Logo quality predicting companies' revenues

Abstract

This research would like to study how companies' logo could have an impact on their revenues or profit.Logo are the face and represent more than just the name of the company, because through colors, shape, fonts and other design characteristics can transmit the company's values and qualities.In today's market, they logo are used for mass communication and the company's logo are synomis of the trademark and the company's brand. Therefore, we used the data from Fortune 1000 companies and we added their information of their logo to better understand how their revenues or profit related to their logo. There are a lot of design patterns used by the companies, so it was easy for us to categorize and we created new variable for the data set to add the logo's information. Each variable will be measuring something different, the variables are: `Networth` (Categorical), `Revenue` (in Millions), `Profit` (in Millions), `Ranking` (1-1000). Networth is a variable that will not be in used, however Revenue and Profit will both be measure in million of dollars. The ranking also come from the Fortune 1000, depending what their ranking is in that list. In this paper we will build a logistic model to predict if the logo of the companies can effectively predict the company's revenue.Then we would like to create more data, from the data-set that we already build, using some Machine Learning techniques for example like a decision tree to then try to predict the revenue or profit of a company given certain charismatics of a logo.

The dataset

The data-set that is used in this project comes from Fortune 1000. Our data-set has about 60 logo's companies however, the companies were picked randomly by their ranked in the Fortune 1000. In additions to having the basic information of the logo as variable, we added more variable to help us distinguished the logo even more. The additional variable were: `Font` which is the information on the type of Font character that they used for the logo, `Color - hex #` which are the colors presented in the logo as hexadecimal values, `Type of Logo` which are Combination logos (0), Wordmarks(1), Lettermarks(2), Emblems(3), Abstract icons (4) and Pictorial icons(5), `Living thing in the logo` if the logo was representing an animal or some living thing, `Gradient (y,n)` which represent if it has gradients colors in the logo. In figure 1, there is a sample of the data-set that we are going to be using to answer this question.

type of logo

Figure 1

	C1-T	C2-T	C3	C4	C5	C6-T	C7-T	C8	C9	C10	C11
	Logo Na...	Networth	Revenue	Profit	Ranking	Font	Color - h...	Number...	Type of L...	Living thi...	Gradient...
1	Nike	$ 26 billi...	34350	4240.0	89	Swoosh	#0D0D0...	1	4	0	0
2	Apple	$1 trillion	229234	48351.0	4	Bitten Ap...	#AAAAAA	1	5	0	0
3	Walmart	$514.405...	500343	9862.0	1	*Missing*	#F2B74...	2	0	0	0
4	Amazon	$1 trillion...	177866	3033.0	8	sans seri...	#F3993E...	3	0	0	0
5	Mircosoft	$1 trillion	89950	21204.0	30	*Missing*	#46A5F...	5	0	0	0
6	Target	$62.6 bill...	71879	2934.0	39	Helvetica...	#CD2F2...	2	4	0	0
7	Disney	$130 billi...	55137	8980.0	55	Waltogra...	#202D7...	3	0	0	0
8	Facebook	*Missing*	40653	40653.0	76	Klavika	#253659...	3	2	0	0
9	Coca-Cola	$230 billi...	35410	1248.0	87	Spenceri...	#BF2A2A...	2	1	0	0
10	3M	$32.8 bill...	31657	4858.0	97	Helvetica...	#E3353...	2	1	0	0
11	Pepsi Co.	$18.8 billi...	63525	4857.0	45	Sans Serif	#0B0B0...	5	4	0	0

Data Cleaning

The testing of our data is going to be done in R Studio and also in JUMP. However the cleaning of the data was being done by using R. We want to create a cleaner data set therefore it would be much easier to use in R Studio. By cleaning the data-set it made it much easier to find the outliers in the data. By running the following code we were able to plot the data point and see which data were the outliers. Using the Z-score that were greater than `3.29` we found that there were 57 outliers. But when we actually plot it we can see that there is actually less outliers as it shown in figure 2. This could be possible because our data set does have some companies that have a larger profit than the others.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)

logorev <- read_csv("logorev.csv")
```

```
## Parsed with column specification:
## cols(
##   `Logo Names` = col_character(),
##   Networth = col_character(),
##   Revenue = col_double(),
##   Profit = col_double(),
##   Ranking = col_double(),
##   Font = col_character(),
##   `Color - hex #` = col_character(),
##   `Number of Colors` = col_double(),
##   `Type of Logo` = col_double(),
##   `Living thing in the logo` = col_double(),
##   `Gradient (y,n)` = col_double()
## )
```

```
View(logorev)
attach(logorev)
```

Renamining

We would like to create a cleaner data set, and make the name of the variable easy to read and without space so it will be easire to call them in `R`.

```
logorev_cleaned <- logorev
names(logorev)
```

```
##  [1] "Logo Names"               "Networth"
##  [3] "Revenue"                  "Profit"
##  [5] "Ranking"                  "Font"
##  [7] "Color - hex #"            "Number of Colors"
##  [9] "Type of Logo"             "Living thing in the logo"
## [11] "Gradient (y,n)"
```

```
logorev_cleaned <- logorev_cleaned %>%
  rename(`Name` = `Logo Names`, `Colors` = `Color - hex #`, `Number_of_colors` = `Number of Colo
rs`, `Type_of_logo` = `Type of Logo`, `Living_in_logo` = `Living thing in the logo`, `Gradient`
= `Gradient (y,n)`)
```

Variables Type

Because the variable `Gradient` and `Living in logo` are dichotomus variables we need to transform them as factors.

```
### Making variables from numeric to factor
cols <- c('Gradient', 'Living_in_logo')
logorev_cleaned[cols] <- lapply(logorev_cleaned[cols], as.factor)
```

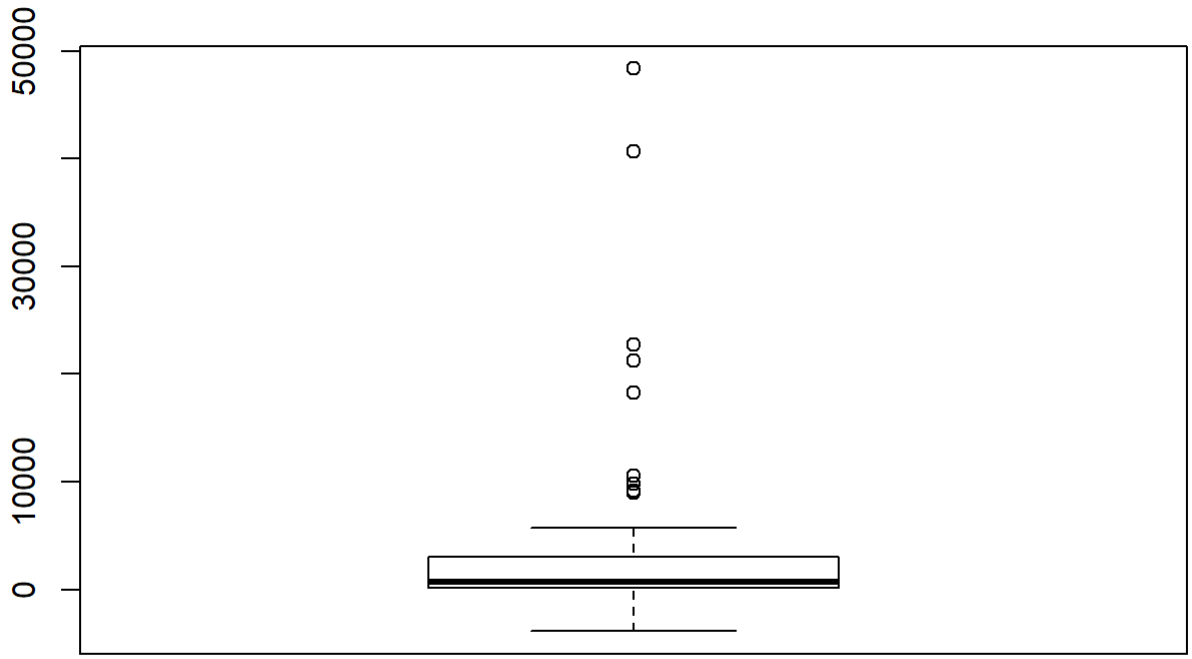
Outliers

From the z scorse greate than `3.29` we can see that there are 57 outliers, but in the boxplot we would see that there actually less outliers. We although have some big companies that have larger profit than others.The variabilieties within cases is very large and only few companies have higher profit than other. We will need to consider this characteristic of our dataset later in our Analysis.

```
attach(logorev_cleaned)
```

```
## The following objects are masked from logorev:
##
##     Font, Networth, Profit, Ranking, Revenue
```

```
boxplot(Profit)
```

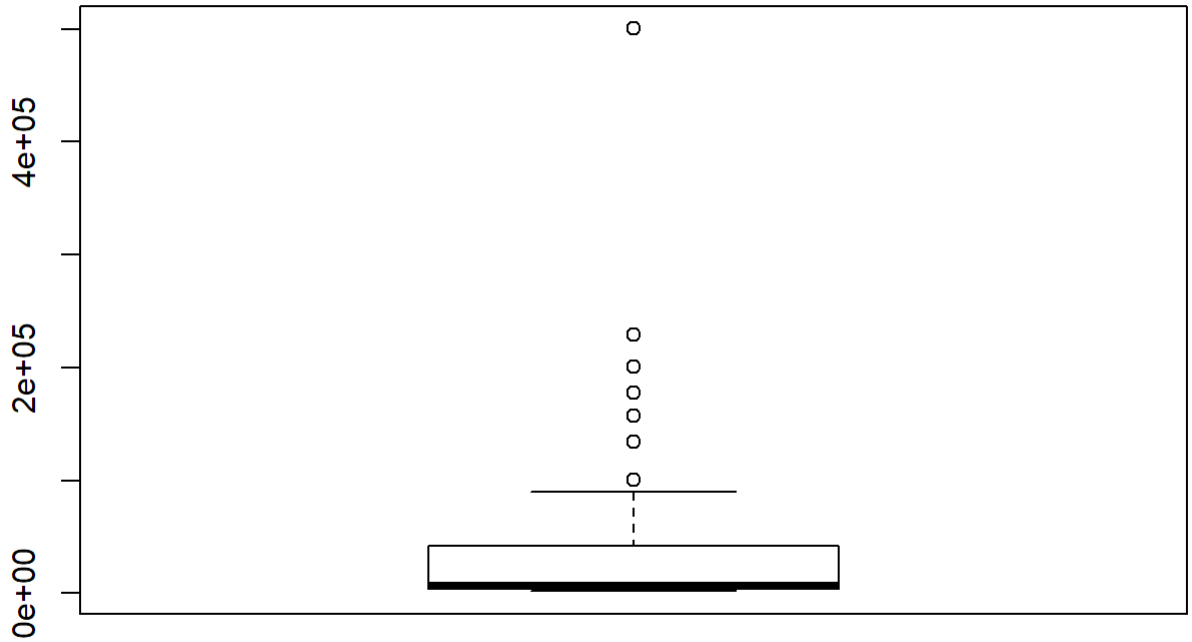



```
sum(abs(Profit - mean(Profit)/ sd(Profit)) > 3.29)
```

```
## [1] 57
```

From the boxplot we can clearly see that there is a company that have a revenue greater than 400000 which is Walmart. So we might need to take in consideration this in our Analysis that there are some Companies' revenue that might influence the results.

```
boxplot(Revenue)
```

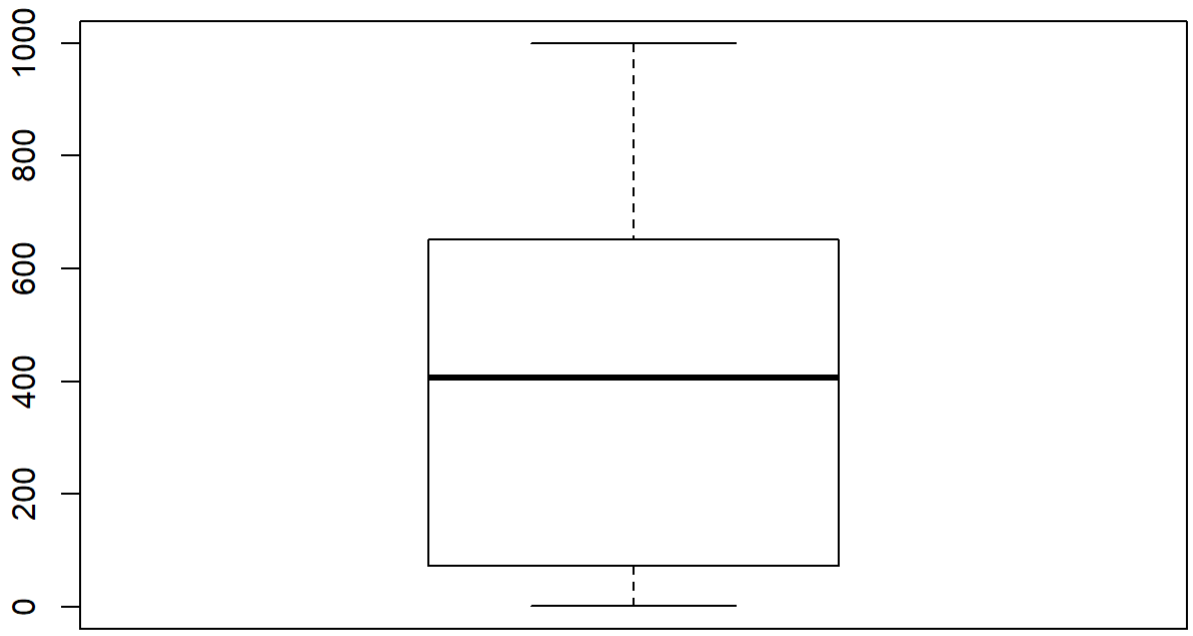



```
sum(abs(Revenue - mean(Revenue)/ sd(Revenue)) > 3.29)
```

```
## [1] 57
```

For the variable `Ranking` we have almost all of the cases layed as outliers for the rule of Z-scores greater than 3.29. The boxplot shows that there are not any outliers. This discrepancy is related to the fact that we have large variances between the datapoints.

```
boxplot(Ranking)
```

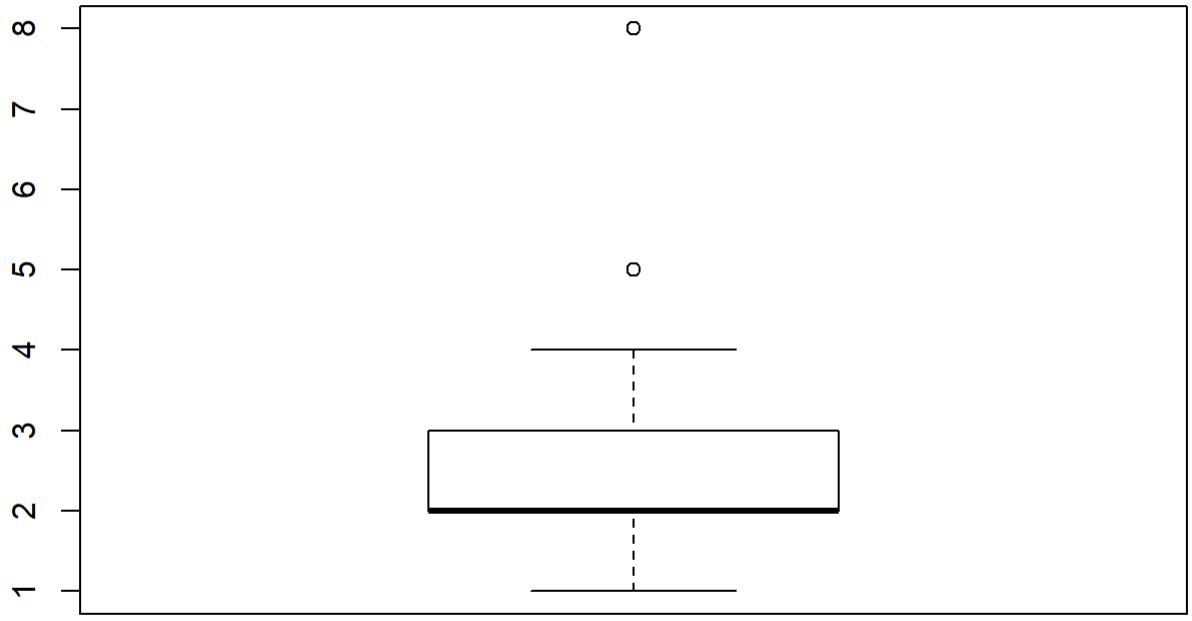



```
sum(abs(Ranking - mean(Ranking)/ sd(Ranking)) > 3.29)
```

```
## [1] 55
```

The variable `Number of colors` have also few outliers. If we localized them using the boxplot we can see that only two companies have a logo with 5 and 8 colors. If we use the z-scores we can calculate that there are 7 cases that can be possibly be outliers.

```
boxplot(Number_of_colors)
```




```
sum(abs(Number_of_colors - mean(Number_of_colors)/ sd(Number_of_colors)) > 3.29)
```

