Neurological Assesment Analysis Report

Ryota Nishida

Executive Summary

This report summarizes a graphical exploratory analysis of the number of mistakes performed when partaking in a neurological assessment through a trail making app.The analysis is done using data taken from a small sample of subjects performing two different types of tests over the span of 5 days. The analysis indicates that the possible factors that influence the number of errors made during the test are variables such as test type, duration, and attempts.

Data Description

The data set comes from a file called "errors.R" and has been entered into R using the source function. The data was collected from a group of 25 subjects who used two different types of neurological tests on a Trail Making App. Each participant took 50 attempts at the Trail Making app using both using both types (A and B).10 attempts were done on A followed by 10 on B each day for 5 days. Duration was measured for each attempt.Numbers of errors made by subjects were also recorded. The five categories are:

- Name: Subject number(e.g SUB01)
- Types: Test type(A or B)
- Errors: Number of Errors committed during test
- Duration: Time taken to complete test
- Attempts: The number of the attempt being taken

Objectives

The errors data will be used to see how many mistakes the subjects make on the different test types. The data are divided into two groups by test type. The differences and similarities in the number of mistakes performed for the different test types will be explored.Also, a model will be built that can determine possible factors that influence the number of errors made during the test. The factors that will be observed for their impact on the number of mistakes are test type, Duration, and Attempts. Then some conclusions about the possible factors that influence the number of errors made while using a trail making app can be drawn from this study and applied to the larger population from which the sample was taken, giving us an idea about how trail neurological tests differ.

Methods

A few graphical techniques will be employed; these will include box-plots,compound residual plots,scatter plots,and histograms.Box plots are used for displaying basic distributional features of uni-variate data, but they can be extended to multivariate settings by being placed side-by-side. We use side-by-side box-plots to compare means and outliers in relation to the dependent variable Errors and the regressors of Duration, Attempts, and test type.

Scatter plots help to showcase the effects of the regression model in a visual manner. The use of a curve of fit helps to also explain the data within a scatter plot by helping the viewer to observe the relationship between

the dependent variable and regressors and showcasing things like whether the model is linear or non linear in nature.Scatter plots are useful for displaying bivariate data, but they can also be extended in a limited way to reveal multivariate patterns through the use of pairwise scatter plots, where each pair of variables in the data set is used to produce a scatter plot.

The compound residual plot will showcase: a residual vs fitted,scale location,constant leverage, and QQ plot. These plots will give insights about the distribution and linearity of the errors data set. Along with this, an understanding of the spread and scedasticity of the data set can also be gained in order to test out different assumptions. The histogram will provide a visual interpretation of the numerical data by showing the number of data points that fall within a specified range of values (called "bins"). The histogram will be used to obtain a first overview of the dependent variable.

Results and Discussion

```
##       Name      Type         Errors          Duration         Attempts
##  SUB22  : 100   A:1250   Min.   : 0.000   Min.   : 5.954   Min.   : 1.0
##  SUB14  : 100   B:1250   1st Qu.: 0.000   1st Qu.: 9.640   1st Qu.:13.0
##  SUB20  : 100            Median : 1.000   Median :11.367   Median :25.5
##  SUB15  : 100            Mean   : 1.674   Mean   :12.604   Mean   :25.5
##  SUB19  : 100            3rd Qu.: 2.000   3rd Qu.:14.135   3rd Qu.:38.0
##  SUB06  : 100            Max.   :19.000   Max.   :98.855   Max.   :50.0
##  (Other):1900
```

A quick summary of the data frame has shown that there are 25 participants each with 100 values. It also shows that there is a factor called type with two levels. The factor errors has a mean of 1.674 and median of 1. Duration has a mean of 12.604 and a median of 11.367. Attempts has a mean of 25.5 and a median of 25.5. The equal mean and median of attempts is most likely due to there being exactly 10 attempts for test A and B daily for five days for all 25 participants.

```
## Attempts :  int [1:2500] 1 2 3 4 5 6 7 8 9 10 ...
## Duration :  num [1:2500] 11.58 10.76 13.42 9.28 11.59 ...
## Errors :  num [1:2500] 1 0 0 0 2 0 0 3 9 0 ...
## Name :  Ord.factor w/ 25 levels "SUB22"<"SUB14"<..: 19 19 19 19 19 19 19 19 19 19 ...
## Type :  Factor w/ 2 levels "A","B": 1 1 1 1 1 1 1 1 1 1 ...
```

```
## [1] 1.6744
```

```
## [1] 6.109228
```

