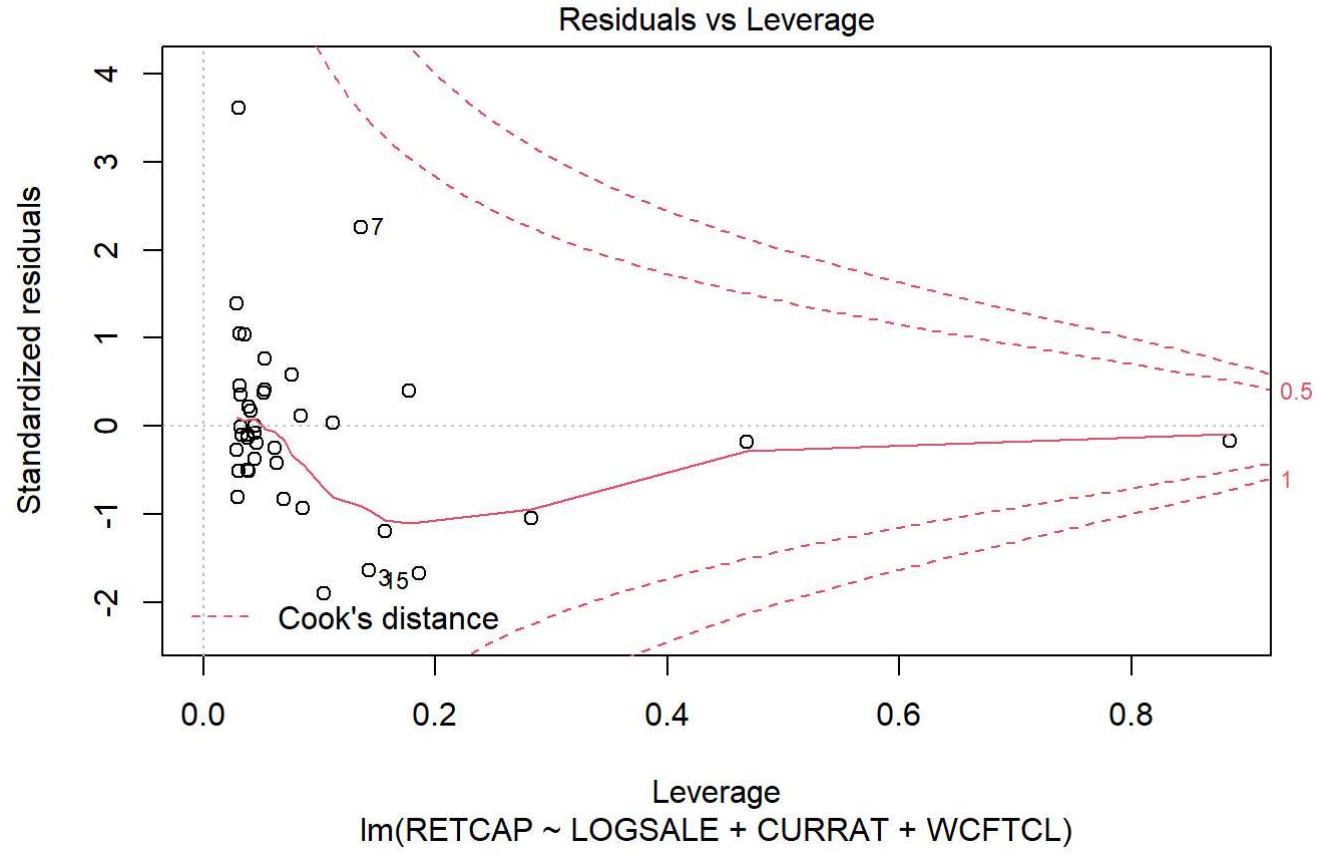
```
##
## Call:
## lm(formula = RETCAP ~ LOGSALE + CURRAT + NFATAST + GEARRAT +
##     CAPINT + PAYOUT + WCFTCL + FATTOT, data = trainukcomp)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.126438 -0.039070  0.000141  0.029937  0.269654
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.07474    0.08680   0.861 0.395823
## LOGSALE      0.04456    0.01850   2.409 0.022128 *
## CURRAT      -0.04692    0.01232  -3.809 0.000619 ***
## NFATAST     -0.32518    0.13098  -2.483 0.018663 *
## GEARRAT     -0.04271    0.07052  -0.606 0.549133
## CAPINT       0.02689    0.01801   1.494 0.145400
## PAYOUT      -0.02219    0.01797  -1.235 0.226205
## WCFTCL       0.39095    0.07141   5.475  5.5e-06 ***
## FATTOT      -0.09384    0.08827  -1.063 0.295932
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07902 on 31 degrees of freedom
## Multiple R-squared:  0.7267, Adjusted R-squared:  0.6562
## F-statistic:  10.3 on 8 and 31 DF,  p-value: 6.788e-07
```

```
lassomodelm2 <- lm(RETCAP ~ LOGSALE + CURRAT +  WCFTCL, data=trainukcomp)
plot(lassomodelm2)
```