```
## [1] 7
```

Missing Values

We have not contiunued to input in our dataset the `networth` variable and the `font` variable, so those variables will have missing values.

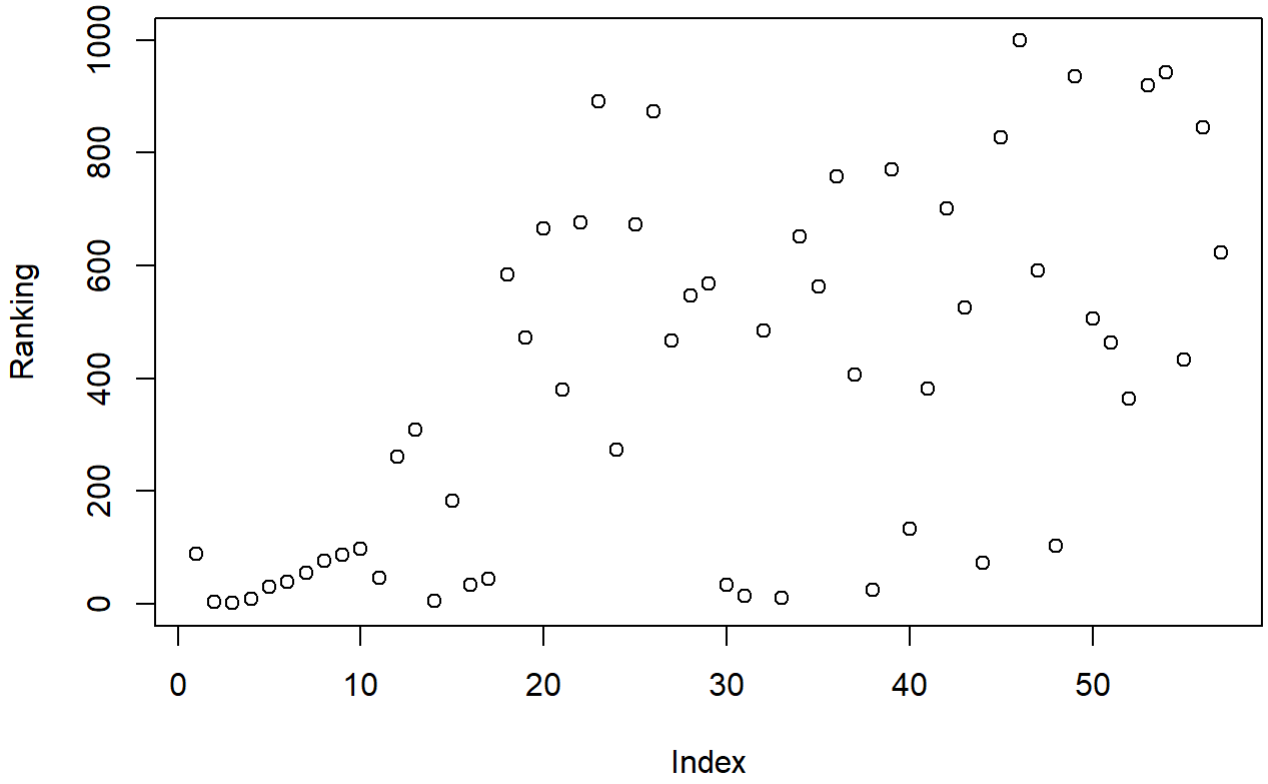
Assumptions

For deciding what would be the best way to predict profits or revenue of a company based on their Logo characteristic we would need to check the conditions of Linearity, Independence, Normality, Equality of Variance and if we have multicollinear variables. This way we can better understand our variable and how to use them in our Analysis.

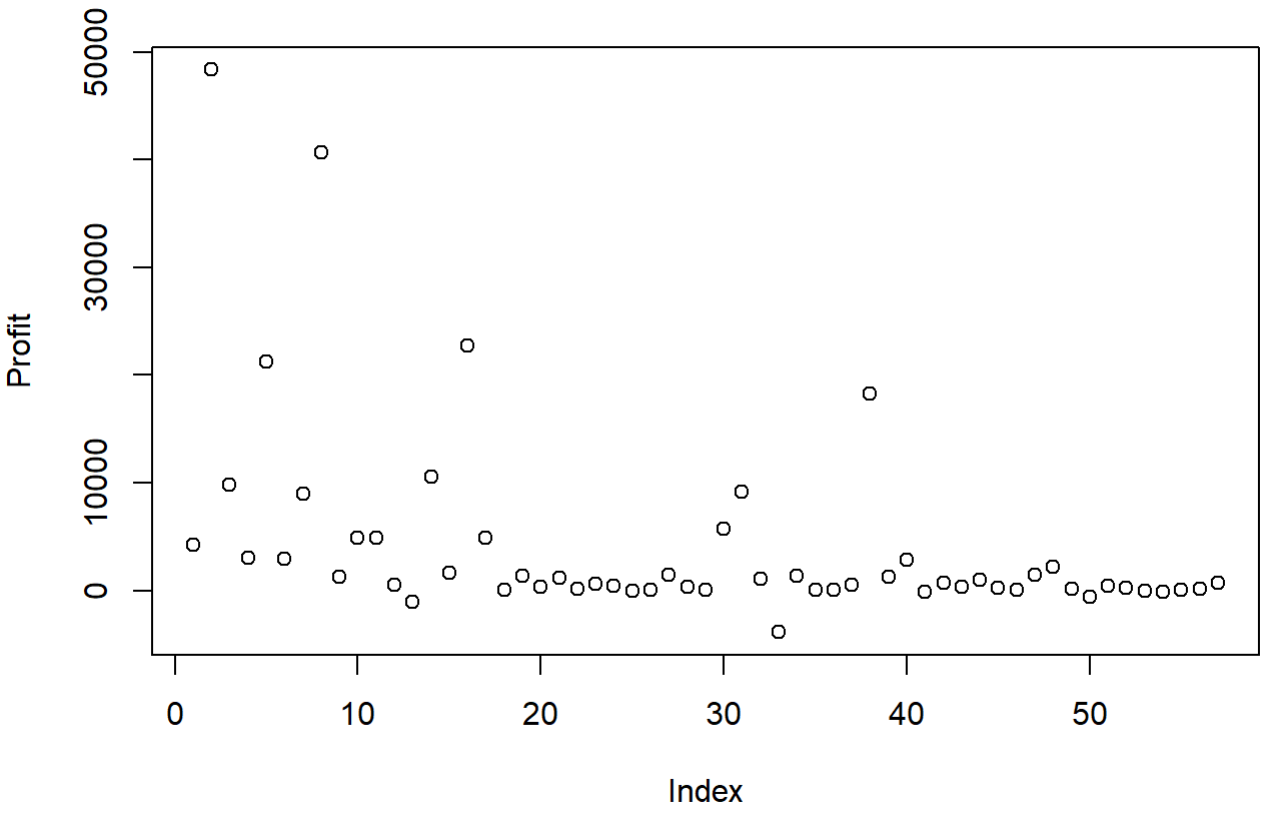
Linearity:

As expected the scotterplot for the variable `Ranking` shows a positive path and it look like it is well randomly spread. The `Profit` variable shows in the scatterplot that there are some outlies because of high leverage because fare away from the companies that have a lower profit. It could be helpful to consider to transform the variable. `Revenue` 's scatterplot also indicates that there are significant outliers and it looks like a logarithmic transformation is needed because it shows a parabolic shape in the distribution of the data points.

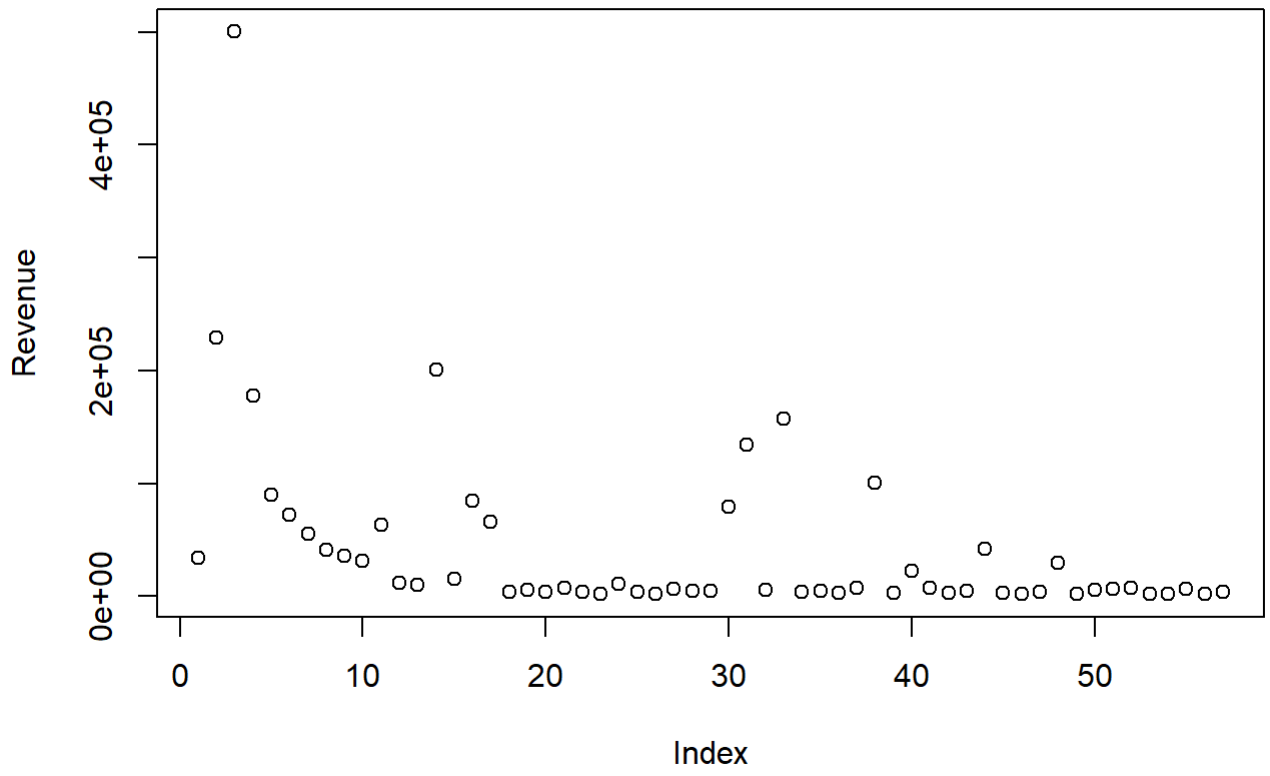
```
plot(Ranking)
```



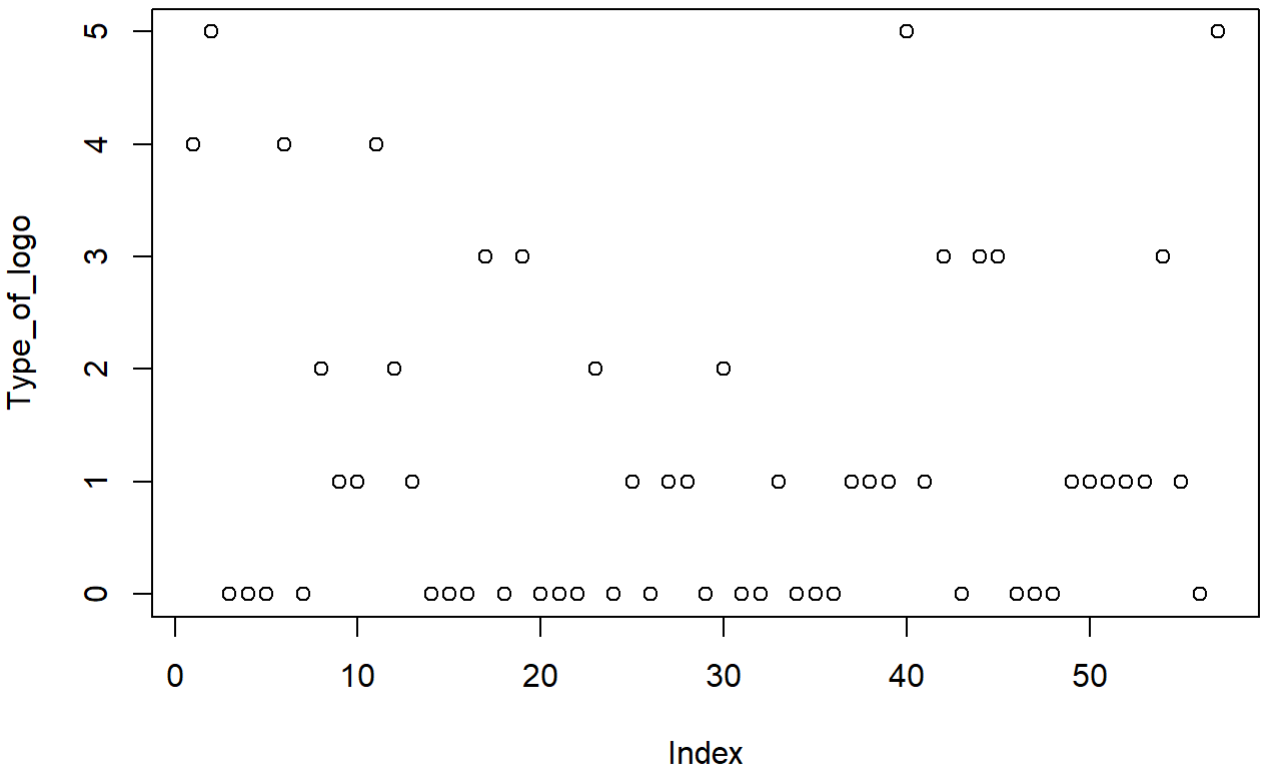

```
plot(Profit)
```



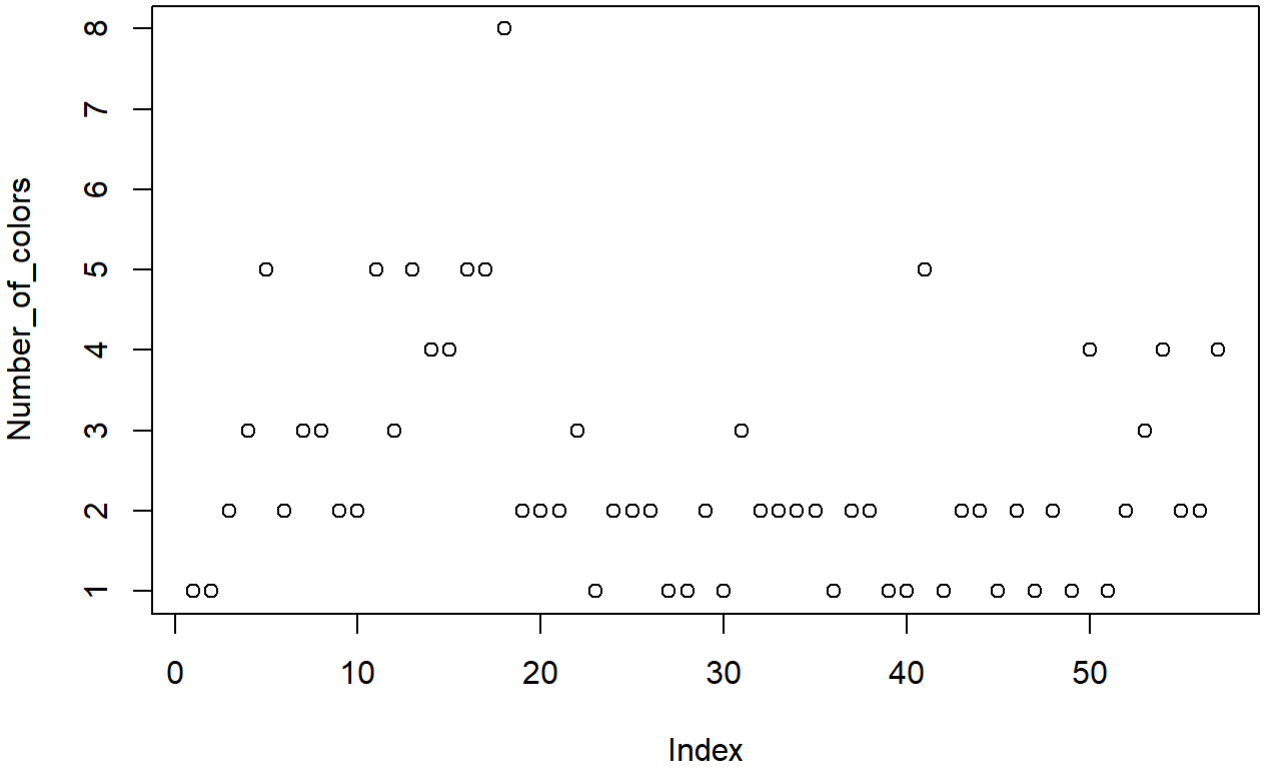

```
plot(Revenue)
```




```
plot(Type_of_logo)
```

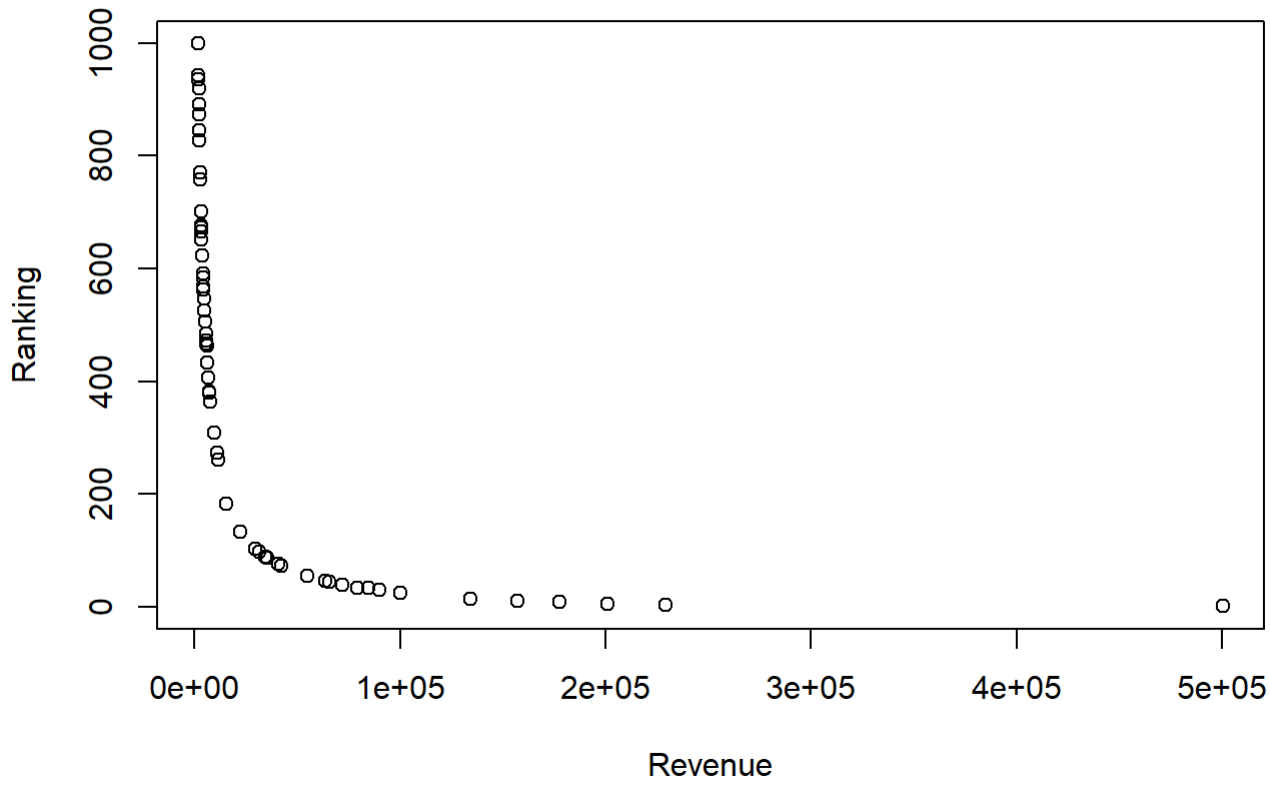



```
plot(Number_of_colors)
```

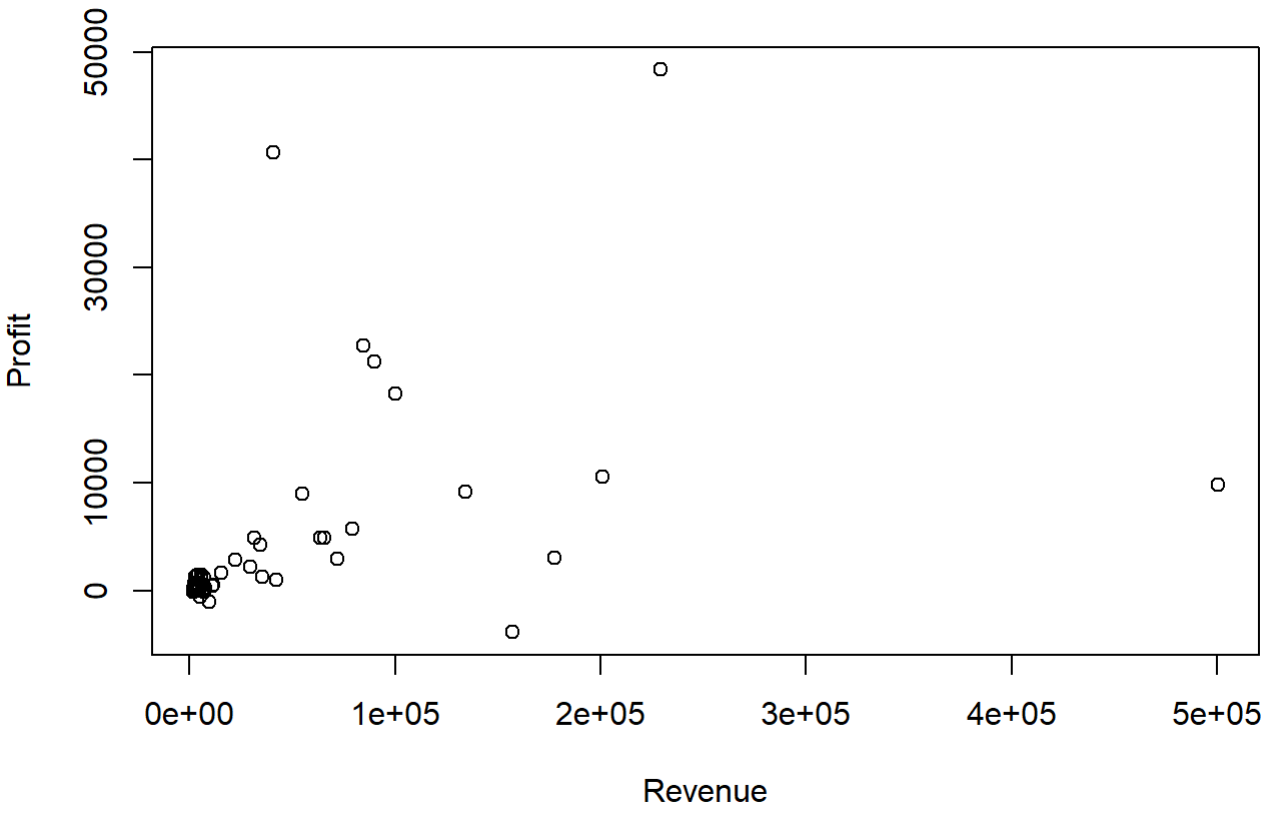


If we try to plot more variables togheter to see which variables can have the best path in a scatterplot we can see that `Revenue` and `Ranking` have a parabolic shape and therefore it might need some transformation. The scatterplot of the relation between `Revenue` and `Profit` is not very defined since there are some outliers, but overall it looks to have a weak and positive relationship. The relationship between `Number of colors` and `Revenue` is very interesting since we do not have any linear pattern, but it looks like that the variablity is larger when there are less colors. We cannot state this without analyze the data more deeply. Overall we do not see any linear pattern.

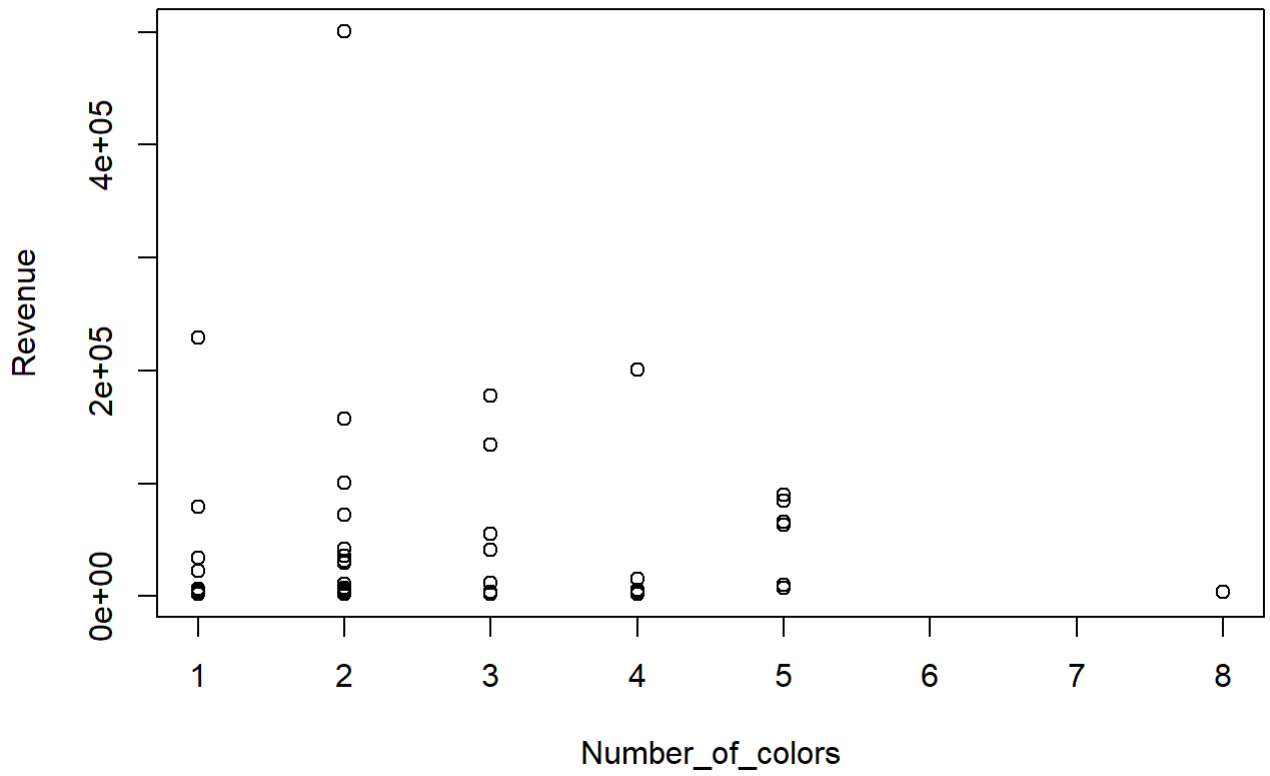
```
plot(Revenue, Ranking)
```