By using the ls.str() command, the structure of the Errors data frame can be observed. This will help in knowing how to input the different variables when using different functions. By looking tat the structure, it can be seen that the Attempts variable is a integer and Type is a factor with two levels, while Duration and Errors are both numbers.The mean and variable functions show that the mean of the Error variable is 1.6744 and its variance is 6.109228. The variance is much greater than the mean, which suggests that there will be over-dispersion within the model.

In figure 1, the histogram showcases the frequency of errors for the range of errors committed by the individuals. By looking at the histogram it clearly looks that the data does not follow a normal distribution. The data is skewed to the right.

The compound residual plot in figure 2 shows that the variance is not constant and that the data is heteroskedastic in nature. This is evident from the dispersion of the residuals viewed in both the the Scale location plot and the Residual vs Fitted plot. Most of the data points are concentrated towards the left which shows that variance is not constant. The QQ plot shows that the points approximately fall on the line and therefore the data most likely follows a normal distribution However, there is a slight non-normality.On the right hand side of the graph, the points lie slightly above the line which suggests a 'fat tail' on the right hand side of the distribution. The scale location plot also sjows that the average magnitude of the standardized

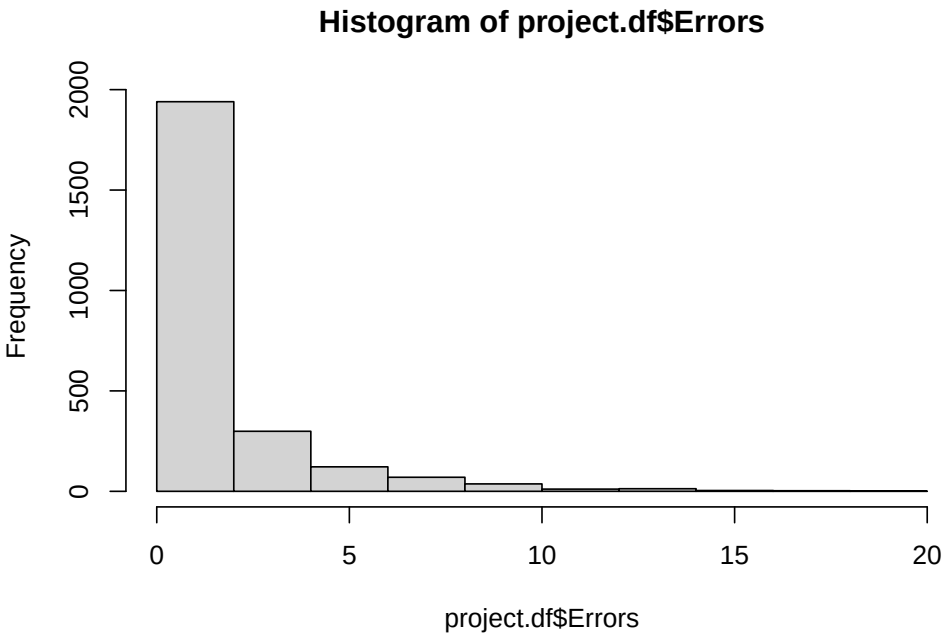

Figure 1: Histogram of the dependent variable Error's data continuity

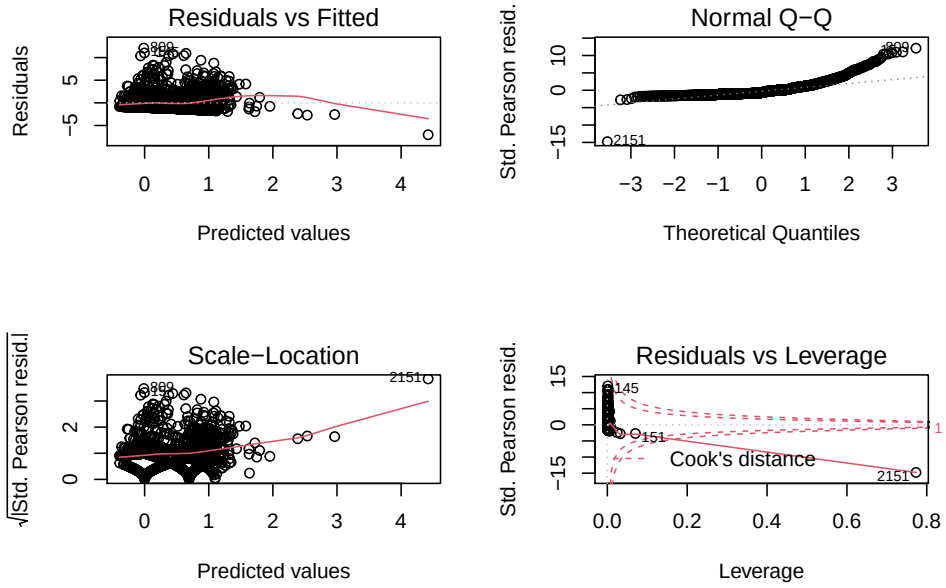

Figure 2: Compound residual plot

residuals increases with the predicted values. The Residuals vs Leverage plot showcases a point that lies outside the cooks distance, this shows that the data may have high inlfuence points. The spread of pearson standardised residuals also suggest that the data may be non linear in nature.

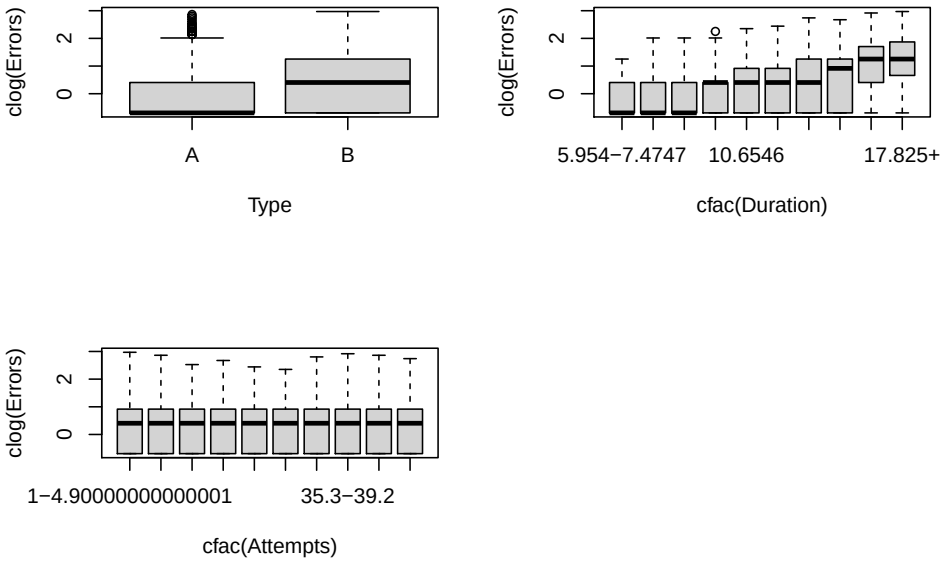

Figure 3: Sideby-side box plots for the dependent variable(Errors) against the regressors

