EDA.Rmd

EDA Autism spectrum discorder quiz

Wouter Zeevat

Contents

Introduction

The AQ-10 test is a test with 10 questions that will try and predict if you have the autism spectrum disorder. Each question has 4 anwsers. Slightly agree, Agree, Slightly disagree and disagree. Every question gives either 1 or 0 points based on the anwser. Slightly agree gives the same points as agree and the same for disagree. They made it like this to make it feel like slightly agree/disagree feels like a better anwser when in doubt. The more points the people have, the more chance there is of them having ADS (Autism disorder spectrum).

The data: https://www.kaggle.com/datasets/faizunnabi/autism-screening

Research question

How accurate can the AQ-10 test predict whether someone has the autism spectrum disorder? The goal of this research question is to find out if this autism spectrum disorder test actually works and predicts someone has it. This would involve machine learning by testing if the computer would find correlations and would be able to predict them actually having ASD, which the test does aswell. This way the accuracy of the test can be measured.

We will start off by looking at the data and codebook. The data contains 21 variables which will be loaded in as the codebook.

Variable.name.original	Variable.name.human.readable	description	type	unit
a1_score	Anwser question 1 score	The first anwser of the Q10 test	numeric 1 or 0	score
a2_score	Anwser question 2 score	The second anwser of the Q10 test	numeric 1 or 0	score
a3_score	Anwser question 3 score	The third anwser of the Q10 test	numeric 1 or 0	score
a4_score	Anwser question 4 score	The fourth anwser of the Q10 test.	numeric 1 or 0	score
a5_score	Anwser question 5 score	The fifth anwser of the Q10 test	numeric 1 or 0	score
a6_score	Anwser question 6 score	The sixth anwser of the Q10 test	numeric 1 or 0	score
a7_score	Anwser question 7 score	The seventh anwser of the Q10 test	numeric 1 or 0	score
a8_score	Anwser question 8 score	The eighth anwser of the Q10 test	numeric 1 or 0	score
a9_score	Anwser question 9 score	The ningth anwser of the Q10 test	numeric 1 or 0	score
a10_score	Anwser question 10 score	The tenth anwser of the Q10 test	numeric 1 or 0	score
age	age	The age of the corresponding person	numeric	years
gender	gender	The gender of the corresponding person	nominal	male or female
ethnicity	ethnicity	the ethnicity of the corresponding person	nominal	type of ethnicity
jaundice	jaundice	Does the person have jaundice.This is sometimes linked with ASD	boolean	yes or no
autism	autism	Does the person actually have autism	boolean	yes or no
country_of_r	country of residence	The country where the corresponding person lives	nominal	country name

Variable.name.original	Variable.name.human.readable	description	type	unit
used_app_before	used the app before	Did the corresponding person used the app that is used to take the Q10 test before	boolean	yes or no
end_score	final test score	The score of all 10 questions added up to eachother	numeric	0-10 score
age_desc	age descending	The age of the person in a string. Usually 18 older	nominal factor	years
relation	relationship	relation user compared to person of interest	nominal	string of relationship
class_asd	classified ASD	Does the test classify the person as having ASD	boolean	yes or no

The data

This is the data that will be used in the following project. it contains various information about adults doing an autism test. The columns speak for themselves except for the first 10. These columns represent the anwsers of the following question list.

https://www.nice.org.uk/guidance/cg142/resources/autism-spectrum-quotient-aq10-test-pdf-186582493

After knowing all of this it's time to see if the data is right. The data is supposed to have 21 columns and 704 rows.

```
## [1] 21
```

```
## [1] 704
```

Checking the data

The data also needs to be checked of missing data (A row that's missing certain values). The ones that are missing important data will be removed. This needs to be done in order to not mess everything up. For example if someone is missing an anwser of the quiz, their score will be messed up and invalid.