```
plot(Revenue, Profit)
```




```
plot(Number_of_colors, Revenue)
```



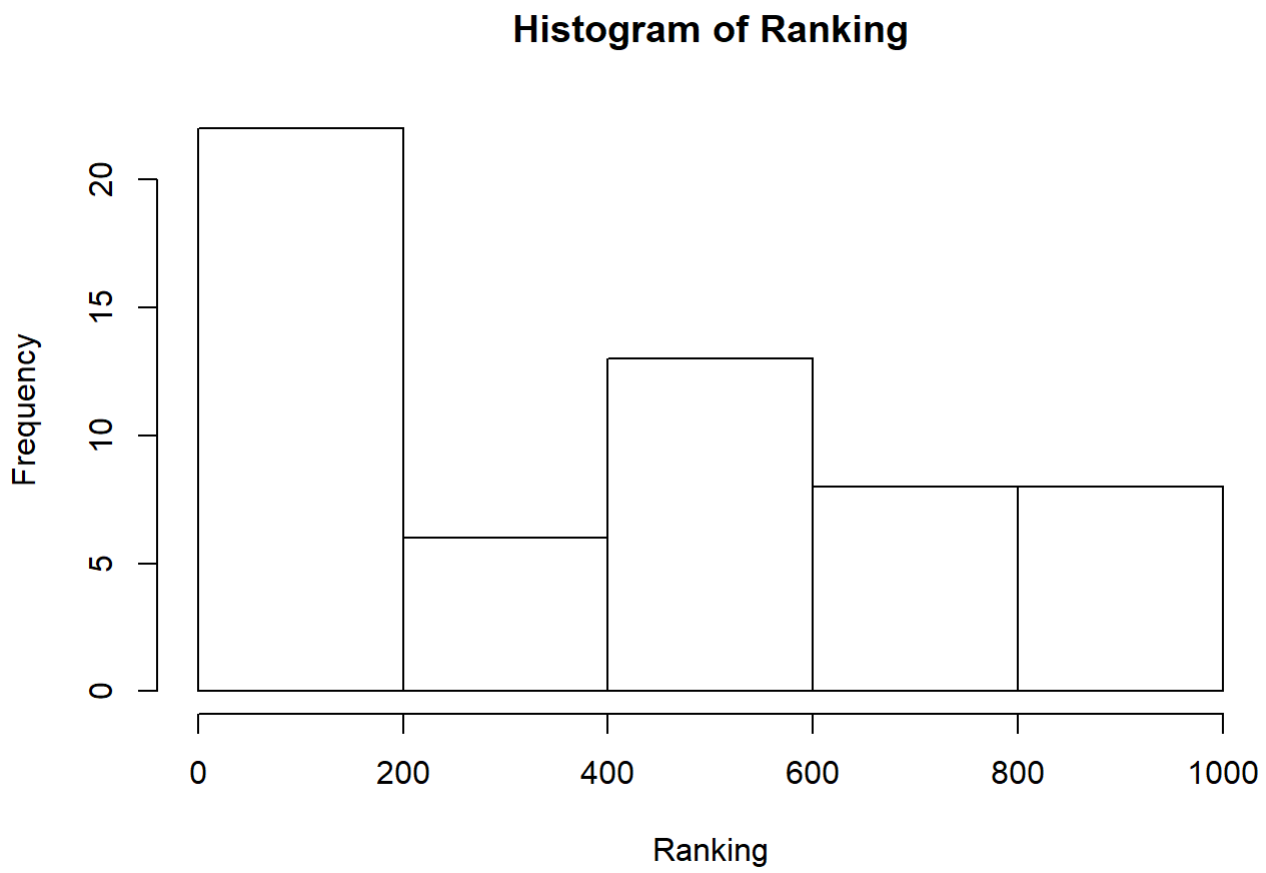
Independence:

We assume that our data come from a indpendent sample where given one variable we do not know anything about another variable. This is true, infact, for all our variables.

Normality:

From the histogram of the distribution of our variables of `Ranking`, `Profit`, `Revenue`, `Number Of Colors` and `Type of logo` we can state that all of these variables are not normally distributed because they are skewed right.

```
library(moments)

hist(Ranking)
```



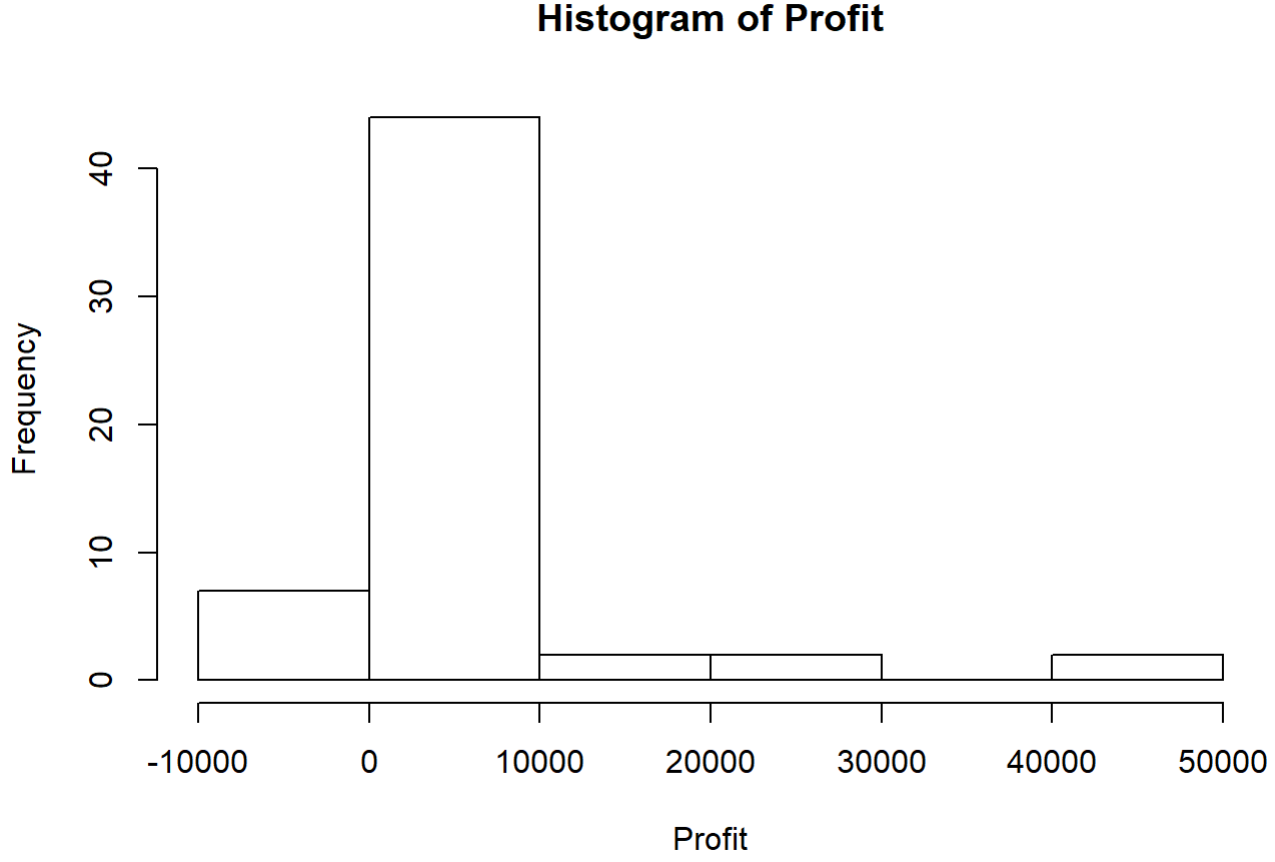

```
skewness(Ranking)
```

```
## [1] 0.2358989
```

```
kurtosis(Ranking)
```

```
## [1] 1.733534
```

```
hist(Profit)
```



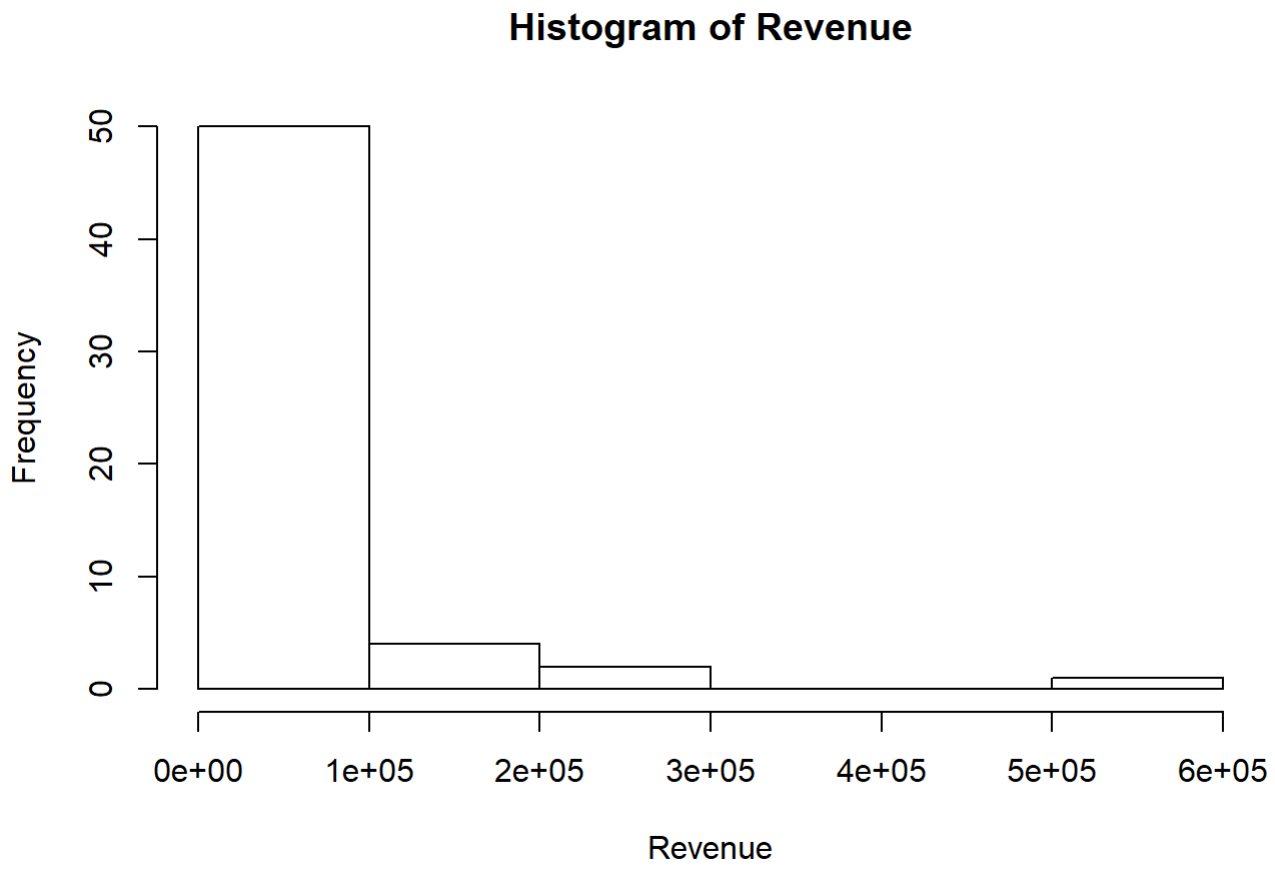

```
skewness(Profit)
```

```
## [1] 3.277405
```

```
kurtosis(Profit)
```

```
## [1] 14.11682
```

```
hist(Revenue)
```



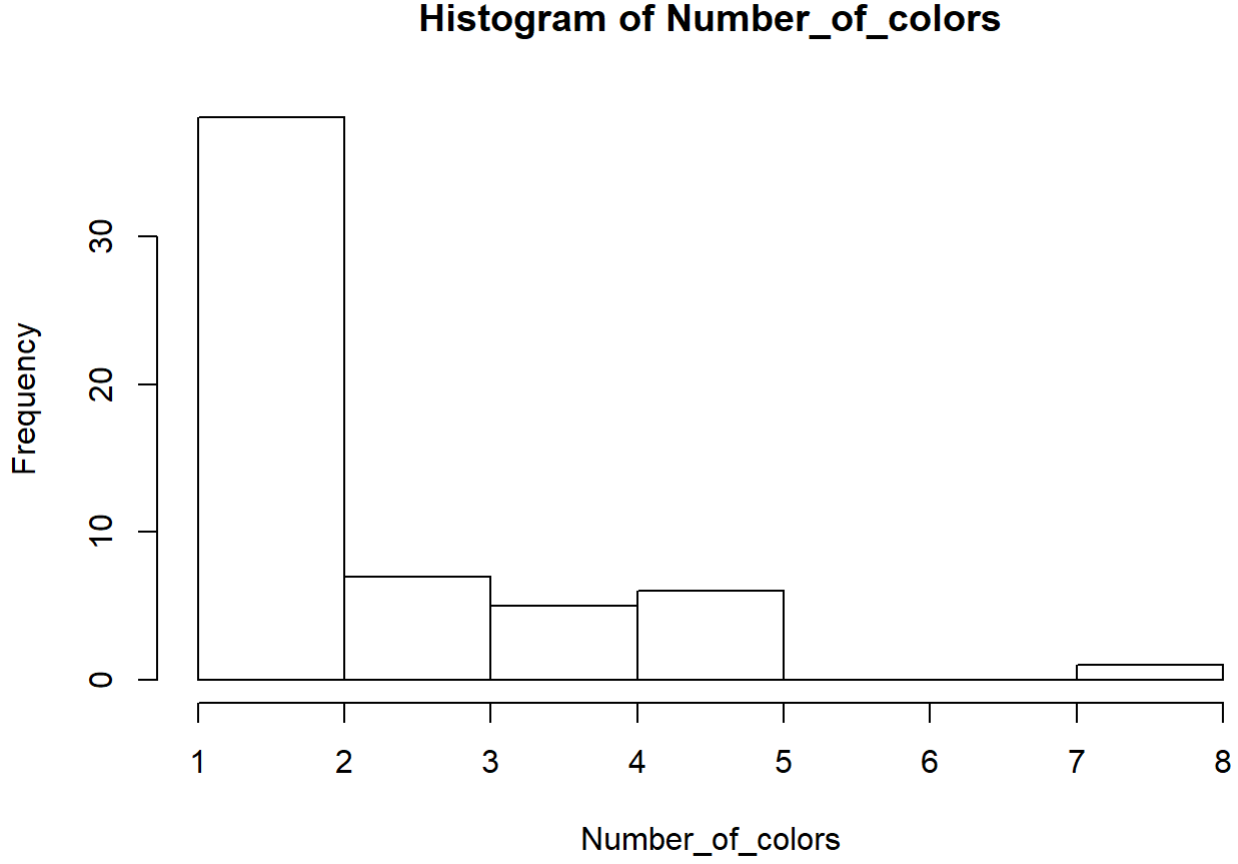

```
skewness(Revenue)
```

```
## [1] 3.650869
```

```
kurtosis(Revenue)
```

```
## [1] 19.13349
```

```
hist(Number_of_colors)
```



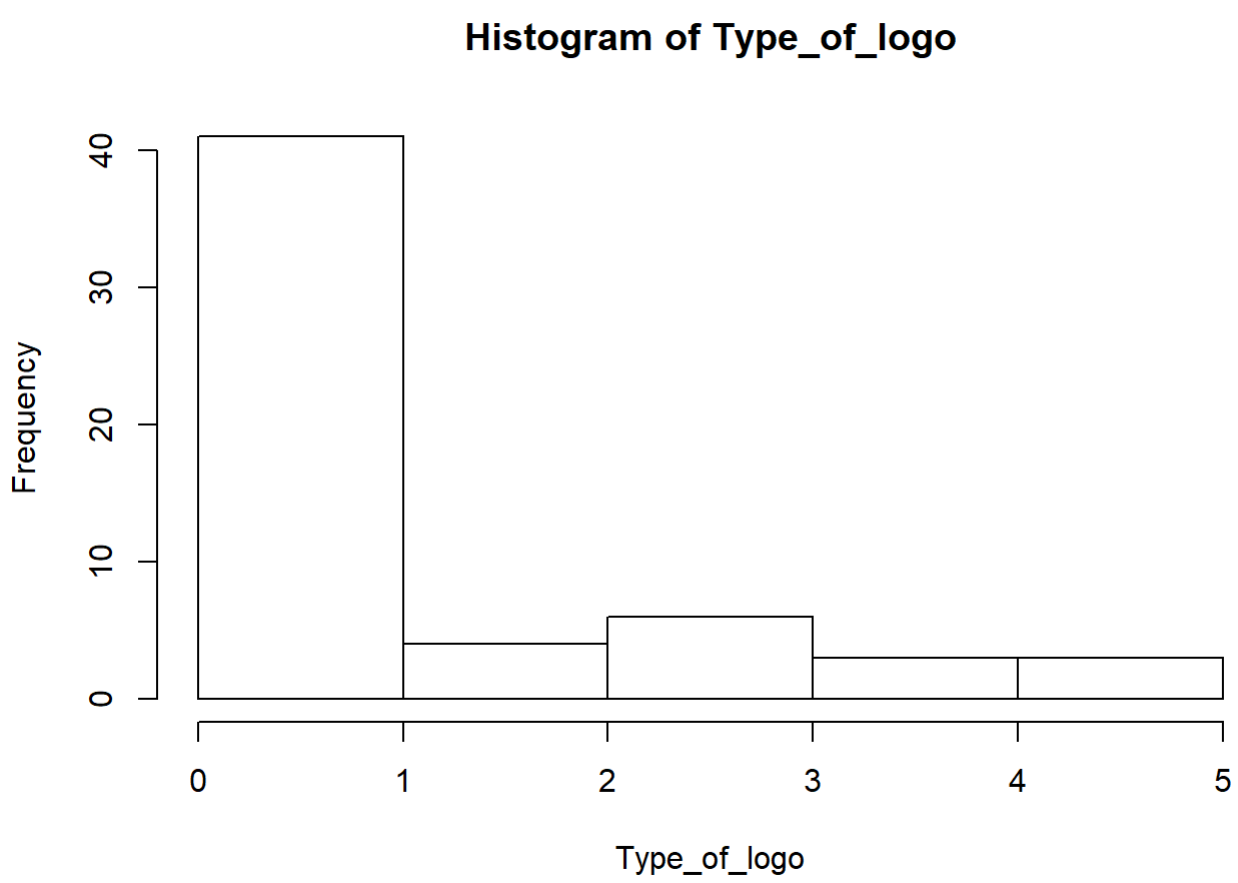

```
skewness(Number_of_colors)
```

```
## [1] 1.391956
```

```
kurtosis(Number_of_colors)
```

```
## [1] 5.188652
```

```
hist(Type_of_logo)
```




```
skewness(Type_of_logo)
```

```
## [1] 1.177559
```

```
kurtosis(Type_of_logo)
```

```
## [1] 3.332914
```

Equality Of Variance:

For assessing if there is **homoscedasticity** we would need to refer to the scatterplot that we used to assess Linearity. None of the scotterplot shows equal variances between the cases. So we can conclude that none of the independent and dependet variables have equal variance.

Multicollinearity & Singularity:

By looking at the covariance matrix we do not have any correlations abot 0.7, which means that we do not have any Multicollinearity. We do not have any correlations between variables that are greater than 1, so we do not have Singularity.