The boxplots in figure 3 show that the average number of errors is higher for test type B than for A. It also shows that B has a higher variation in the amount of errors that occur during the test. Test A seems to be more skewed to the right and features outliers. The box plot for Duration against Errors shows that amount of errors that occur seems to increase as the duration of the test extends. The boxplot for Attempts against Errors shows that the amount of erros seem to be unchanged with the number of attempts performed. The dependent variable has been configured using a convenience function which makes the variable a continuity-corrected logarithm.

```
##
## Call:
## glm(formula = Errors ~ Type + Attempts + Duration, family = poisson(),
##     data = project.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.5059  -1.4526  -0.7177   0.4510   6.6487
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.812003   0.047868  -16.96   <2e-16 ***
## TypeB        0.551150   0.034909   15.79   <2e-16 ***
## Attempts     0.013295   0.001131   11.76   <2e-16 ***
## Duration     0.047234   0.001283   36.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 7317.4  on 2499  degrees of freedom
## Residual deviance: 5968.8  on 2496  degrees of freedom
```

```
## AIC: 9841.4
##
## Number of Fisher Scoring iterations: 6
```

Since Errors is a count variable and we want to see possible factors that influence the number of errors made during the test, the poisson regression will be used as it is a reasonable way to model data with counts. A summary of the poisson regression featuring the independent variables Duration, Attempts, and Type for the dependent variable errors, shows that all the regressors are statistically significant meaning that they are all possible factors that influence the number of errors made during the test. They are all significant at the 5 percent significance level with p-values below 2e-16. However, it can be seen that the residual deviance and degrees of freedom are not equal. The Residual deviance is much greater than the degrees of freedom. This suggests that there is the presence of over dispersion within the model and therefore the standard errors are incorrect and unaccounted for by the model. To have more accurate standard errors, the quasi-poisson model should be used instead.

```
##
## Call:
## glm(formula = Errors ~ Type + Attempts + Duration, family = quasipoisson(),
##     data = project.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.5059  -1.4526  -0.7177   0.4510   6.6487
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.812003   0.080260 -10.117  < 2e-16 ***
## TypeB        0.551150   0.058532   9.416  < 2e-16 ***
## Attempts     0.013295   0.001896   7.013 2.98e-12 ***
## Duration     0.047234   0.002151  21.961  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 2.811252)
##
##     Null deviance: 7317.4  on 2499  degrees of freedom
## Residual deviance: 5968.8  on 2496  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
```

The quasi poisson regression model shows that the coefficients of the regressors are the same as the coefficients within the poisson model. However, the standard errors are different. The standard error for TypeB has gone from 0.034909 to 0.058532.The standard error for Attempts has gone from 0.001131 to 0.001896. The standard error for Duration has gone from 0.001283 to.0.002151

```
## [1] -73.52474
```

```
## [1] -1.338377
```

```
## [1] -4.83673
```

Through using the coefficients, the effect of the regressors on errors can be observed. As can be seen above, going from test type A to B results in a 73.52474 percent decrease in errors. A unit increase in the number of attempts results in a 1.338377 percent decrease in errors, and a unit increase in duration results in a 4.83673 percent decrease in errors.

Figure 4, shows a visual way to get information regarding the regression model. It does this by plotting the