Residuals vs Fitted
Residuals
0.4
0.3
0.2
0.1
0.0
-0.2
2
7
27
0.0
0.1
0.2
0.3
Fitted values
lm(RETCAP ~ LOGSALE + CURRAT + WCFTCL)


Normal Q-Q
Standardized residuals
4
3
2
1
0
-1
-2
2
7
27
-2
-1
0
1
2
Theoretical Quantiles
lm(RETCAP ~ LOGSALE + CURRAT + WCFTCL)

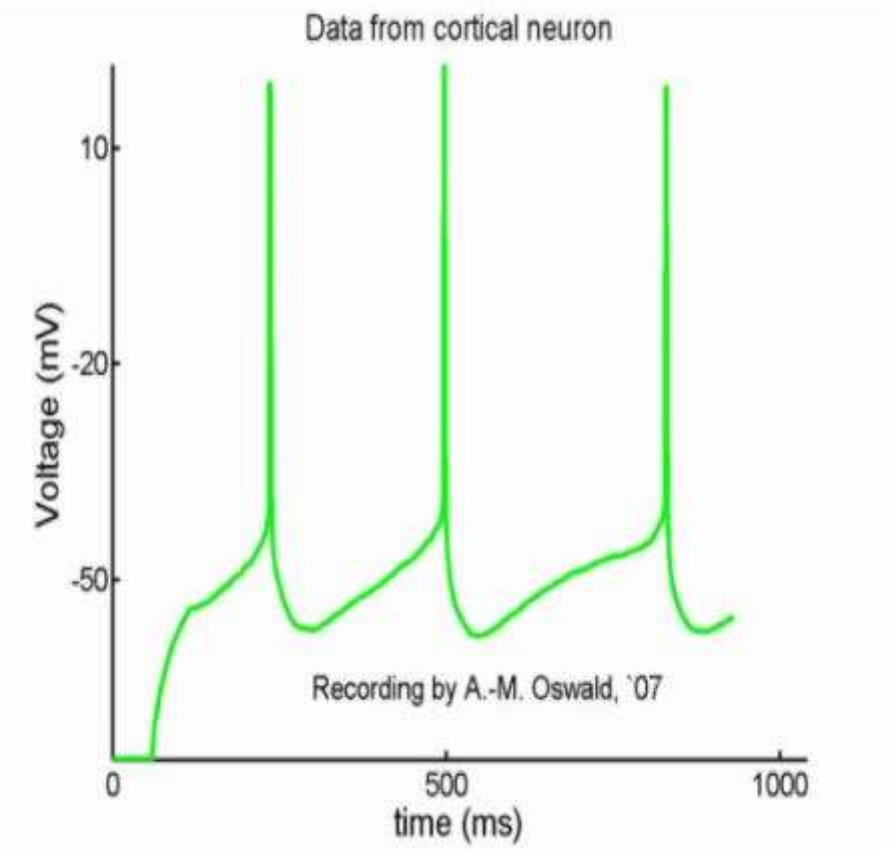

Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2
7
27
0.0
0.1
0.2
0.3
Fitted values
lm(RETCAP ~ LOGSALE + CURRAT + WCFTCL)