```
round(cor(cbind(Ranking, Profit, Revenue, Number_of_colors, Type_of_logo, Living_in_logo, Gradient)),3)
```

```
##                 Ranking Profit Revenue Number_of_colors Type_of_logo
## Ranking           1.000 -0.471  -0.565           -0.206       -0.153
## Profit           -0.471  1.000   0.443            0.085        0.203
## Revenue          -0.565  0.443   1.000            0.048        0.000
## Number_of_colors -0.206  0.085   0.048            1.000       -0.100
## Type_of_logo     -0.153  0.203   0.000           -0.100        1.000
## Living_in_logo   -0.078 -0.080  -0.091           -0.043        0.283
## Gradient          0.000 -0.063  -0.041            0.199       -0.132
##                 Living_in_logo Gradient
## Ranking                 -0.078    0.000
## Profit                  -0.080   -0.063
## Revenue                 -0.091   -0.041
## Number_of_colors        -0.043    0.199
## Type_of_logo             0.283   -0.132
## Living_in_logo           1.000   -0.085
## Gradient                -0.085    1.000
```

Analysis

```
head(logorev_cleaned)
```

```
## # A tibble: 6 x 11
##   Name  Networth Revenue Profit Ranking Font  Colors Number_of_colors
##   <chr> <chr>      <dbl>  <dbl>   <dbl> <chr> <chr>             <dbl>
## 1 Nike  "$ 26 b~   34350   4240      89 Swoo~ #0D0D~                1
## 2 Apple "$1 tri~  229234  48351       4 Bitt~ #AAAA~                1
## 3 Walm~ "$514.4~  500343   9862       1 <NA>  #F2B7~                2
## 4 Amaz~ "$1 tri~  177866   3033       8 sans~ #F399~                3
## 5 Mirc~ "$1 tri~   89950  21204      30 <NA>  #46A5~                5
## 6 Targ~ "$62.6 ~   71879   2934      39 Helv~ #CD2F~                2
## # ... with 3 more variables: Type_of_logo <dbl>, Living_in_logo <fct>,
## #   Gradient <fct>
```

```
summary(logorev_cleaned)
```

```
##      Name              Networth            Revenue           Profit
##  Length:57          Length:57          Min.   :  1849   Min.   :-3864.0
##  Class :character   Class :character   1st Qu.:  3574   1st Qu.:  107.3
##  Mode  :character   Mode  :character   Median :  6939   Median :  743.4
##                                        Mean   : 42656   Mean   : 4190.7
##                                        3rd Qu.: 42151   3rd Qu.: 3033.0
##                                        Max.   :500343   Max.   :48351.0
##     Ranking        Font              Colors          Number_of_colors
##  Min.   :  1   Length:57          Length:57          Min.   :1.000
##  1st Qu.: 72   Class :character   Class :character   1st Qu.:2.000
##  Median :407   Mode  :character   Mode  :character   Median :2.000
##  Mean   :395                                         Mean   :2.474
##  3rd Qu.:652                                         3rd Qu.:3.000
##  Max.   :999                                         Max.   :8.000
##   Type_of_logo   Living_in_logo Gradient
##  Min.   :0.000   0:53           0:52
##  1st Qu.:0.000   1: 4           1: 5
##  Median :1.000
##  Mean   :1.228
##  3rd Qu.:2.000
##  Max.   :5.000
```

Decision Tree

A quick and easy decision Tree using `R`. We will use the packages `MASS` and `rpart`.

```
# Loading the libraries
library(MASS)
```

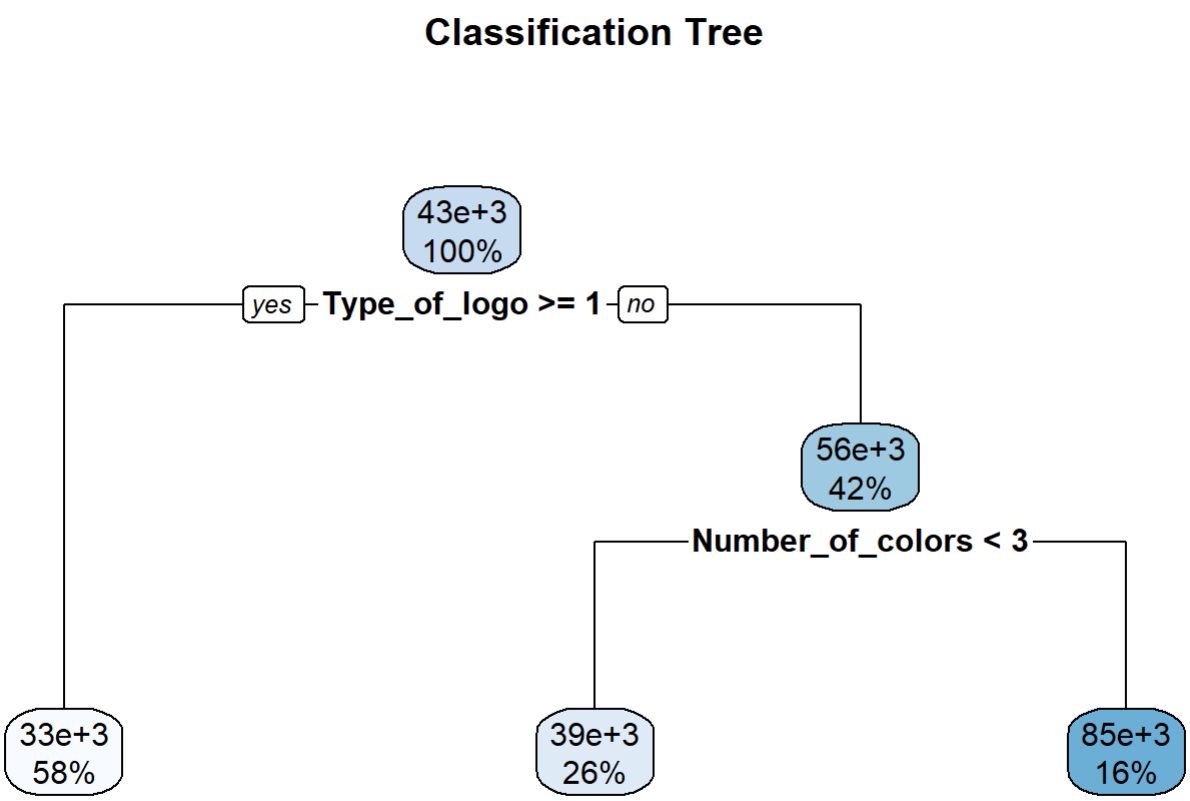
```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(rpart)
```

The Decsion Tree successfully split the top node based on the type of logo and it defined high revenues companies the one that have a type of logo equal to 0, which is the combination logo (A/N such as Walmart's logo). Then the split was based on the number of colors of the logo. IF the logo has more than 3 colors, then the company would be considered as a company with high revenues.

```
### Making the
library(rpart.plot)
set.seed(1)
train <- sample(1:nrow(logorev_cleaned), 0.75 * nrow(logorev_cleaned))

logoTree <- rpart(Revenue ~ Type_of_logo + Number_of_colors + Living_in_logo + Gradient, data =
 logorev_cleaned)

rpart.plot(logoTree, main="Classification Tree")
```



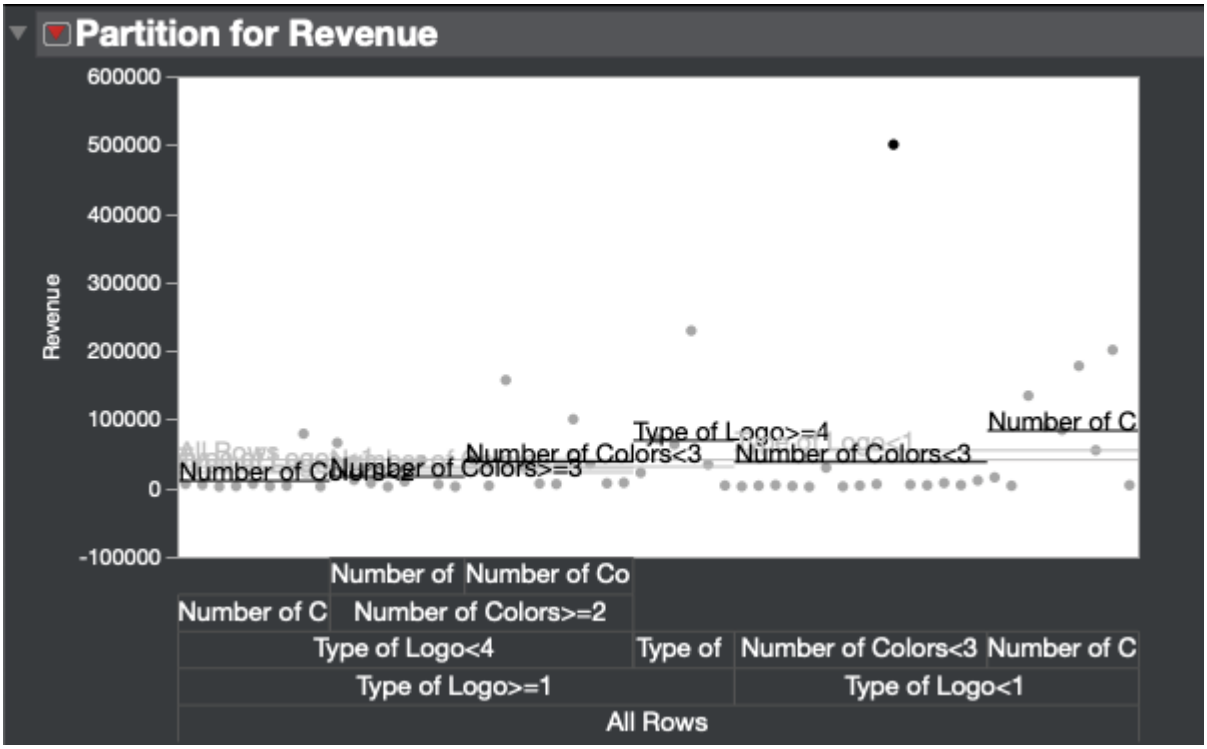

```
summary(logoTree)
```

```
## Call:
## rpart(formula = Revenue ~ Type_of_logo + Number_of_colors + Living_in_logo +
##     Gradient, data = logorev_cleaned)
##   n= 57
##
##           CP nsplit rel error   xerror      xstd
## 1 0.02657677      0 1.0000000 1.062557 0.5904755
## 2 0.01000000      2 0.9468465 1.173482 0.6632139
##
## Variable importance
## Number_of_colors     Type_of_logo         Gradient
##               56               36                8
##
## Node number 1: 57 observations,    complexity param=0.02657677
##   mean=42655.53, MSE=6.522562e+09
##   left son=2 (33 obs) right son=3 (24 obs)
##   Primary splits:
##       Type_of_logo     < 0.5 to the right, improve=0.021274340, (0 missing)
##       Number_of_colors < 1.5 to the left,  improve=0.009717432, (0 missing)
##   Surrogate splits:
##       Number_of_colors < 1.5 to the left,  agree=0.596, adj=0.042, (0 split)
##       Gradient         splits as  LR,      agree=0.596, adj=0.042, (0 split)
##
## Node number 2: 33 observations
##   mean=32609.7, MSE=2.42692e+09
##
## Node number 3: 24 observations,    complexity param=0.02657677
##   mean=56468.54, MSE=1.182451e+10
##   left son=6 (15 obs) right son=7 (9 obs)
##   Primary splits:
##       Number_of_colors < 2.5 to the left,  improve=0.04176439, (0 missing)
##   Surrogate splits:
##       Gradient splits as  LR, agree=0.667, adj=0.111, (0 split)
##
## Node number 6: 15 observations
##   mean=39255, MSE=1.52302e+10
##
## Node number 7: 9 observations
##   mean=85157.78, MSE=4.831433e+09
```

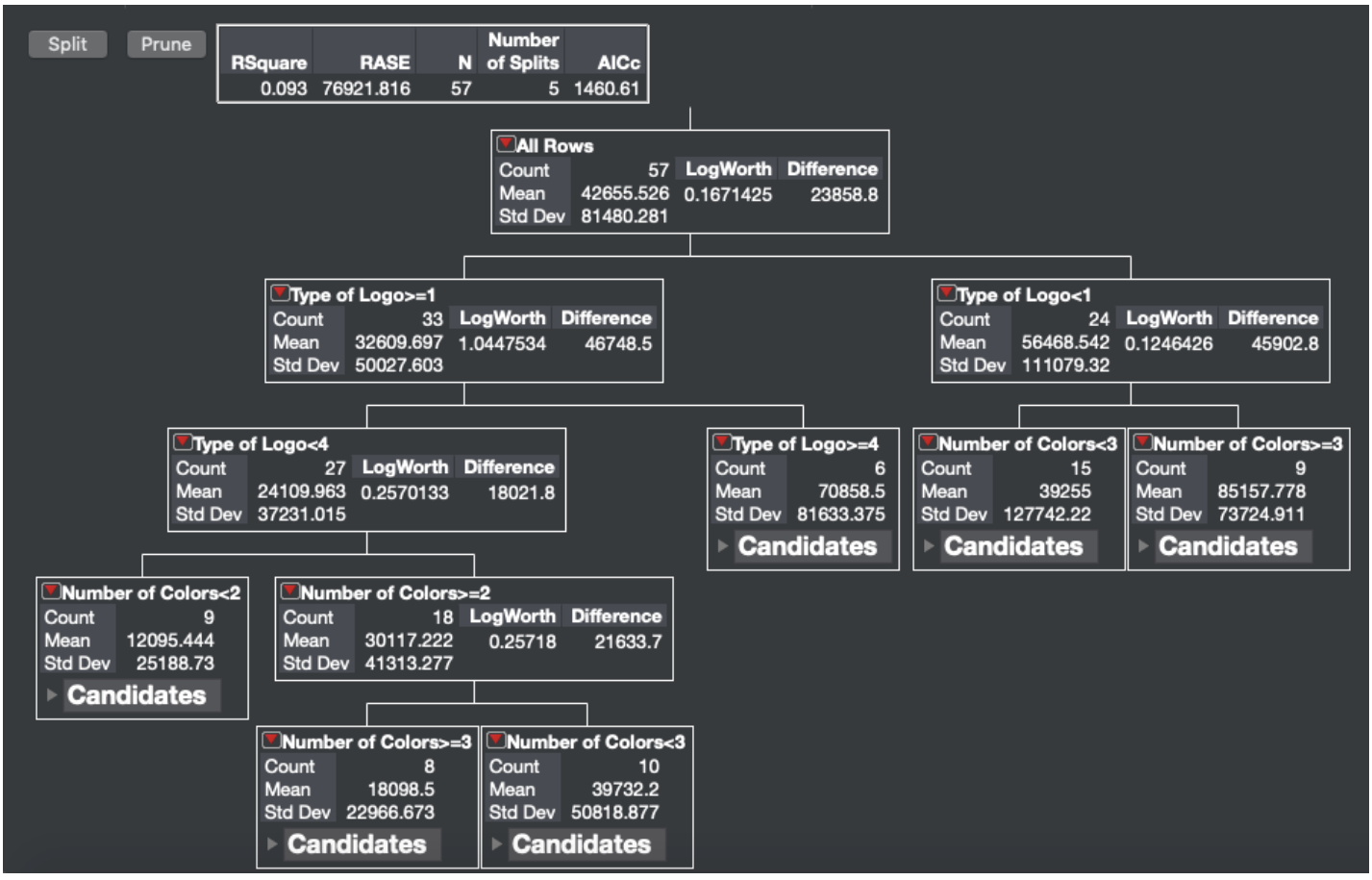
Overall this Decision tree found that only 16% of the cases in our dataset can be classified with companies that have high revenues.

Decision tree using JUMP

First, we did our full data decision tree analysis using JMP Pro 15. We saw one data that was way up high, and the difference between each child doesn't seem to be really significant. We were concerned that this data point would extremely skew our data. Therefore, we deleted the Walmart data point.

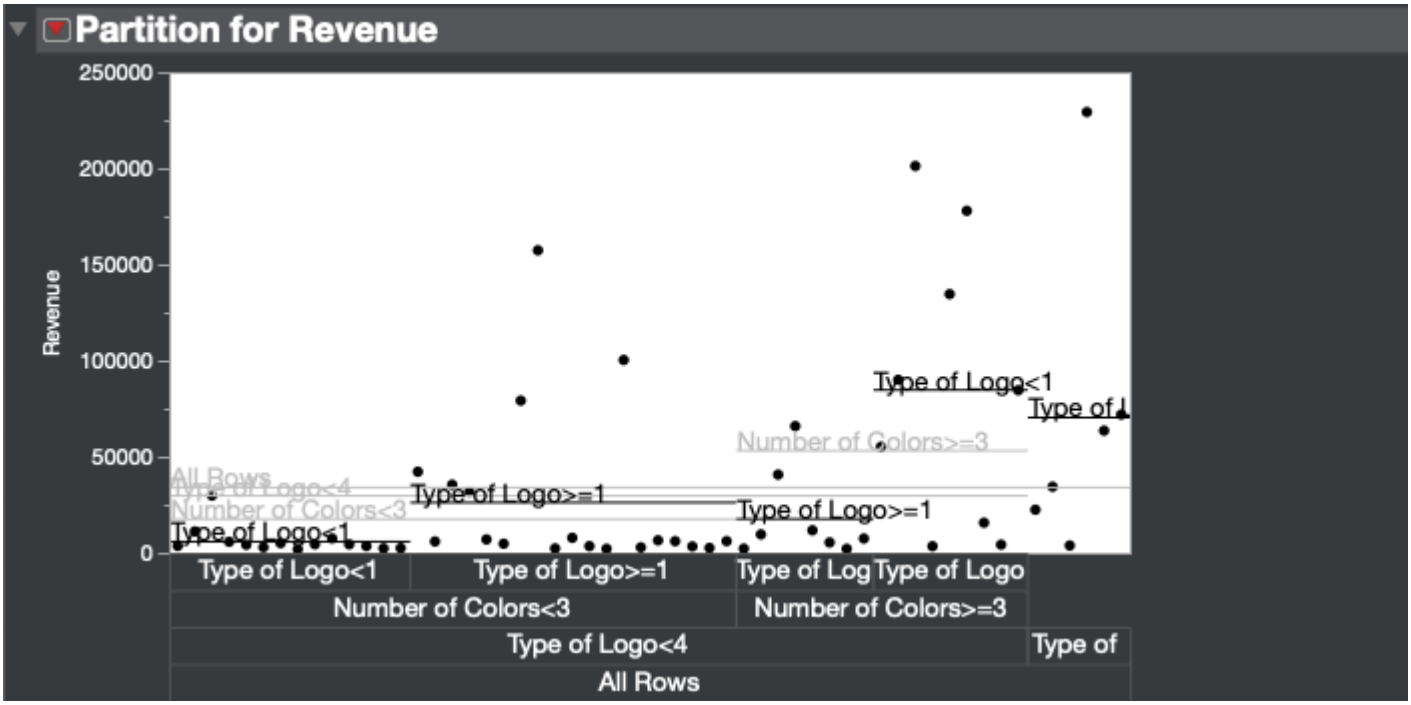


Alt text

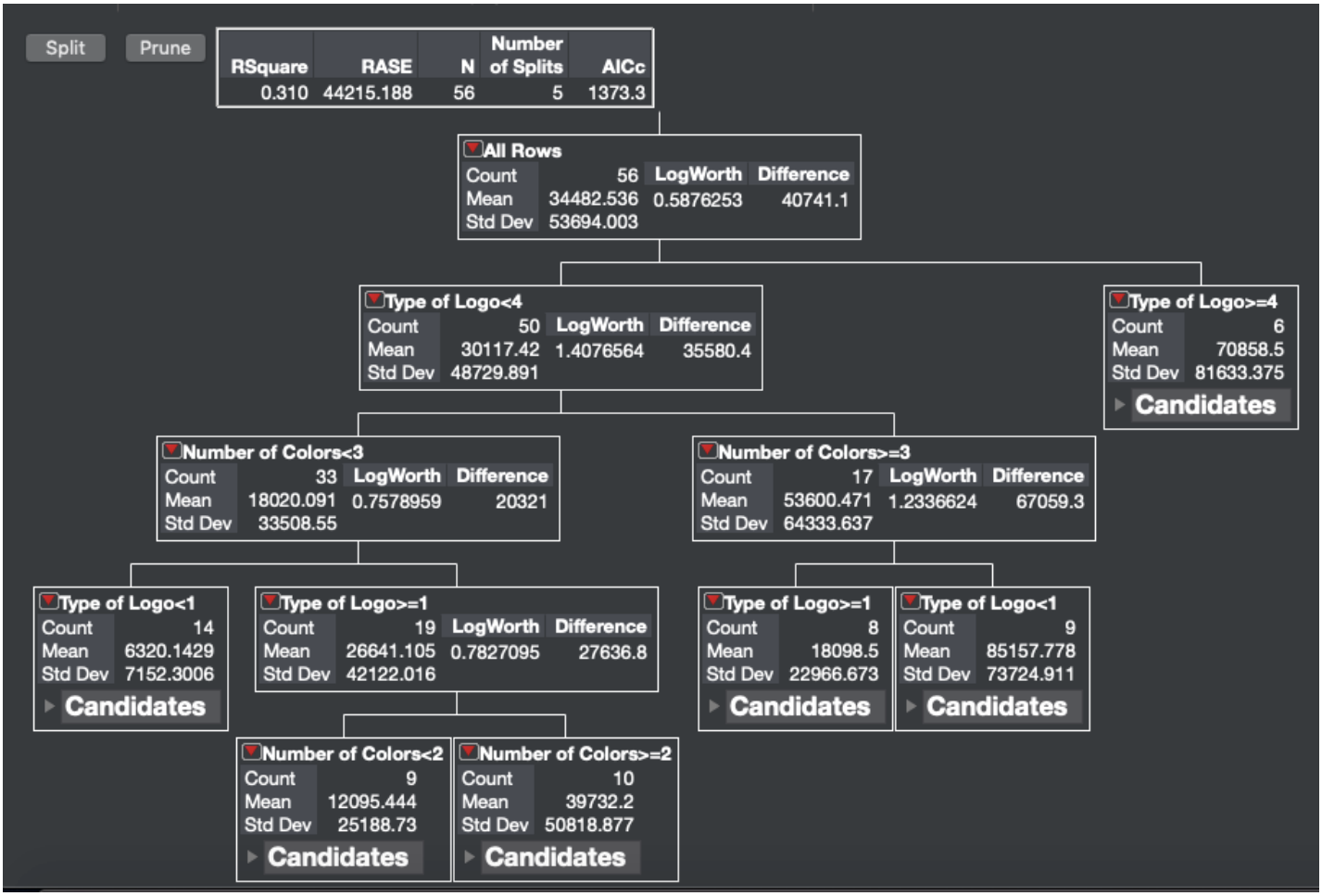


first partition

Our second decision tree analysis, after throwing out one outlier, looked so much better even though most data points are at the lower end of revenue and few data points scattered across the middle and higher end of revenue. Additionally, the children's partitions seemed to be more defined.



first partition



first partition

From both decision tree analysis, we can conclude that it is more likely for a company to have a higher revenue if their type of logo is a combination logo (`Type of logo` = 0) and have between 3 and 5 number of colors.

Creating the variable “top company”

Now we wish to use these informations to create a new variable called `top_company` so that we can define which company are having higher revenue based on the decision tree. So, we would consider a top company one case that have the `Type of logo` of 0, which is the Combinational Logo, and have more than 3 colors in the logo.