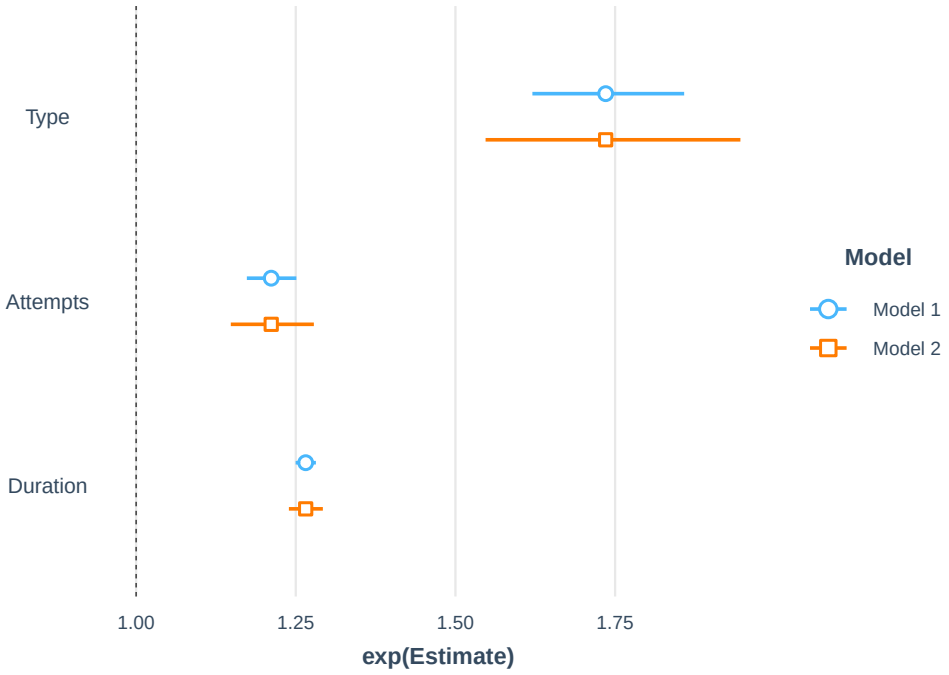

Figure 4: Regression coefficient plot

information on the values of the coefficients of the regression model and their corresponding uncertainties. Figure 4 has performed this using the coefficients of the quasi poisson regression. This makes information easier to interpret and view. It can visually be seen that, the quasi poisson model has higher coefficient estimates at the exponential level compared to the quasi model.

```
## Warning: Attempts and Duration are not included in an interaction with one another
## in the model.
```

```
## Warning: Duration and Attempts are not included in an interaction with one another
## in the model.
```

Figures 5 and 6 show case the scatter plots of the errors data frame. They also showcase the quasi poisson regression. As can be seen, it visually evident that the data frame does not follow a linear pattern, and is rather non linear in nature. This makes it more evident that the non linear poisson regression is a better model to be used in the analysis of the errors data.

Conclusion

In general, the errors for test B seem to be higher on average compared to test A. The number of errors also seem to decrease with added duration and attempts. The data in the model is evidently nonlinear in nature, and the count attribute of the errors make the poisson regression a usefull model. However, due to the large difference between the residual deviance and the degrees of freedom, the quasi poisson regression proved to be a a better model fit in terms of providing more accurate standard errors.The statistical significance within the regressors shows that they are all possible factors that influence the number of errors made during the test. The fact that test type can impact the number of errors performed on a neurological test shows that the test taker should take the results with a grain of salt when evaluating their cognitive capabilities. Quality control in trail making apps is recommended in order to make sure that the results of neurological test are more accurate and consistent.