Residuals vs Leverage
Standardized residuals
-2
-1
0
1
2
3
4
7
35
Cook's distance
0.5
1
0.0
0.2
0.4
0.6
0.8
Leverage
lm(RETCAP ~ LOGSALE + CURRAT + WCFTCL)

Performing another linear regression model on the variables obtained in the Lasso Regression model, confirms with the previous variable selection methods performed (Stepwise and Backward selection) that the most essential variables that explains RETCAP are WCFTDT, LOGSALE, CURRAT, and NFATAST The best performing model for variable selection in this instance were Stepwise and Backward selection as they both yielded higher Adjusted R2 at .70

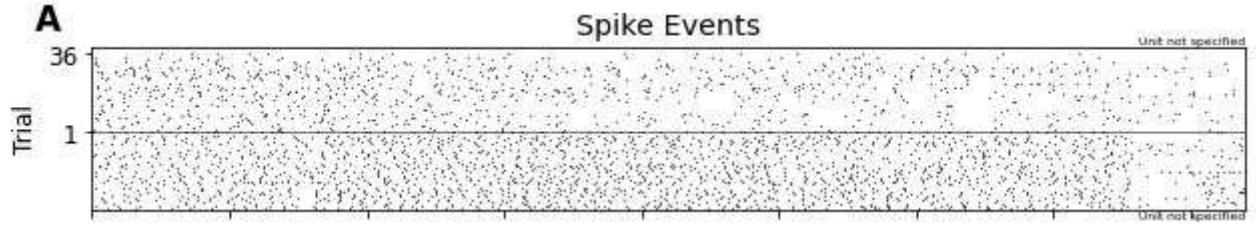
Exercise 3:

The Multiple tests based on Gaussian approximation of the Unitary Events (MTGAUE) method with delayed coincidence count is an improvement to the existing Unitary events (UE) method. This method has been used in the previous decades to detect patterns of coincident joint spike activity among simultaneously recorded neurons.



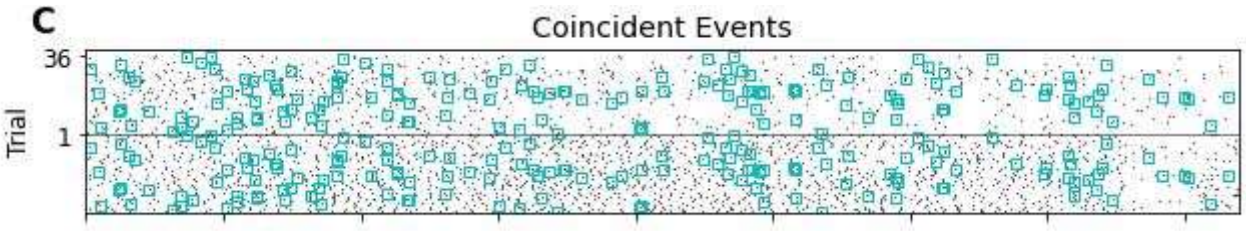
Spike train of a neuron voltage over time

In the field of neuroscience, neurons are made up of made up of intracellular and extracellular membranes, their main function are conducting charges across the entire membrane just like a battery. During a spike, sudden electrical impulses are shot through one brain cell on its way to the next. Spikes are known as the carrier of information in the brain, which drives what we think and do as human beings. Understanding how neural networks work and how neurons work in tangent with each other is to produce these spikes is one of the biggest questions in the twenty-first century in neuroscience and mathematics The prior technique of spike detection, UE analysis methods detects spicuous patterns of synchronous spike among simultaneous recorded single neurons. Give the firing rates of the neurons, the statistical significance of this pattern is evaluated my comparing the empirical number of occurrences to the number of expected. The focus of the UE method is the proper formulation of the null hypothesis and the derivation of the corresponding count distribution of synchronous spike events (A) used in the significance test.