```
logorev_cleaned[,'top_company'] <- ifelse((Type_of_logo == 0) | (Number_of_colors >= 3), 1, 0)
```

Multiple regression

We would like to see if we can create a multiple regression for predicting `Revenue` based on the variable `Type of logo`, `Number of colors` and maybe `Gradient` and `Living in logo` if they are significant. This method is really effiecient when we have a Normal distribution of the data, no many outliers, the scatterplot of the data show a linear pattern, the variance is equal and that the variables are independent. As shown before we can only assume that our data follow independency, but from the scatterplots and histograms we understood that these variables are strongly skewed and non-normal. Therefore we are expecting not significan coefficients and very low `R^2`.

Multiple regression

For prediciting `Revenue` we first build a multiple regression with `Type of logo` and `Number of colors` as Independent Variables. For the summary of the model we reached an `R^2` of 0.002 which demonstrates that the power of explaning the variability with this model is almost absent. Infact, none of the coefficients are significant and therefore we would need to make some transformation so that we have normal distributions of the variables with equal variance and follow linearity.

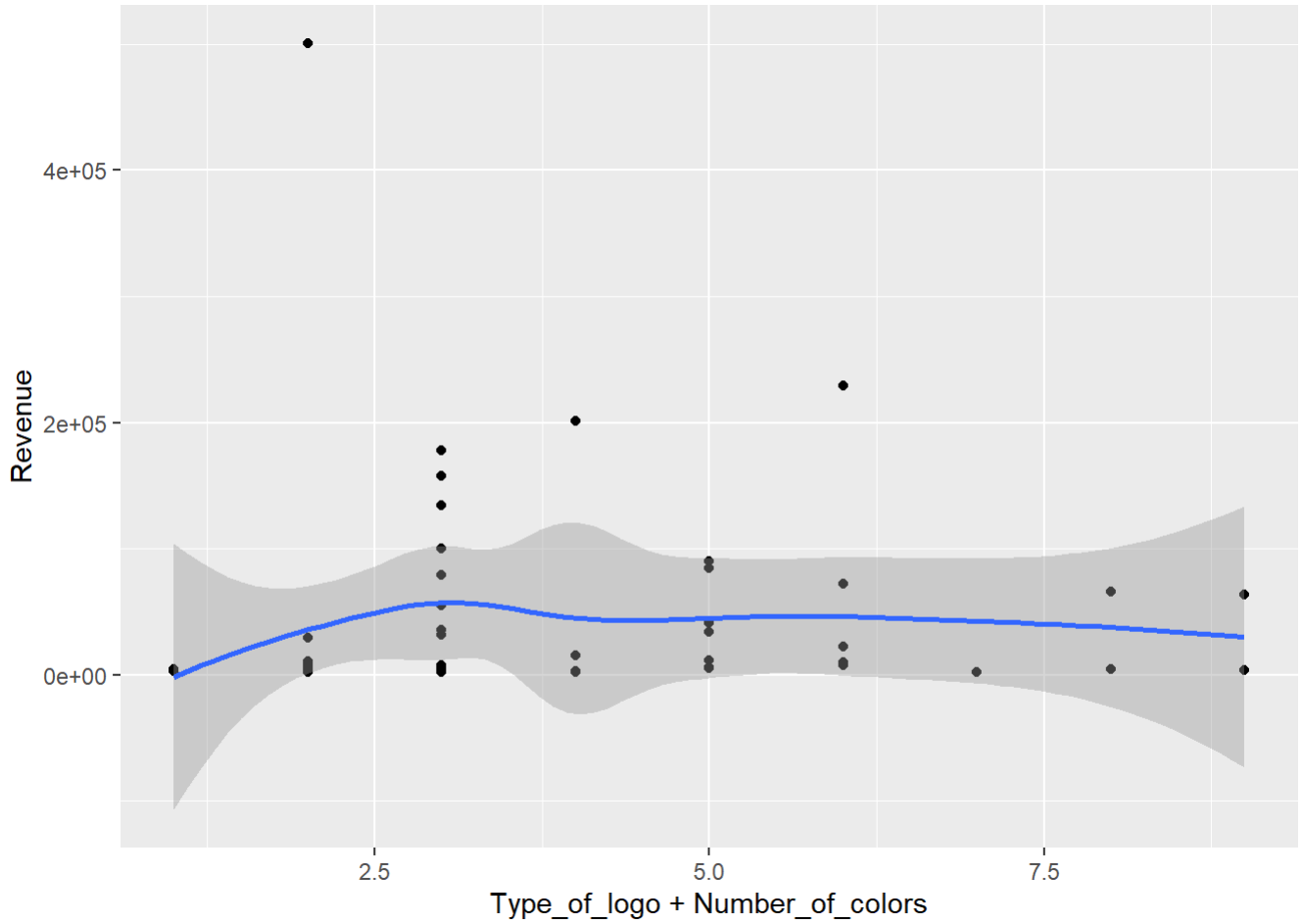
```
multiple_regression = lm(Revenue ~ Type_of_logo + Number_of_colors, data = logorev_cleaned)

summary(multiple_regression)
```

```
##
## Call:
## lm(formula = Revenue ~ Type_of_logo + Number_of_colors, data = logorev_cleaned)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -53037 -36692 -33917    328 459273
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)         35677      24475   1.458    0.151
## Type_of_logo          251       7480   0.034    0.973
## Number_of_colors     2697       7662   0.352    0.726
##
## Residual standard error: 82880 on 54 degrees of freedom
## Multiple R-squared:  0.002289,   Adjusted R-squared:  -0.03466
## F-statistic: 0.06194 on 2 and 54 DF,  p-value: 0.94
```

The scatterplot of the Multiple Regression also does not show any linear pattern.

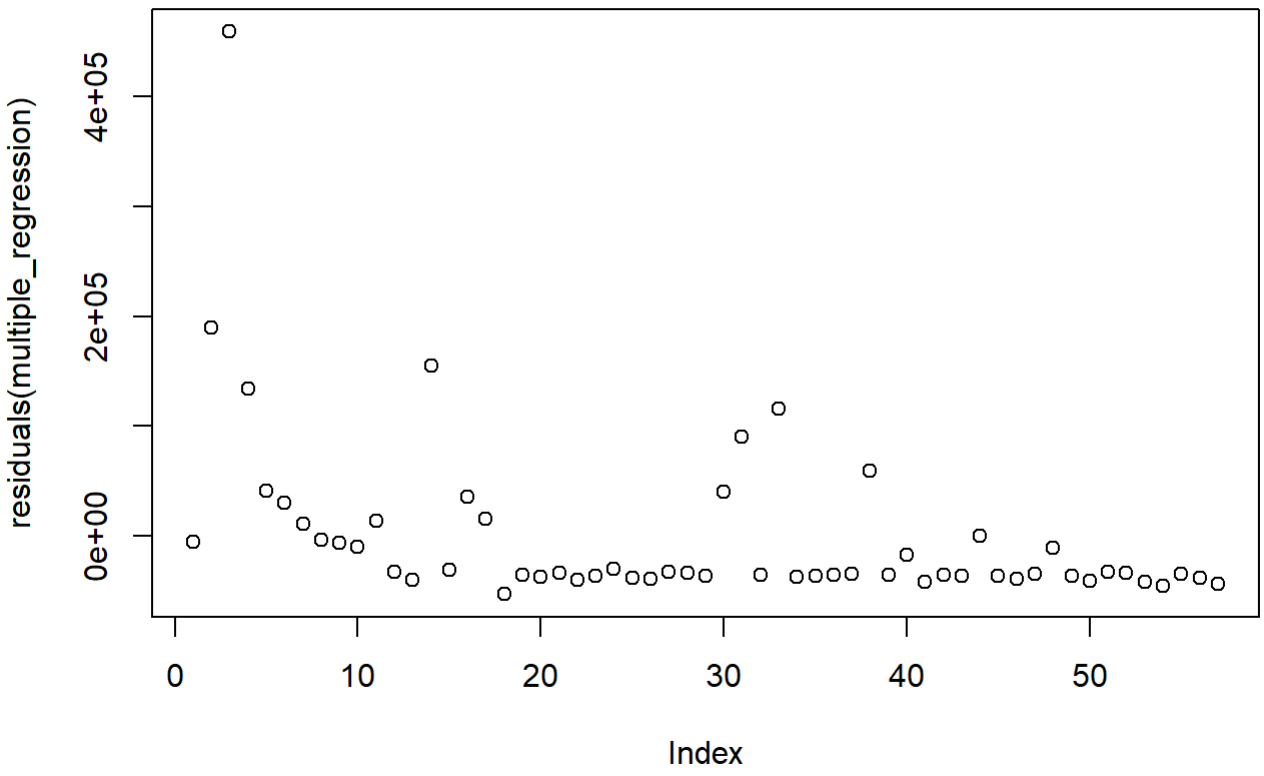
```
ggplot(logorev_cleaned, aes(x=Type_of_logo + Number_of_colors, y=Revenue)) +
  geom_point()+
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

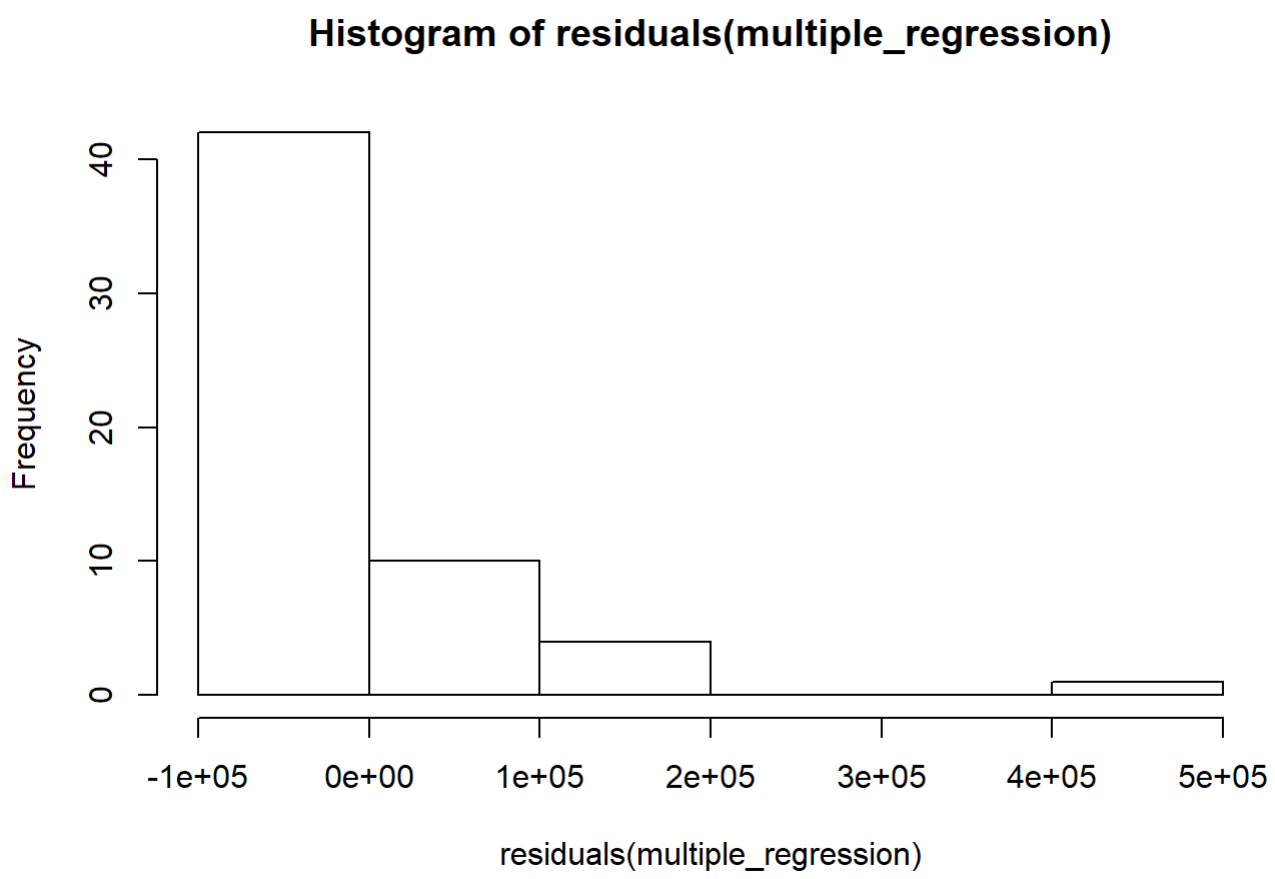


If we take a look at the resdal histogram we see that infact the multiple regression makes the risiduals not normal.The Residual vs. Fitted plot shows also that there are few outliers, but it looks pretty normal. The QQ plot is curved and that is becasue there are some big outliers.