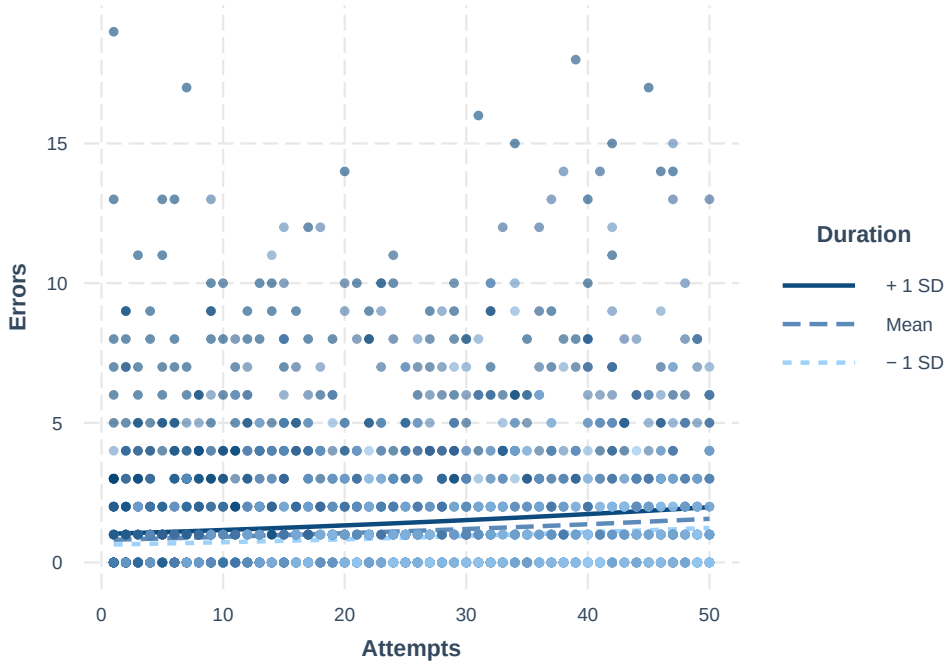

Figure 5: Interaction plot

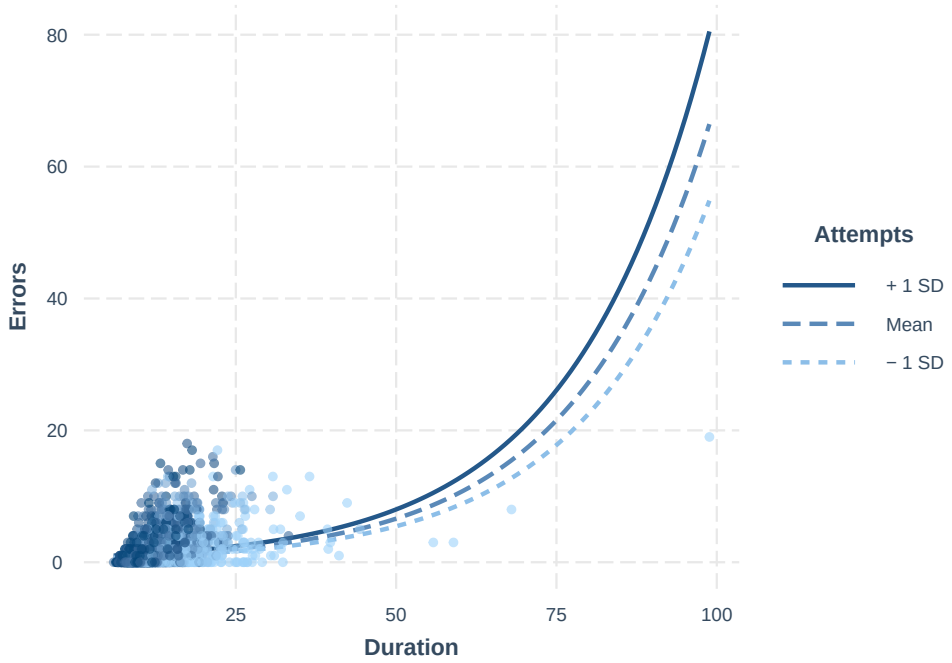

Figure 6: Interaction plot

Appendices

```
knitr::opts_chunk$set(echo = TRUE)
library(gridExtra)
library(dplyr)
library(coin)

source("/Users/ryotanishida/R/Errors.R")

project.df$Type<- as.factor(project.df$Type)

summary(project.df)
```

```
##       Name      Type         Errors          Duration         Attempts
##  SUB22  : 100   A:1250   Min.   : 0.000   Min.   : 5.954   Min.   : 1.0
##  SUB14  : 100   B:1250   1st Qu.: 0.000   1st Qu.: 9.640   1st Qu.:13.0
##  SUB20  : 100            Median : 1.000   Median :11.367   Median :25.5
##  SUB15  : 100            Mean   : 1.674   Mean   :12.604   Mean   :25.5
##  SUB19  : 100            3rd Qu.: 2.000   3rd Qu.:14.135   3rd Qu.:38.0
##  SUB06  : 100            Max.   :19.000   Max.   :98.855   Max.   :50.0
##  (Other):1900
```

```
ls.str(project.df)
```

```
## Attempts :  int [1:2500] 1 2 3 4 5 6 7 8 9 10 ...
## Duration :  num [1:2500] 11.58 10.76 13.42 9.28 11.59 ...
## Errors :  num [1:2500] 1 0 0 0 2 0 0 3 9 0 ...
## Name :  Ord.factor w/ 25 levels "SUB22"<"SUB14"<..: 19 19 19 19 19 19 19 19 19 19 ...
## Type :  Factor w/ 2 levels "A","B": 1 1 1 1 1 1 1 1 1 1 ...
```

```
mean(project.df$Errors)
```

```
## [1] 1.6744
```

```
var(project.df$Errors)
```

```
## [1] 6.109228
```

```
hist(project.df$Errors)
```