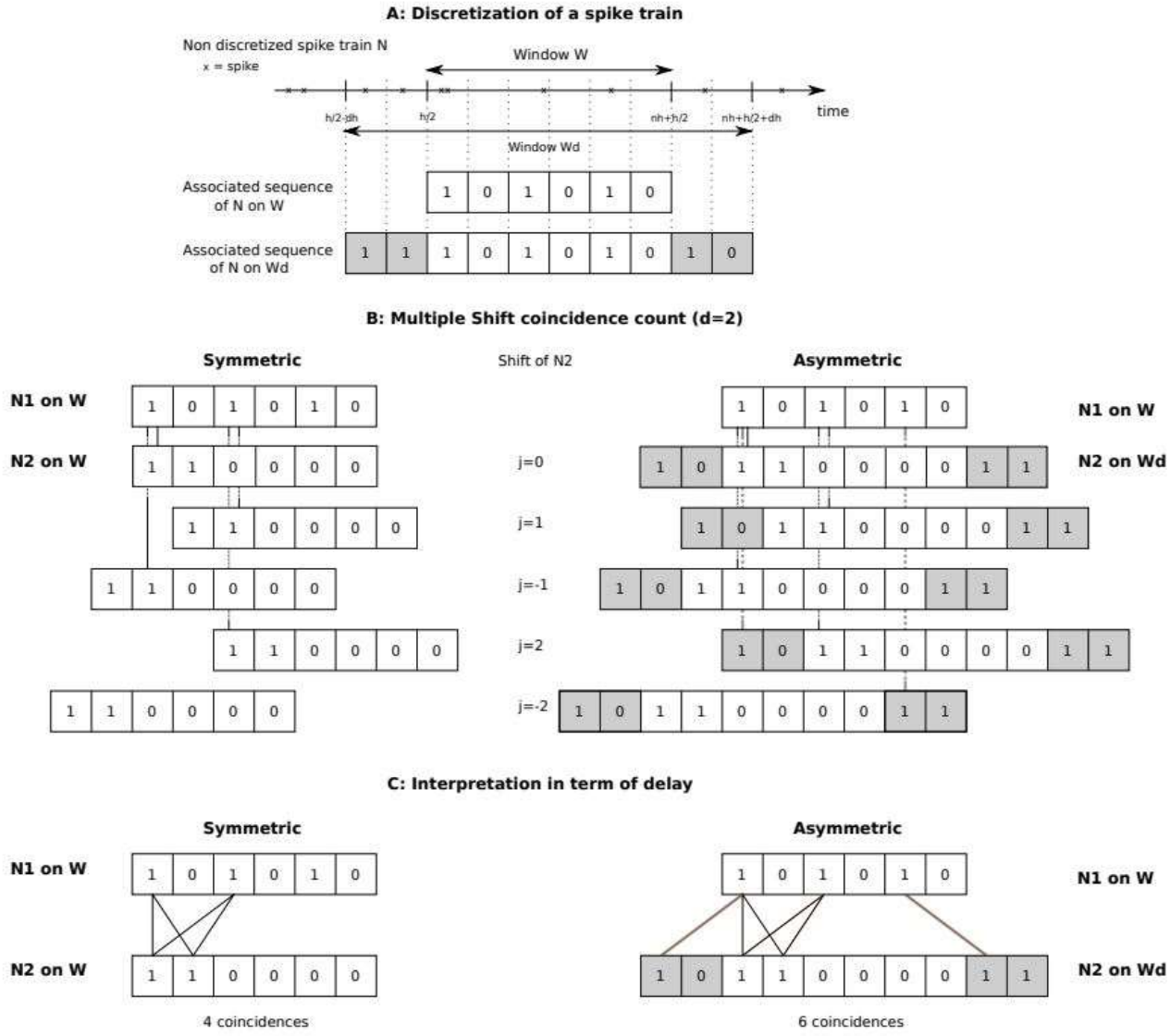
Spike Events

Another attribute of this UE method was to relate the occurrence of spike synchrony behavior. This test used a sliding window analysis to calculate expected number of coincidences (C) and make evaluations if they were significant.