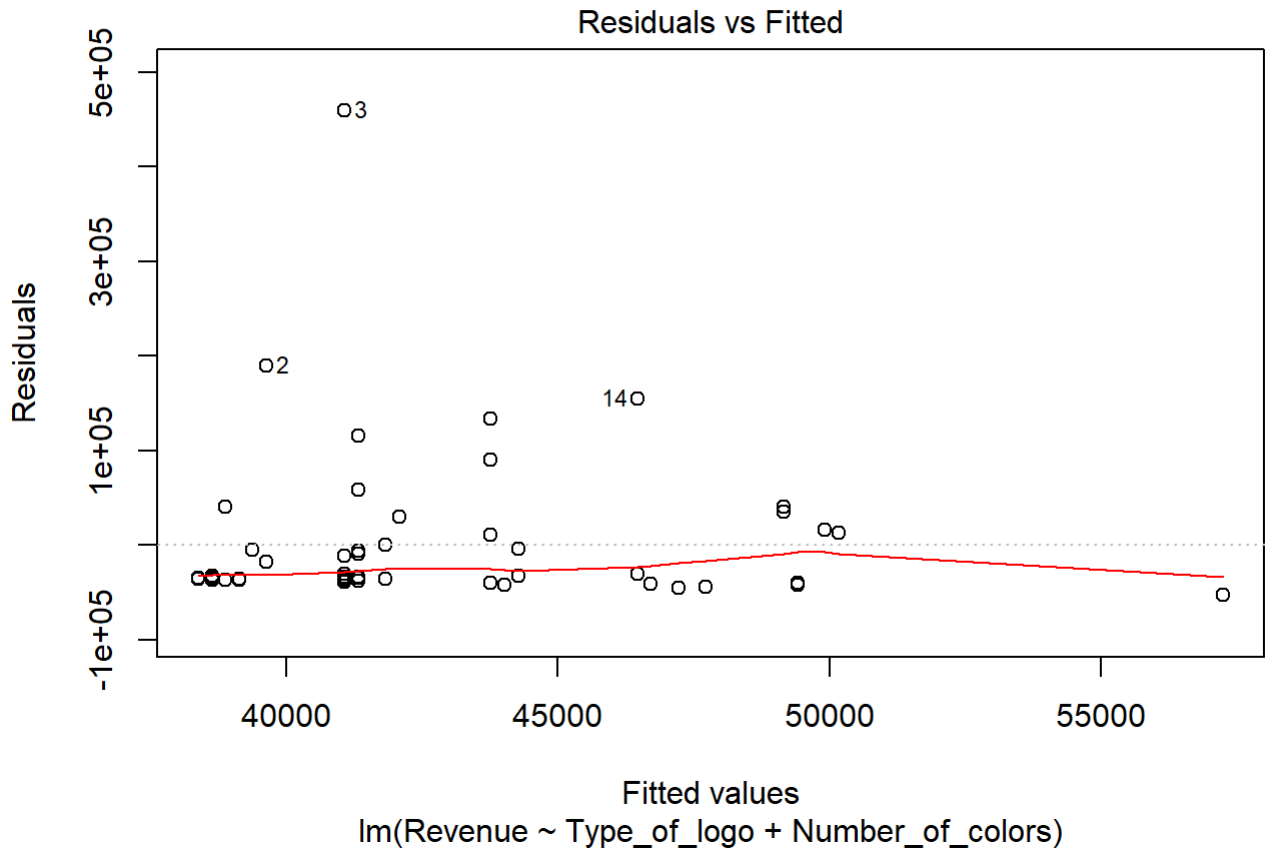
```
plot(residuals(multiple_regression))
```



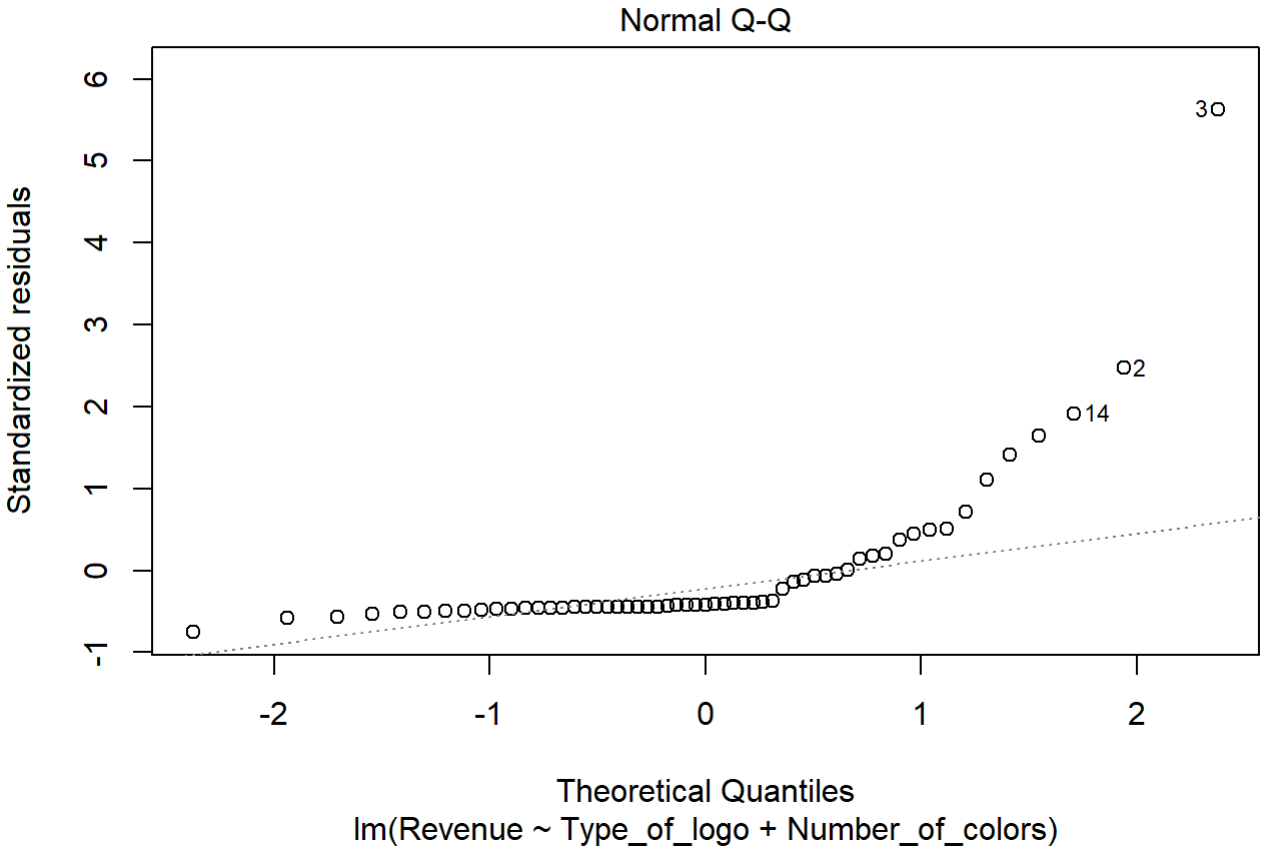

```
hist(residuals(multiple_regression))
```



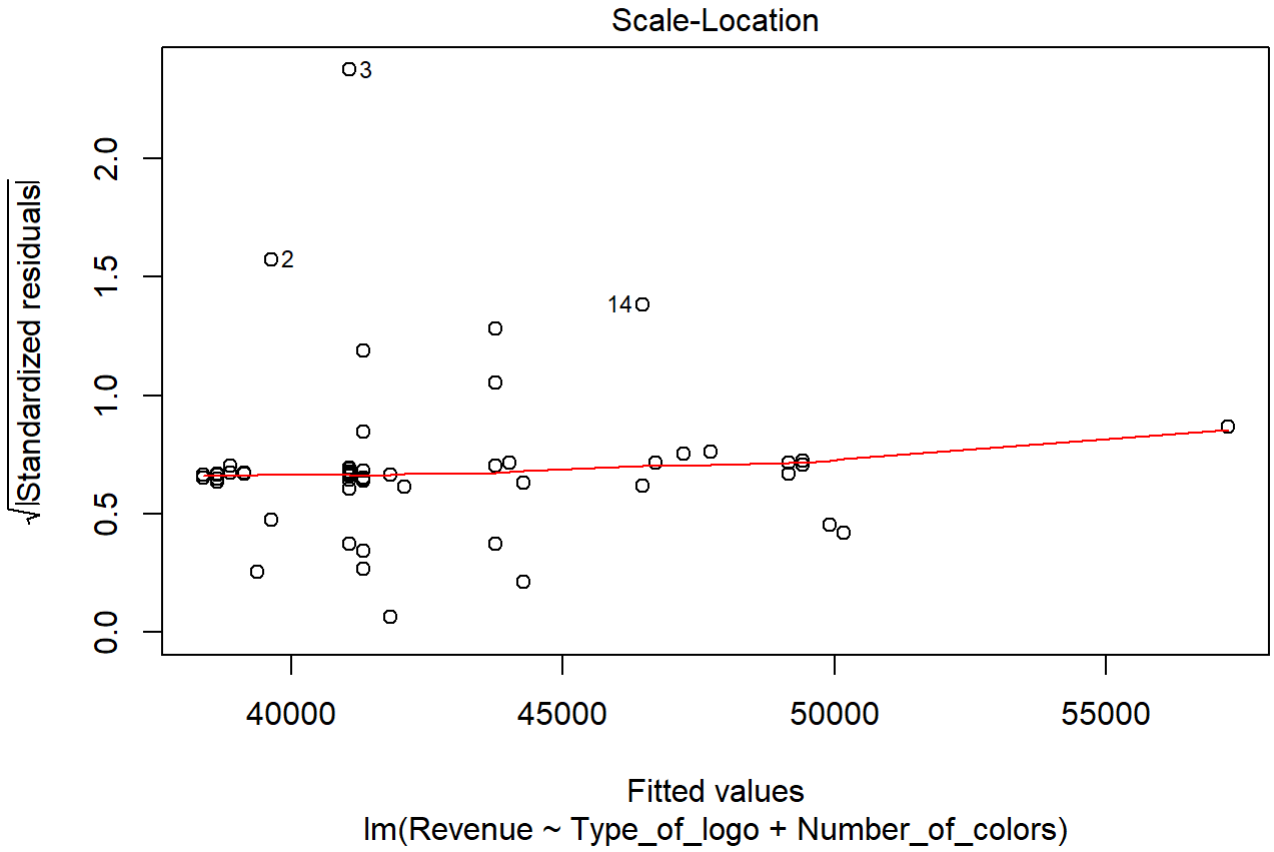

```
plot(multiple_regression, which = 1)
```



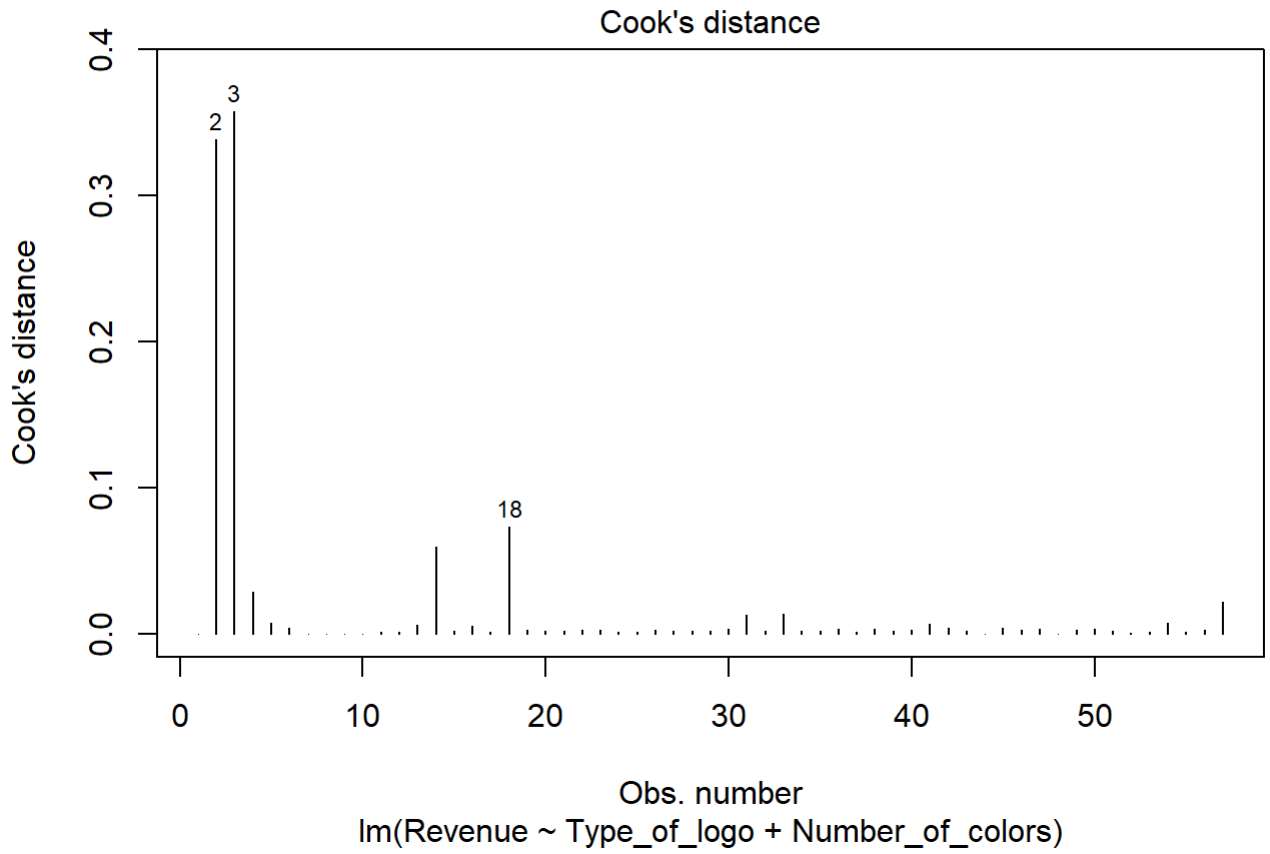

```
plot(multiple_regression, which = 2)
```




```
plot(multiple_regression, which = 3)
```




```
plot(multiple_regression, which = 4)
```



In conclusion we have build a weak and not useful multiple regression, becuase none of the predictors are significant and the `R^2` value is almost zero.

Multiple regression after deleting outliers and after a Log transformation

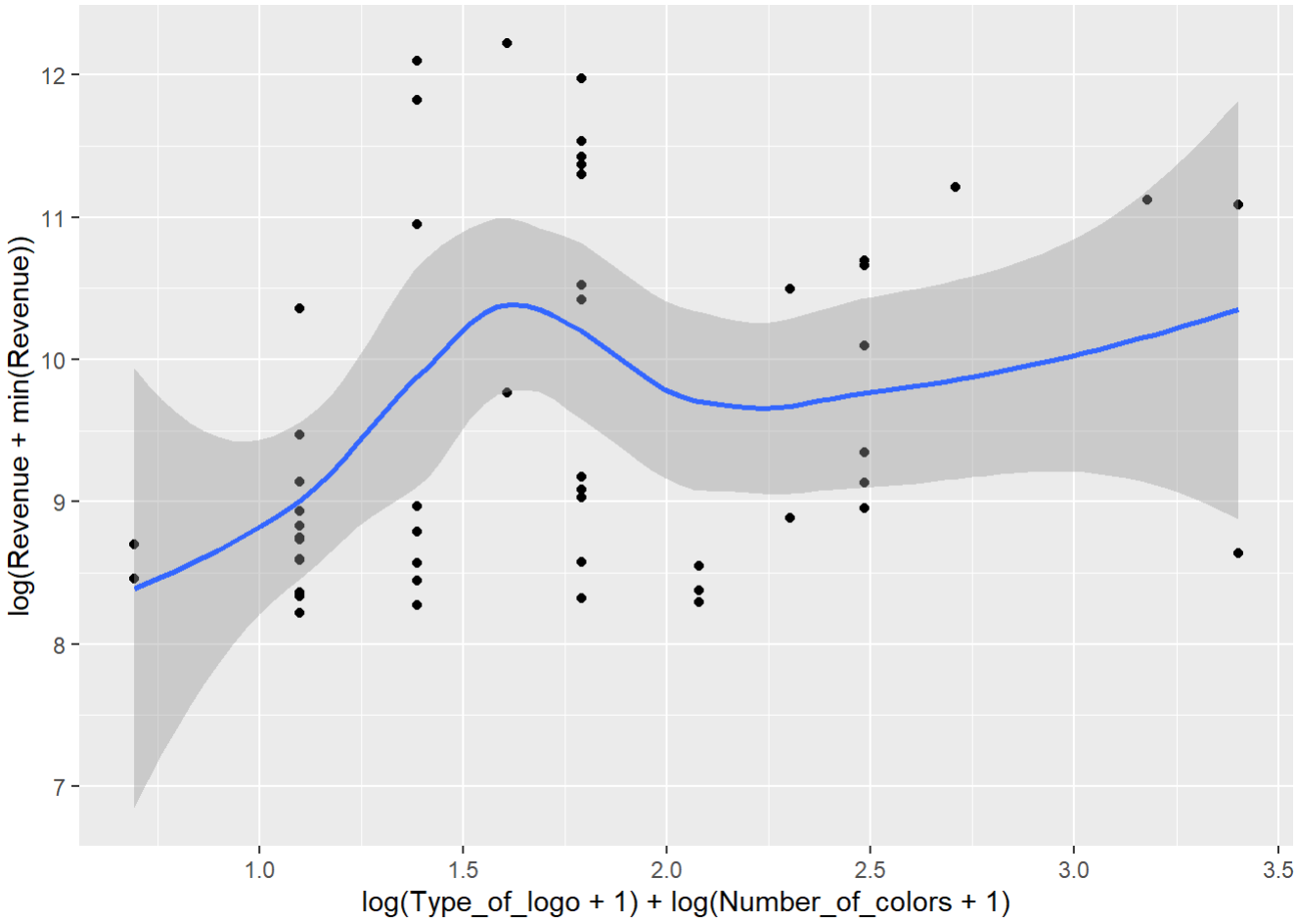
We can now try to use the information from the previous Multiple regression. We would first need to remove some outliers that are present in our dataset. The cook's distance plot showed us that some bossible outliers are 2, 3 and 18. From the other scatterplot, we would consider that also cases 51,54 and 12 can be considered as outliers. So we created a new dataset called `no_outliers` in which cases 3, 2, 18, 51, 54 and 12 are removed. Then we would also need to make the distribuition of `Revenue`, `Type of logo` and `number of colors` look normal, so we need to transform the variables with a **logarithmic** transformation (for which we would need to make all number positive and non zeros). The result of this transformation of the dataset is not perfect since the new multiple regression has a `R^2` of 0.17 which means that the model only explain 17% of the variability in the sample. The only significant coefficient in the multiple regression is the `log(Number of colors + 1)` which is very difficult to interpret. Overall, we can conclude that the Multiple regression model is unsatifactory.

```
no_outliers <- logorev_cleaned[- c(3, 2, 18, 51, 54, 12),] # Getting rid of the outlier
multiple_regression_2 = lm(log(Revenue + min(Revenue)) ~ log(Type_of_logo + 1) + log(Number_of_c
olors + 1), data = no_outliers)

summary(multiple_regression_2)
```

```
##
## Call:
## lm(formula = log(Revenue + min(Revenue)) ~ log(Type_of_logo +
##     1) + log(Number_of_colors + 1), data = no_outliers)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0332 -0.7965 -0.3404  0.9130  2.3901
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  7.7849     0.6132  12.697  < 2e-16 ***
## log(Type_of_logo + 1)        0.3124     0.2785   1.121  0.26768
## log(Number_of_colors + 1)    1.4438     0.4712   3.064  0.00358 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.163 on 48 degrees of freedom
## Multiple R-squared:  0.1706, Adjusted R-squared:  0.136
## F-statistic: 4.936 on 2 and 48 DF,  p-value: 0.01123
```