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

```
par(mfrow=c(1,1))

cfac <- function(x, breaks = NULL) { if(is.null(breaks)) breaks <- unique(quantile(x, 0:10/10))
x <- cut(x, breaks, include.lowest = TRUE, right = FALSE)
 levels(x) <- paste(breaks[-length(breaks)], ifelse(diff(breaks) > 1,
 c(paste("-", breaks[-c(1, length(breaks))] - 1, sep = ""), "+"), ""),
 sep = "")
 return(x)
}

clog <- function(x) log(x + 0.5)

  par(mfrow=c(2,2))
  plot(clog(Errors) ~ Type, data = project.df, varwidth = TRUE)
```

```
 plot(clog(Errors) ~ cfac(Duration), data = project.df)
 plot(clog(Errors) ~ cfac(Attempts), data = project.df, varwidth = TRUE)

 trailmaking <- glm(Errors ~ Type + Attempts+Duration, data = project.df,family = poisson())
summary(trailmaking)
```

```
##
## Call:
## glm(formula = Errors ~ Type + Attempts + Duration, family = poisson(),
##     data = project.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.5059  -1.4526  -0.7177   0.4510   6.6487
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.812003   0.047868  -16.96   <2e-16 ***
## TypeB        0.551150   0.034909   15.79   <2e-16 ***
## Attempts     0.013295   0.001131   11.76   <2e-16 ***
## Duration     0.047234   0.001283   36.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 7317.4  on 2499  degrees of freedom
## Residual deviance: 5968.8  on 2496  degrees of freedom
## AIC: 9841.4
##
## Number of Fisher Scoring iterations: 6
```

```
trailmaking2 <- glm(Errors ~ Type+Attempts+Duration, data = project.df,family = quasipoisson())
summary(trailmaking2)
```

```
##
## Call:
## glm(formula = Errors ~ Type + Attempts + Duration, family = quasipoisson(),
##     data = project.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.5059  -1.4526  -0.7177   0.4510   6.6487
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.812003   0.080260 -10.117  < 2e-16 ***
## TypeB        0.551150   0.058532   9.416  < 2e-16 ***
## Attempts     0.013295   0.001896   7.013 2.98e-12 ***
## Duration     0.047234   0.002151  21.961  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 2.811252)
```

```
##
##     Null deviance: 7317.4  on 2499  degrees of freedom
## Residual deviance: 5968.8  on 2496  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
```

```
pType<-(1-exp(0.551150))*100
pType
```

```
## [1] -73.52474
```

```
pAttempts<-(1-exp(0.013295))*100
pAttempts
```

```
## [1] -1.338377
```

```
pDuration<-(1-exp(0.047234))*100
pDuration
```

```
## [1] -4.83673
```

```
library(jtools)
library(ggstance)
library(broom)
library(broom.mixed)
library(interactions)
plot_summs(trailmaking,trailmaking2, scale = TRUE, exp = TRUE)
```

```
## Note: Pseudo-R2 for quasibinomial/quasipoisson families is calculated by
## refitting the fitted and null models as binomial/poisson.
```

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

```
interact_plot(trailmaking2, pred = Attempts, modx = Duration,plot.points = TRUE)
```

```
## Warning: Attempts and Duration are not included in an interaction with one another
## in the model.
```

```
par(mfrow=c(2,2))
interact_plot(trailmaking2, pred = Duration, modx = Attempts,plot.points = TRUE)
```

```
## Warning: Duration and Attempts are not included in an interaction with one another
## in the model.
```

References

[1] Hothorn, T and Everitt, B. (2014) *A Handbook of Statistical Analyses Using R* Florida: CRC Press.