Coincidence Events

One major drawback to the UE method, that is later proved by the MTGAUE, is that assumptions of the individual neuron spike and coincidence of counts are following a Poisson distribution as well as the use of the binned coincidence count. Up to 60% of coincidences can be lost when the bin length is the typical delay between two spikes participating to the same coincidence. MTGAUE aims to complete the study of this notion of multiple shift coincidence count by proposing a new method that extends the validity of the original UE. This article focuses on the symmetric multiple shift coincidence count, which is more adapted to purpose of testing independence between two spike trains in each window. The issue is that neurons are not homogeneous in time. There are variations in spikes throughout a given time period. The main point is that there needs to be distinction between symmetric and asymmetric multiple shift coincidence count.



Coincidence Events

The null hypotheses in this test is that the spike trains of two neurons, N_1 and N_2, are independent (H_0) and the alternative is that both of them are dependent (H_1). 2 assumptions are put in place, the first is that N1 and N2 are poisson variables and the second is that they are stationary on the given window. Thus, expectation and variance can be given by.

$$\mathbb{E}(X) = \lambda_1\lambda_2\left[2\delta T - \delta^2\right]$$
$$Var(X) = \lambda_1\lambda_2\left[2\delta T - \delta^2\right] + \left[\lambda_1^2\lambda_2 + \lambda_1\lambda_2^2\right]\left[4\delta^2 T - \frac{10}{3}\delta^3\right].$$

Expectation and Variance

The coincidence must also be taken into consideration for the binning framework to function, this is where the UE method takes the Poisson distribution, however the article suggests to use the delayed coincidence count with multiple shift methods. In definition, the delayed coincidence count is denoted by

$$X = \int_{W^2} 1_{|x-y|\leq\delta} N_1(dx)N_2(dy)$$

where δ is defined as the delay on the window W. N_1 and N_2 are discretized with resolution h. The new method focuses on the symmetric delayed coincidence count X, this is because the symmetric coincidence count is much more adapted to the purpose of testing hypothesis between N1 and N2 on a fixed wind W. Following this the

authors prove the Gaussian approximation of the UE methods (GAUE) where the symmetric test, unilateral test by upper value, and unilateral test by lower value can be formalized similarly to the original UE multiple shift method.

- the symmetric test $\Delta_{GAUE}^{sym}(\alpha)$ of H_o, which rejects if H_0 if $\bar{m}$ and $\hat{m_0}$ are too different
- the unitaleral test by upper value $\Delta_{GAUE}^{+}(\alpha)$ which rejects H_0 if $\bar{m}$ is too large
- the unilateral test by lower vallue $\Delta_{GAUE}^{-}(\alpha)$ which rejects H_0 if $\bar{m}$ is too small

For both methods, the symmetric test encompasses both the upper and lower cases at a smaller level. Moreover, the GAUE tests are aimed to detect both profusion and lack of coincidences. In addition, the authors mention preference of the Benjamini-Hochberg approach to control the false discovery rate (FDR), over the familywise error rate.

$$\mathrm{FDR} = \mathbb{E}\left(\frac{F_d}{R}\mathbf{1}_{R>0}\right).$$

$$P_{W(1)}^{(1)} \leq \ldots \leq P_{W(\ell)}^{(\ell)} \leq \ldots \leq P_{W(K)}^{(K)}.$$

Expectation and Variance

If the p-values are uniformly and independently distributed under the null hypothesis, then the procedure guarantees an FDR less than q. When combining the GAUE tests, with the assumptions required for Benjamini-Hochberg is not satisfied due to gaps between what is theoretical and practiced, however the difference does not significantly impact the FDR.

Conclusion :

MTGAUE is effective using symmetric tests when the p-value is assigned to different windows. The article concludes that involving the Gaussian approximation and Benjamani-Horchberg multiple test procedure can improve a more precise statistical method without an edge effect. The article proves the theory by running thousands of simulated tests on overlapping windows to compare MTGAUE and UE's detection performance on a wide range of parameters. In fact, they prove that the MTGAUE is a robust improvement to the prior UE method due to its handling of large datasets with a far stable FDR levels set at 0.05.