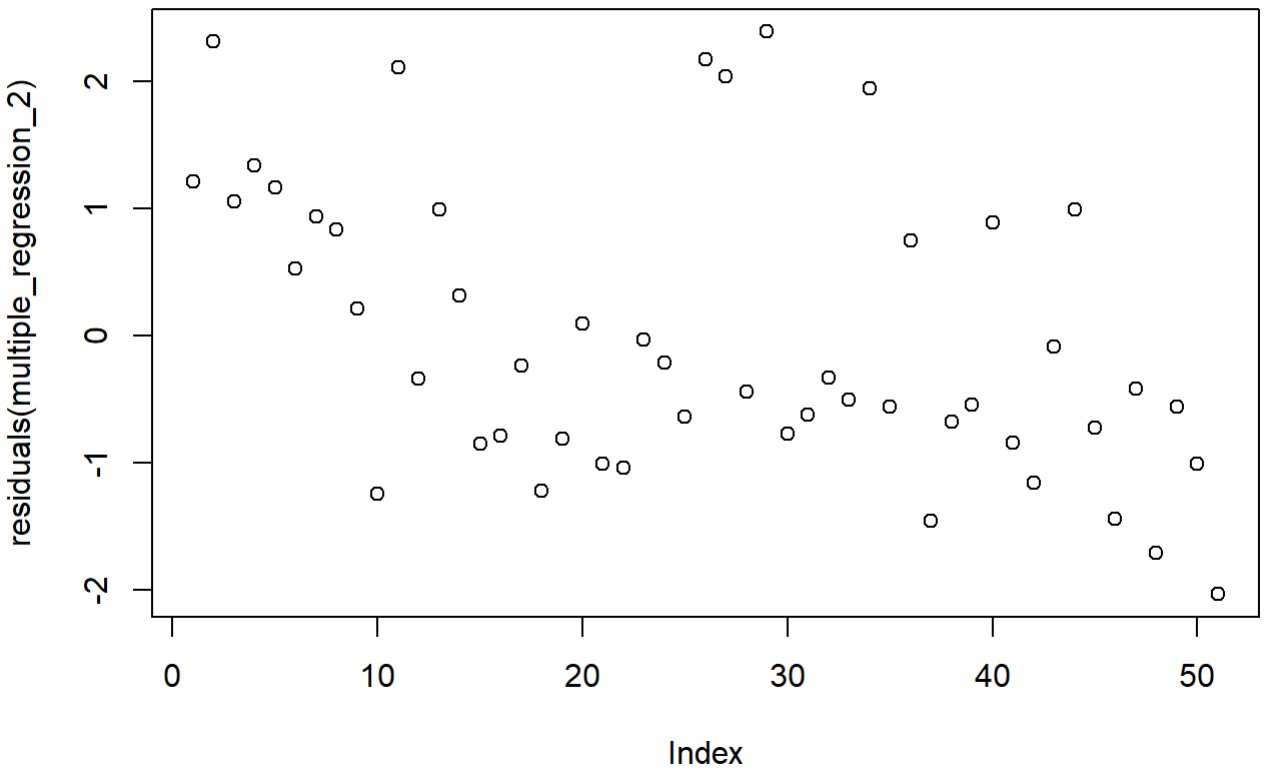
```
ggplot(no_outliers, aes(x=log(Type_of_logo + 1) + log(Number_of_colors + 1), y= log(Revenue + mi
n(Revenue)))) +
  geom_point()+
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

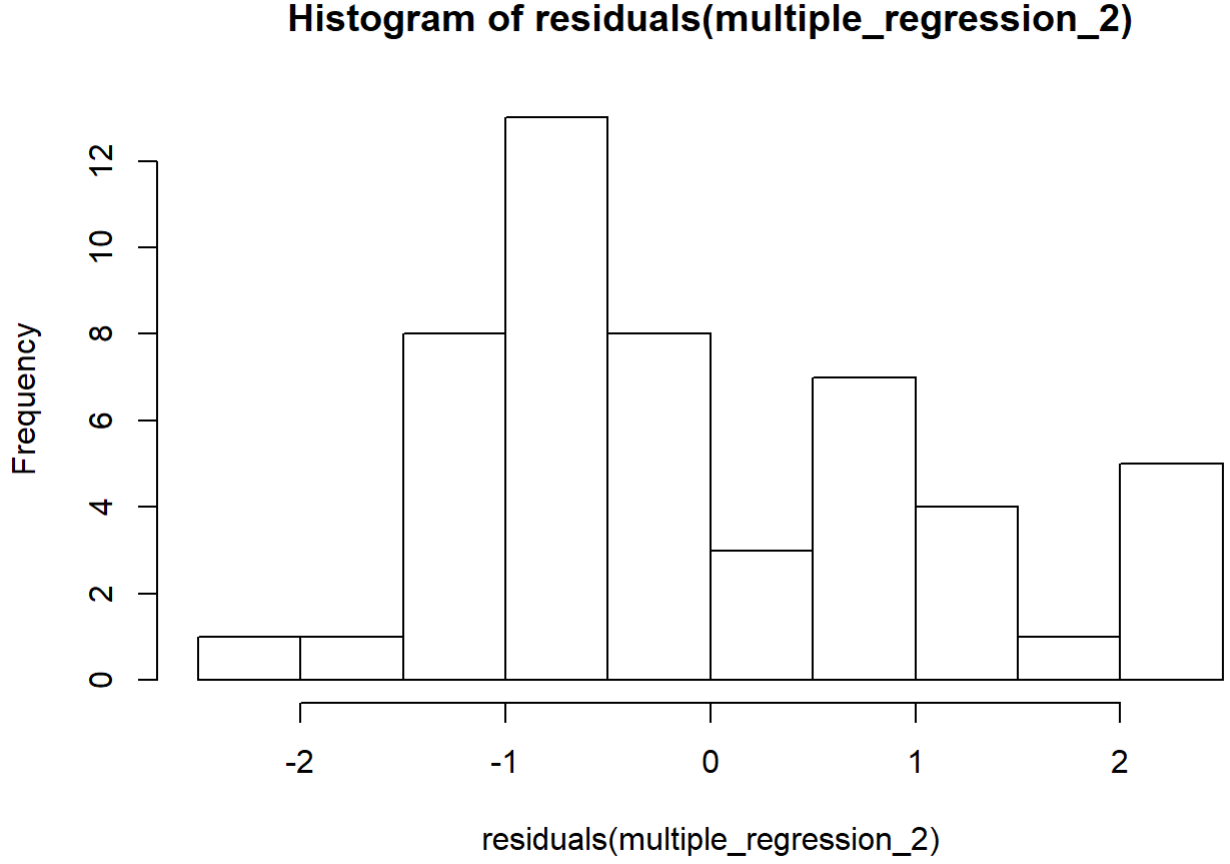


As stated at the beggining of this research paper we found that there could be more outliers. Infact, the new Cook's distance highlits more data points that can identify as outliers such as cases 11, 26 and 51. The histogram of the residuals looks more bell shape, but the QQ plot shows that the data does not fit the model.

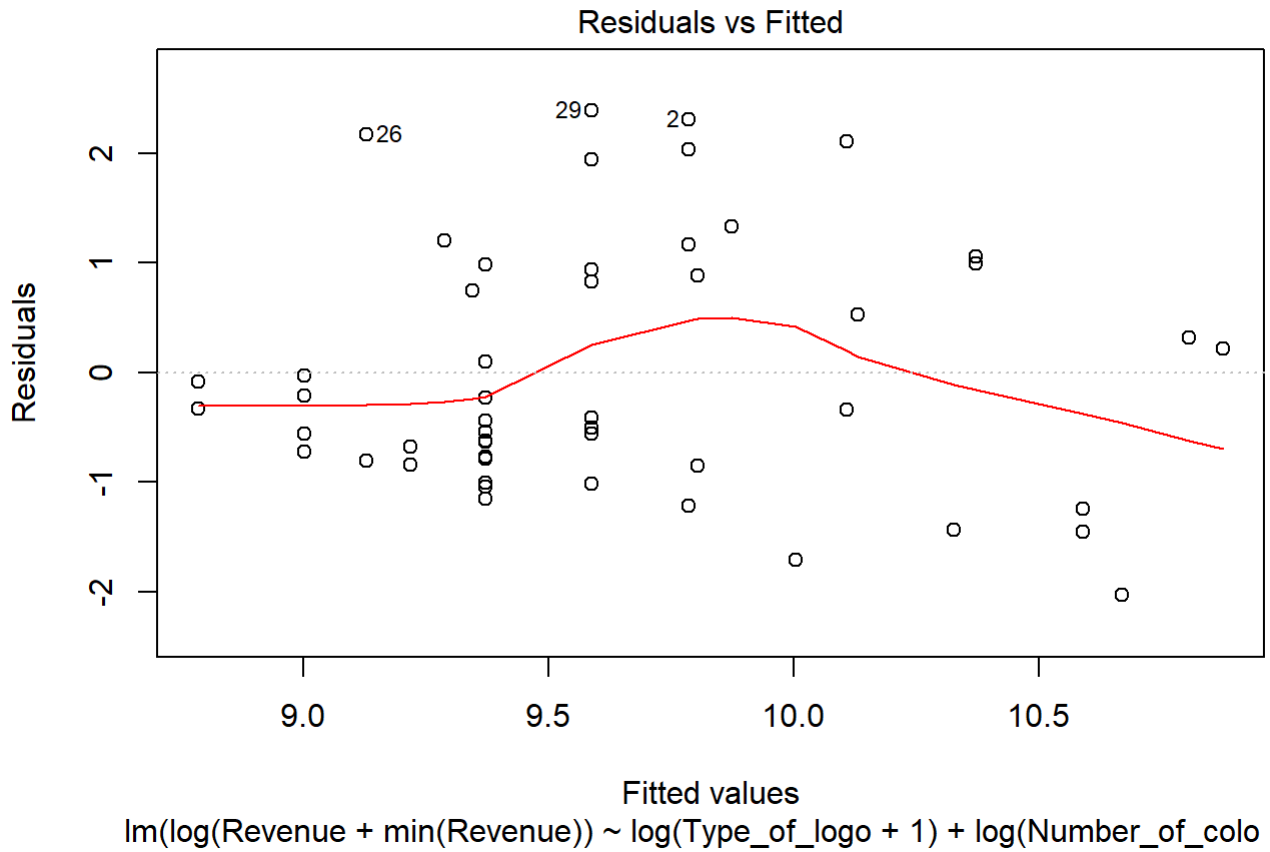
```
plot(residuals(multiple_regression_2))
```



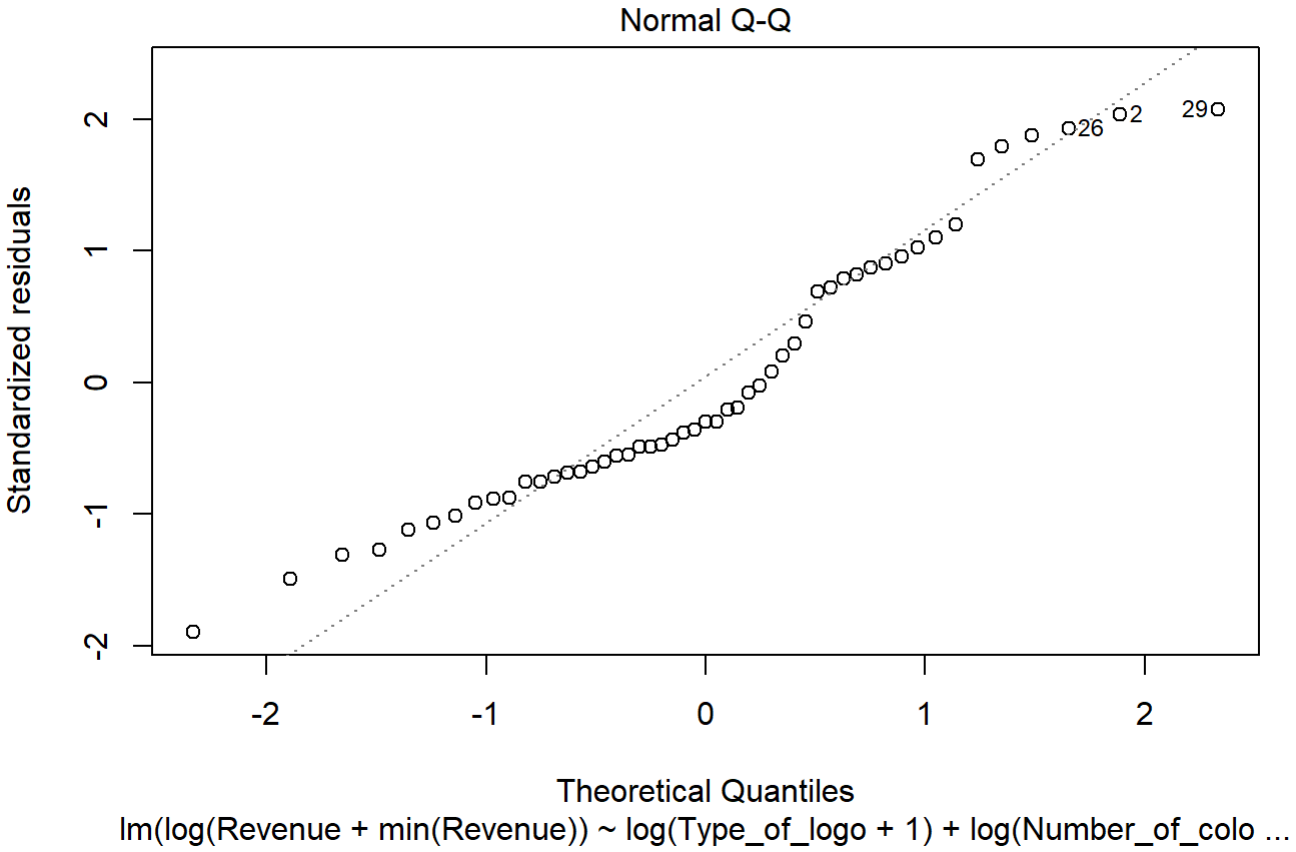

```
hist(residuals(multiple_regression_2))
```



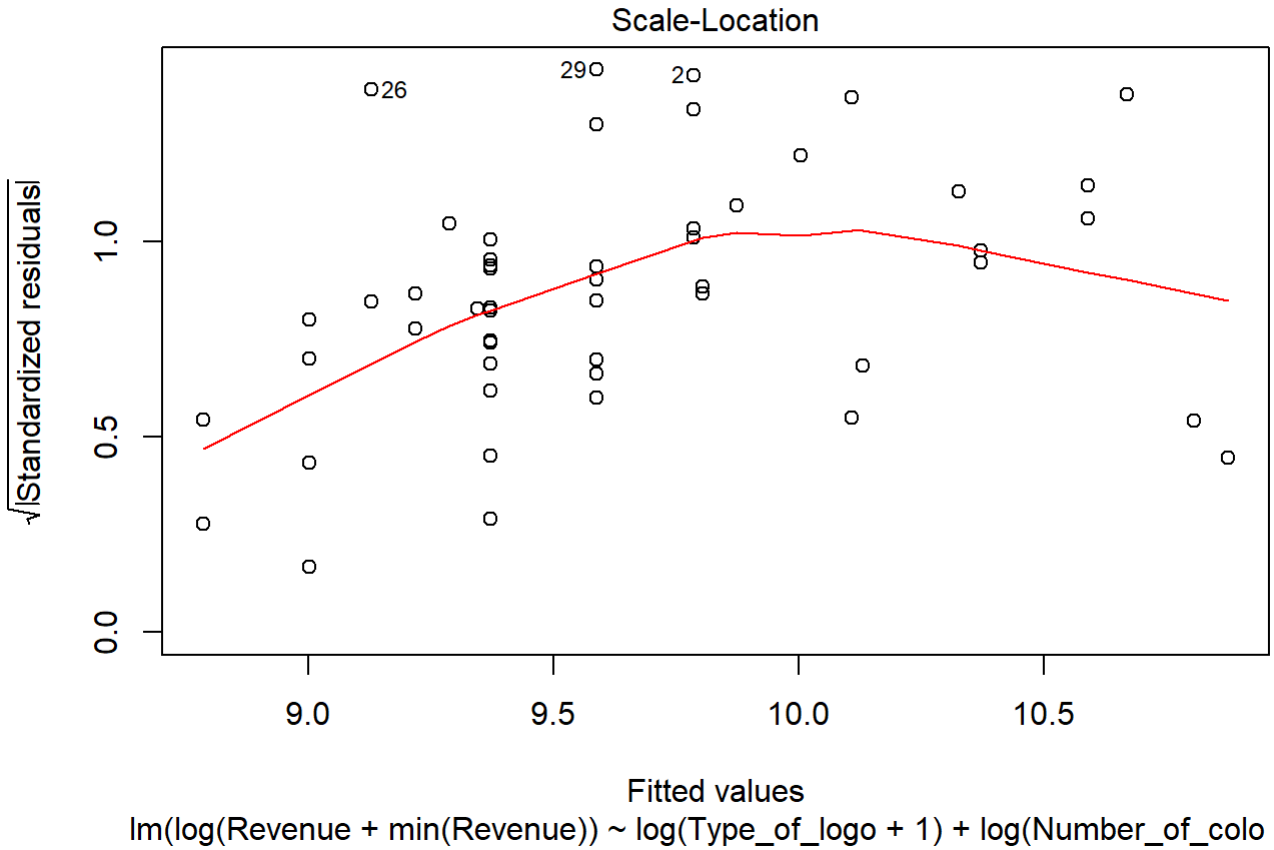

```
plot(multiple_regression_2, which = 1)
```



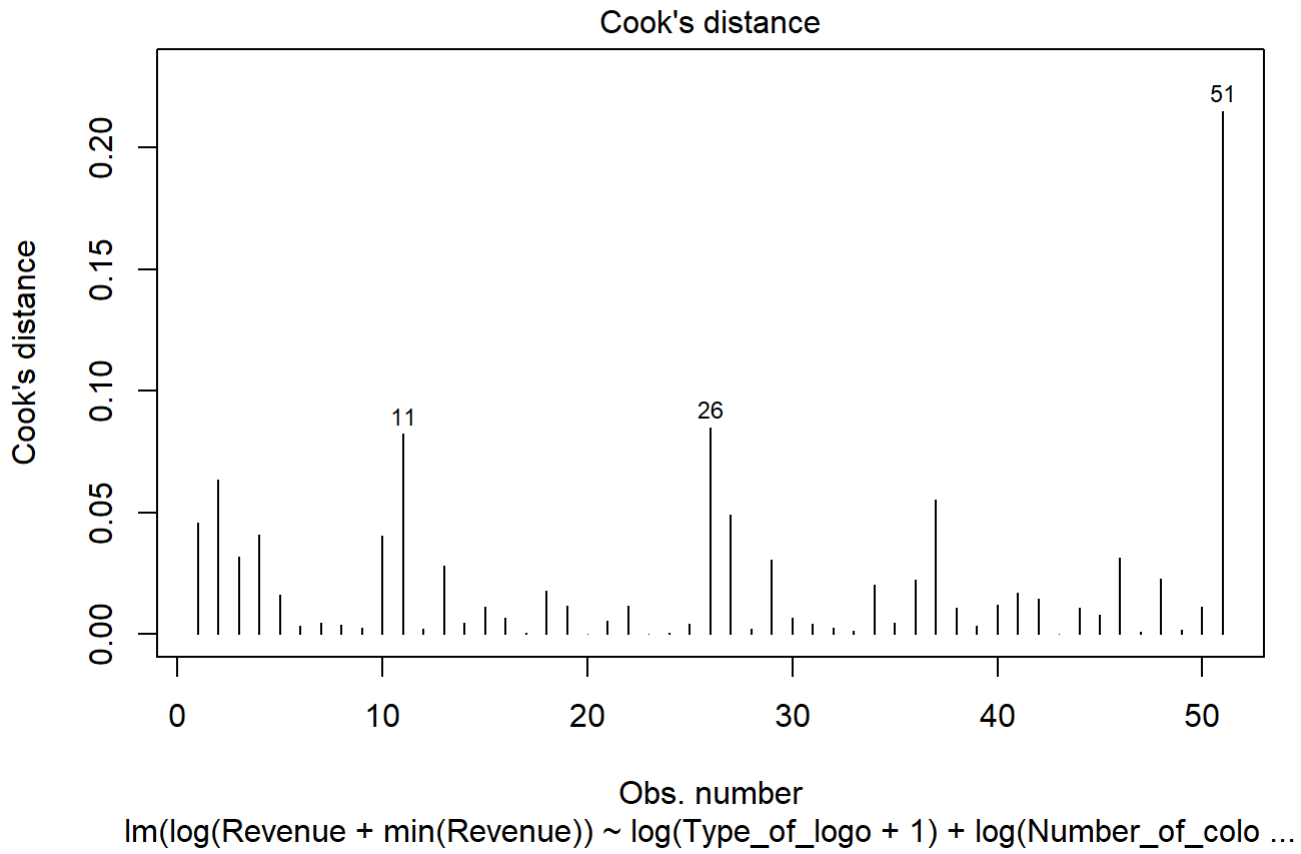

```
plot(multiple_regression_2, which = 2)
```




```
plot(multiple_regression_2, which = 3)
```




```
plot(multiple_regression_2, which = 4)
```



Logistic Regression

We can use Logistic regression to explain the relationship between `top company` and one or more nominal, ordinal, interval or ratio-level independent variables which can be `Type_of_logo` , `Number_of_colors` , `Gradient` and `Living_in_logo` . In our case we are intrestead in predict the odds of a logo to be classified as a top company, with the characteristics defined before by the Decision Tree.

Overall the Logistic Regression model has an AIC of 54.091, which means that the error involved in the logisitic regression is fairly large. Infact, we wish to lower the AIC statistics.

The resulting odds formula is : $ = e^{0.3139 - 0.14286 type of logo+Number of colors0.18211 - Gradient 0.04716 + Livining thing in logo 0.16942} $

Therefore we would expect that the odds for a company, that have an increse in the number of Colors, to have high revenue increases by factor of 0.18 lower, when all features remain the same.

Also form the Logistic Regression model we can predict that for a company that uses a specific type of logo greater than 0, the odds of being classified as a high revenue company would decrease by a factor of 0.14, when all features remain the same.

```
# Logisitic Regression
log_reg = glm(top_company ~ Type_of_logo + Number_of_colors + Gradient + Living_in_logo, data =
 logorev_cleaned)

summary(log_reg)
```

```
##
## Call:
## glm(formula = top_company ~ Type_of_logo + Number_of_colors +
##     Gradient + Living_in_logo, data = logorev_cleaned)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.72344  -0.24933   0.07564   0.32208   0.50420
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.31369    0.10836   2.895 0.005532 **
## Type_of_logo      -0.14286    0.03462  -4.126 0.000134 ***
## Number_of_colors   0.18211    0.03450   5.279 2.58e-06 ***
## Gradient1         -0.04716    0.17648  -0.267 0.790352
## Living_in_logo1    0.16942    0.19841   0.854 0.397075
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1343066)
##
##     Null deviance: 13.7193  on 56  degrees of freedom
## Residual deviance:  6.9839  on 52  degrees of freedom
## AIC: 54.091
##
## Number of Fisher Scoring iterations: 2
```

```
# plot of the logisitic regression
ggplot(logorev_cleaned, aes(x= Type_of_logo + Number_of_colors + Gradient + Living_in_logo, y=to
p_company)) + geom_point() +
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)
```

```
## Warning in Ops.factor(Type_of_logo + Number_of_colors, Gradient): '+' not
## meaningful for factors
```

```
## Warning in Ops.factor(Type_of_logo + Number_of_colors + Gradient,
## Living_in_logo): '+' not meaningful for factors
```

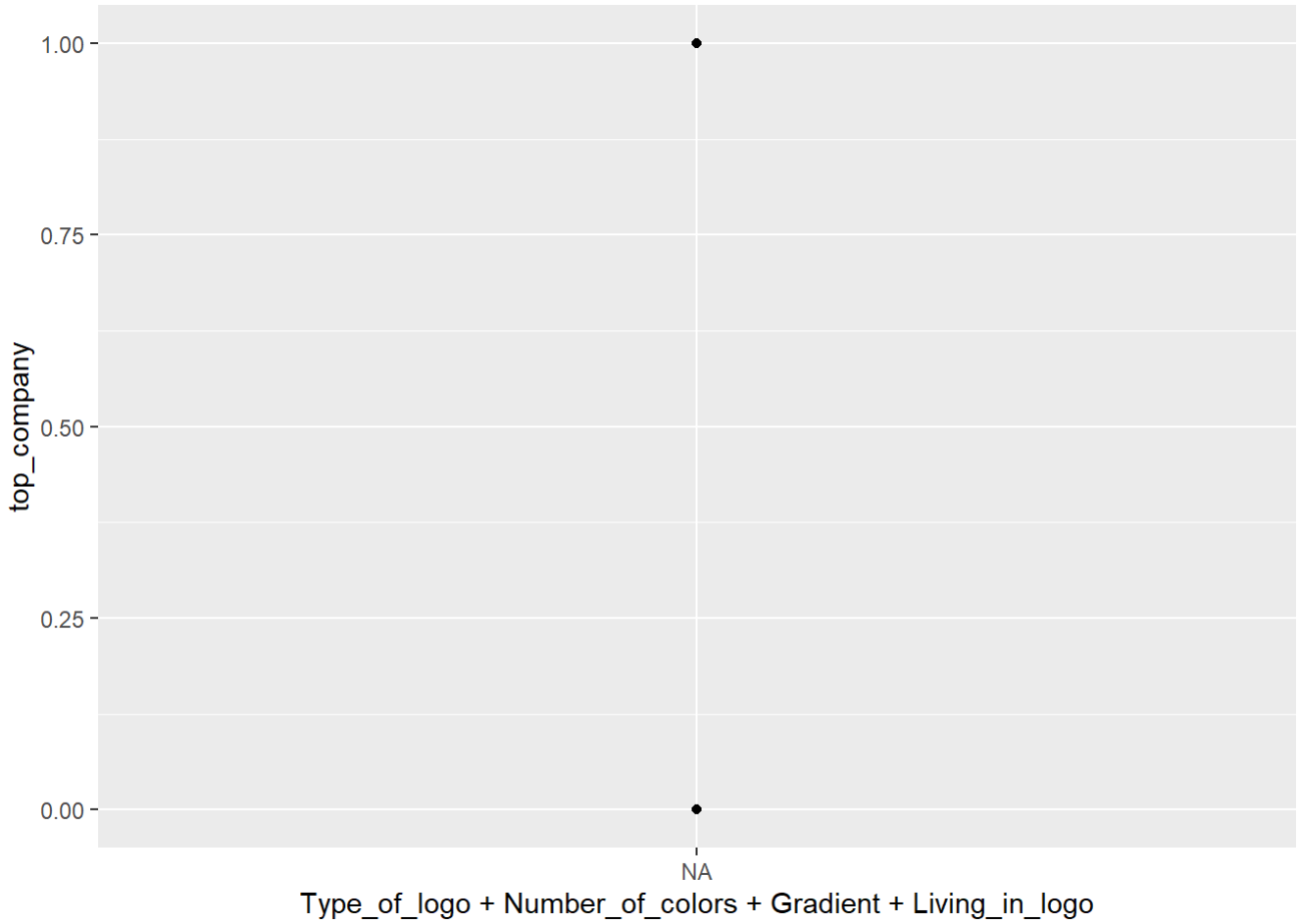
```
## Warning in Ops.factor(Type_of_logo + Number_of_colors, Gradient): '+' not
## meaningful for factors
```

```
## Warning in Ops.factor(Type_of_logo + Number_of_colors + Gradient,
## Living_in_logo): '+' not meaningful for factors
```

```
## Warning in Ops.factor(Type_of_logo + Number_of_colors, Gradient): '+' not
## meaningful for factors
```

```
## Warning in Ops.factor(Type_of_logo + Number_of_colors + Gradient,
## Living_in_logo): '+' not meaningful for factors
```

```
## `geom_smooth()` using formula 'y ~ x'
```



As we did for the Multiple Regression model we would also like to build a Logistic Regression model that would use the dataset without the major outliers and that it would use a **logarithmic** transformation of the variables to reach a state of normality in the Independet Variables ($log(Typeoflogo + 1) + log(Numberofcolors + 1) + Gradient + Livinginlogo$).

We can conclude that, after making the transfomation and after using the dataset without the outliers, we have built a better Logisitic Regression since our **AIC** is less ($AIC = 31.78$) than the previous Logistic Regression model. Although the AIC is better, the only two coefficient that have a significant p-value are the variable `Type of Logo` and the variable `Number of Colors` .

The resulting odds formula is :
$\hat{topcompanyodds} = e^{-0.038 - 0.446 typeoflogo + 0.75 Numberofcolors - Gradient 0.107 + Livininghinginlogo 0.172}$

Therefore we would expect that the odds for a company, that have an increse in the number of Colors, to have high revenue increases by factor of 0.75 lower, when all features remain the same.

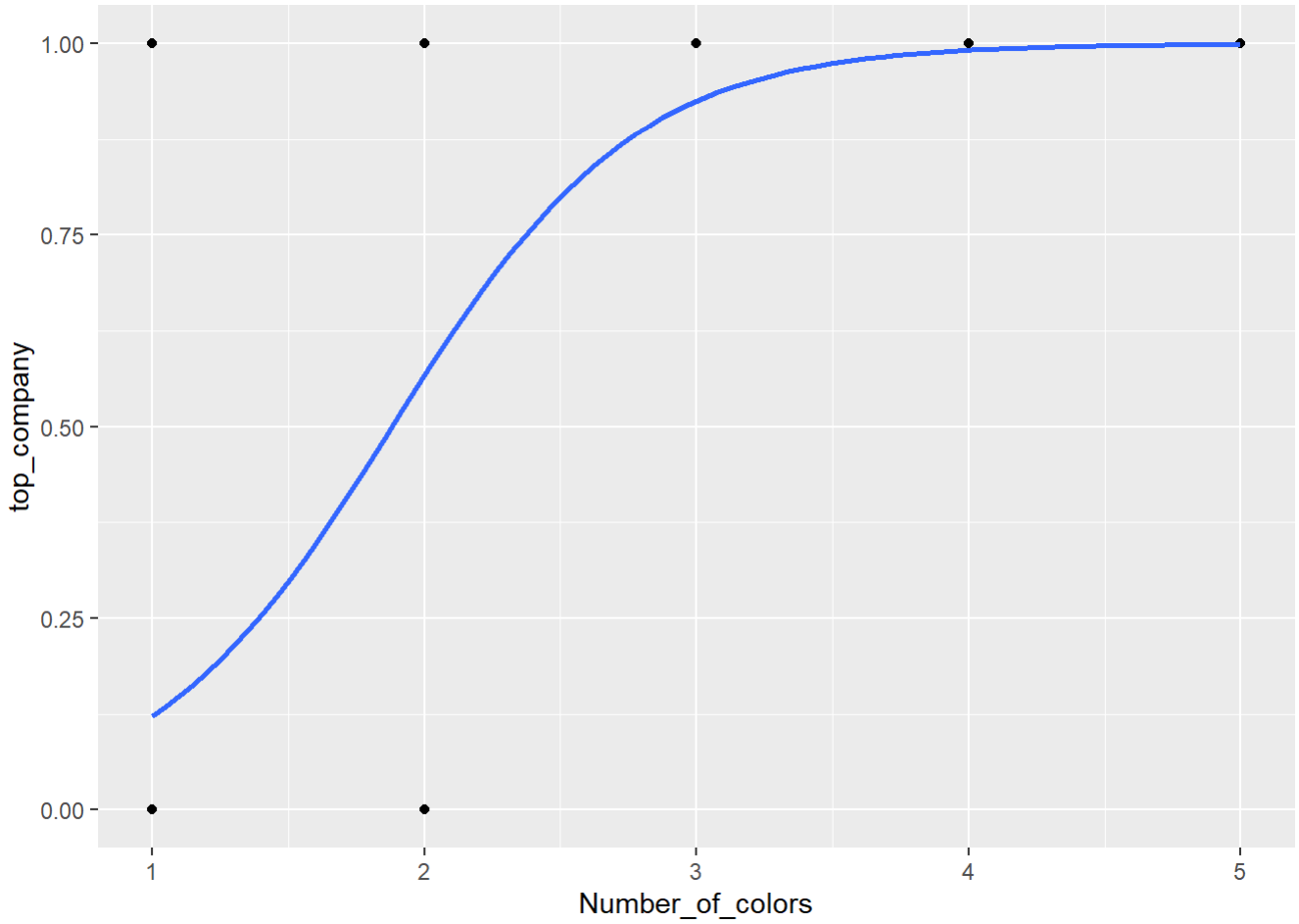
Also form the Logistic Regression model we can predict that for a company that uses a specific type of logo greater than 0, the odds of being classified as a high revenue company would decrease by a factor of 0.44, when all features remain the same.

```
# Logisitic Regression
log_reg = glm(top_company ~ log(Type_of_logo + 1) + log(Number_of_colors + 1) + Gradient + Livin
g_in_logo, data = no_outliers)

summary(log_reg)
```

```
##
## Call:
## glm(formula = top_company ~ log(Type_of_logo + 1) + log(Number_of_colors +
##     1) + Gradient + Living_in_logo, data = no_outliers)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.64790  -0.17221   0.00858   0.21454   0.51873
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -0.03876    0.16376  -0.237    0.814
## log(Type_of_logo + 1)      -0.44587    0.07767  -5.741 7.08e-07 ***
## log(Number_of_colors + 1)   0.75024    0.12538   5.984 3.06e-07 ***
## Gradient1                   0.10662    0.22777   0.468    0.642
## Living_in_logo1             0.17149    0.16597   1.033    0.307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.09567011)
##
##     Null deviance: 12.3529  on 50  degrees of freedom
## Residual deviance:  4.4008  on 46  degrees of freedom
## AIC: 31.78
##
## Number of Fisher Scoring iterations: 2
```

```
# plot of the logisitic regression
ggplot(no_outliers, aes(x= Number_of_colors, y=top_company)) + geom_point() +
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Neural Networks

Another useful method that we can use for predicting the revenue based on the characteristics of a company logo would be building a **Neural Network**.

```
library(caTools)
library(neuralnet)
```

```
##
## Attaching package: 'neuralnet'
```

```
## The following object is masked from 'package:dplyr':
##
##     compute
```

It would be still important to normalize the data before training a neural network on it. Therefore we would use the data set that eliminated the outliers and we would need to perform a **logistic** transformation.

```
no_outliers$Revenue <- log(no_outliers$Revenue + min(no_outliers$Revenue))
no_outliers$Type_of_logo <- log(no_outliers$Type_of_logo + 1)
no_outliers$Number_of_colors <- log(no_outliers$Number_of_colors + 1)
```

Now we would like to have a set of data which will be used for training the neural network and use the rest of the data as a test set to see how well the Neural Netweork works. The activation function used by the Neural Network is the default which is the Sigmoid activation function.

```
set.seed(101)

# Create Split (any column is fine)
split = sample.split(no_outliers$top_company, SplitRatio = 0.70)

# Split based off of split Boolean Vector
train = subset(no_outliers, split == TRUE)
test = subset(no_outliers, split == FALSE)

neural_net_model <- neuralnet(Revenue ~ Type_of_logo + Number_of_colors,train,hidden=c(3,3),line
ar.output=FALSE)

test
```

```
## # A tibble: 15 x 12
##    Name  Networth Revenue  Profit Ranking Font  Colors Number_of_colors
##    <chr> <chr>      <dbl>   <dbl>   <dbl> <chr> <chr>             <dbl>
##  1 Amaz~ $1 tril~   12.1   3033         8 sans~ #F399~            1.39
##  2 Face~ <NA>       10.7  40653        76 Klav~ #2536~            1.39
##  3 Coca~ $230 bi~   10.5   1248        87 Spen~ #BF2A~            1.10
##  4 Peps~ $18.8 b~   11.1   4857        45 Sans~ #0B0B~            1.79
##  5 H.B.~ <NA>        8.33    58.2     873 <NA>  #004B~            1.10
##  6 Chev~ <NA>       11.8   9195        13 <NA>  #0055~            1.39
##  7 Gene~ <NA>       12.0  -3864        10 <NA>  #2255~            1.10
##  8 Vent~ <NA>        8.60  1356.      652 <NA>  #00A2~            1.10
##  9 Harr~ <NA>        9.08   553       407 <NA>  #0000~            1.10
## 10 Bank~ <NA>       11.5  18232        24 Fran~ #E336~            1.10
## 11 Ciena <NA>        8.44  1262       770 <NA>  #C92E~            0.693
## 12 Mosa~ <NA>        9.13  -107.      382 <NA>  #0000~            1.79
## 13 Best~ <NA>       10.7   1000        72 <NA>  #0101~            1.10
## 14 Gene~ <NA>        8.89  -579       505 <NA>  #EB37~            1.61
## 15 Tera~ <NA>        8.30   -67       920 <NA>  #3A48~            1.39
## # ... with 4 more variables: Type_of_logo <dbl>, Living_in_logo <fct>,
## #   Gradient <fct>, top_company <dbl>
```

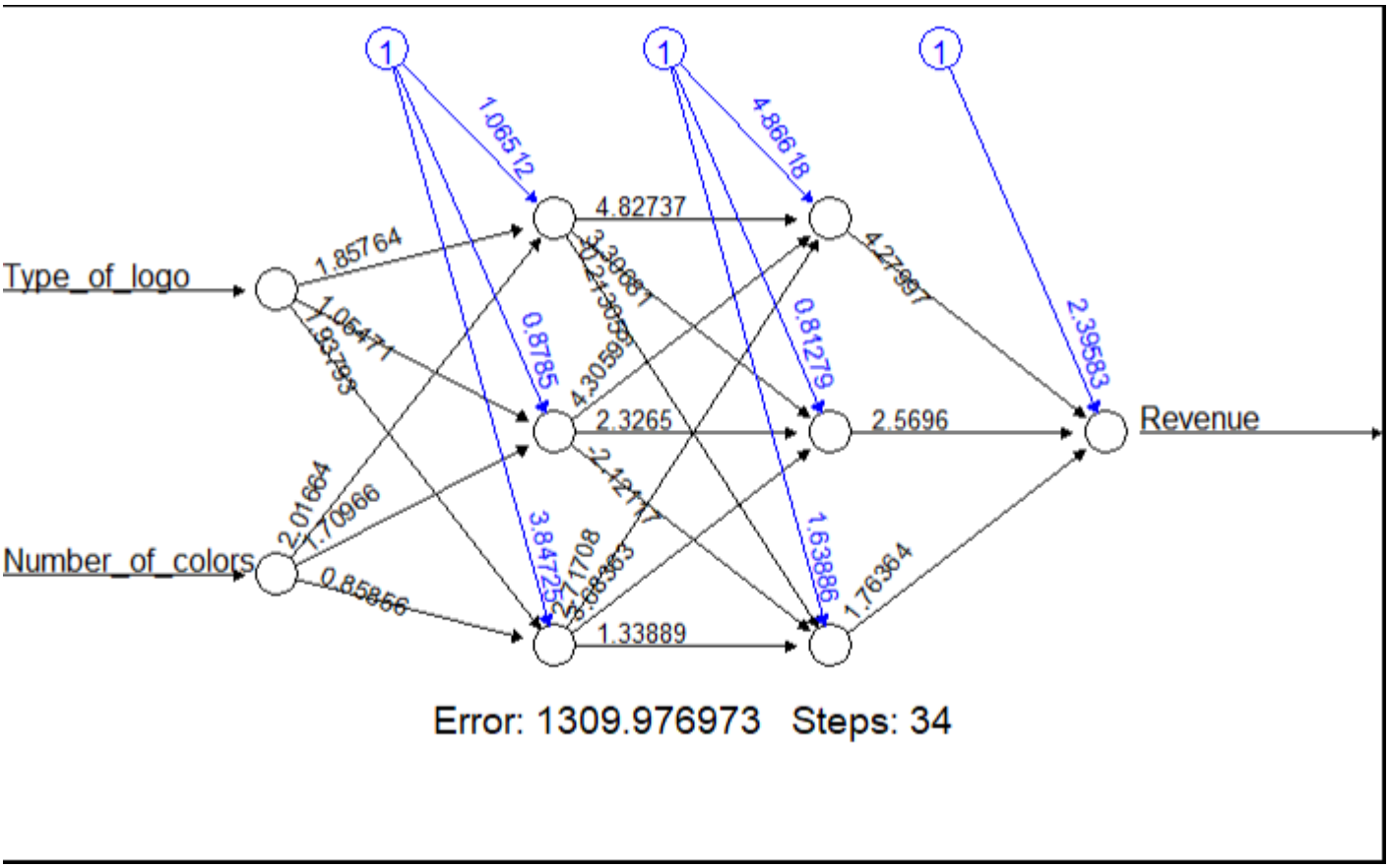
Our results are not perfect, because the neural network predicted for the first 6 cases in the test set to be high revenue companies, but if we check the `top_company` the result, based on the Decision Tree, we would expect case 3 to be not an high revenue company based on the logo characteristics.

```
# Compute Predictions off Test Set
predicted.neural_net_model.values <- compute(neural_net_model,test)

# Check out net.result
print(head(predicted.neural_net_model.values$net.result))
```

```
##           [,1]
## [1,] 0.9999705
## [2,] 0.9999699
## [3,] 0.9999703
## [4,] 0.9999697
## [5,] 0.9999710
## [6,] 0.9999705
```

In this plot we have the details of the two `hidden layers` .

```
plot(neural_net_model)
```



plot neural network

Conclusions

We were able to conclude that it is more likely for a company to have a higher revenue if their type of logo is a *combination logo* (`Type of logo` = 0) and have between 3 and 5 number of colors in the logo. However, our further statistical analysis suggested that we have may outliers that may interfere with out results. The major problem encountered in building the *Multiple Regression* and the *Logistic Regression* model was that the assumptions for Linearity, Equality of Variance and Normality were not met. Outliers were also present in the data, so the resulting models were not have enough explanatory power. We wish we had more data, but doing so would require more time since we collected the majority of the data manually. Therefore, for future studies, we would suggest for a bigger data and still keep the randomization aspect of data sampling like we did. One of the cause we could think of was because our randomized sample reflected multiple sectors or disciplines of the market. We did not take companies from a specific economic sector but, indeed, we took companies from many different sector. This also could have affected out data, because revenues are different depending on the market sector or

on the company type. Therefore, for future studies, we would suggest for a bigger data and still keep the randomization aspect of data sampling like we did. Another suggestion we have is doing separate statistical analysis for different range of company's rankings. For example, we will have a separate analysis for company in ranking 1-100, another analysis for company in ranking 101-200, and so. This may eliminate extreme outliers in data analysis.