This code will check if there are invalid values in any column.

```
data[data == "?"] <- NA
data[1:11] <- data[1:11] %>% mutate_if(is.character,as.numeric) # Changing strings that are numbers to

# Changing the 0,1 to NO and YES. The no means they dont get a point and the yes means they did.
data[1:10][data[1:10] == 0] <- "NO"
data[1:10][data[1:10] == 1] <- "YES"

summary(data)
```

```
##    a1_score           a2_score           a3_score           a4_score
##  Length:704         Length:704         Length:704         Length:704
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
```

```
##    a5_score           a6_score           a7_score           a8_score
##  Length:704         Length:704         Length:704         Length:704
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    a9_score          a10_score             age           gender
##  Length:704         Length:704         Min.   : 17.0   Length:704
##  Class :character   Class :character   1st Qu.: 21.0   Class :character
##  Mode  :character   Mode  :character   Median : 27.0   Mode  :character
##                                        Mean   : 29.7
##                                        3rd Qu.: 35.0
##                                        Max.   :383.0
##                                        NA's   :2
##   ethnicity           jaundice           autism          country_of_r
##  Length:704         Length:704         Length:704         Length:704
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  used_app_before      end_score        age_desc           relation
##  Length:704         Min.   : 0.000   Length:704         Length:704
##  Class :character   1st Qu.: 3.000   Class :character   Class :character
##  Mode  :character   Median : 4.000   Mode  :character   Mode  :character
##                     Mean   : 4.875
##                     3rd Qu.: 7.000
##                     Max.   :10.000
##
##   class_asd
##  Length:704
##  Class :character
##  Mode  :character
##
##
##
##
```

There are two NA's in the data. It is important to remove those in order to keep the data balanced. This will be done by removing their rows.

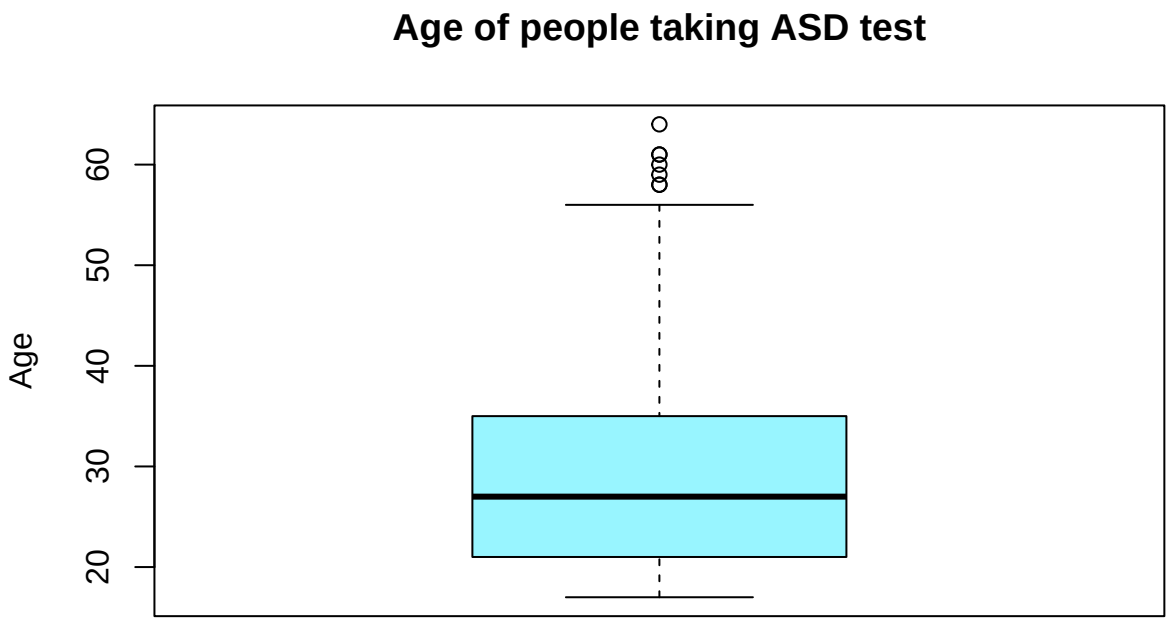
```
data <- data[-c(which(is.na(data$age))), ]
```

We will now take a look at the ages of the people taking the test are.

```
boxplot(data$age, main="Age of people taking ASD test", ylab="Age", col="cadetblue1")
```

Figure 1A, As the boxplot shows, there's one huge outlier. One person would be 383 years old which just isn't humanly possible. The solution to this is taking out the whole row.

```
data <- data[-c(which(data$age == 383)),]
boxplot(data$age, main="Age of people taking ASD test", ylab="Age", col="cadetblue1")
```




```
boxplot(log10(data$age), main="Age of people taking ASD test", ylab="10log Age", col="cadetblue1")
```

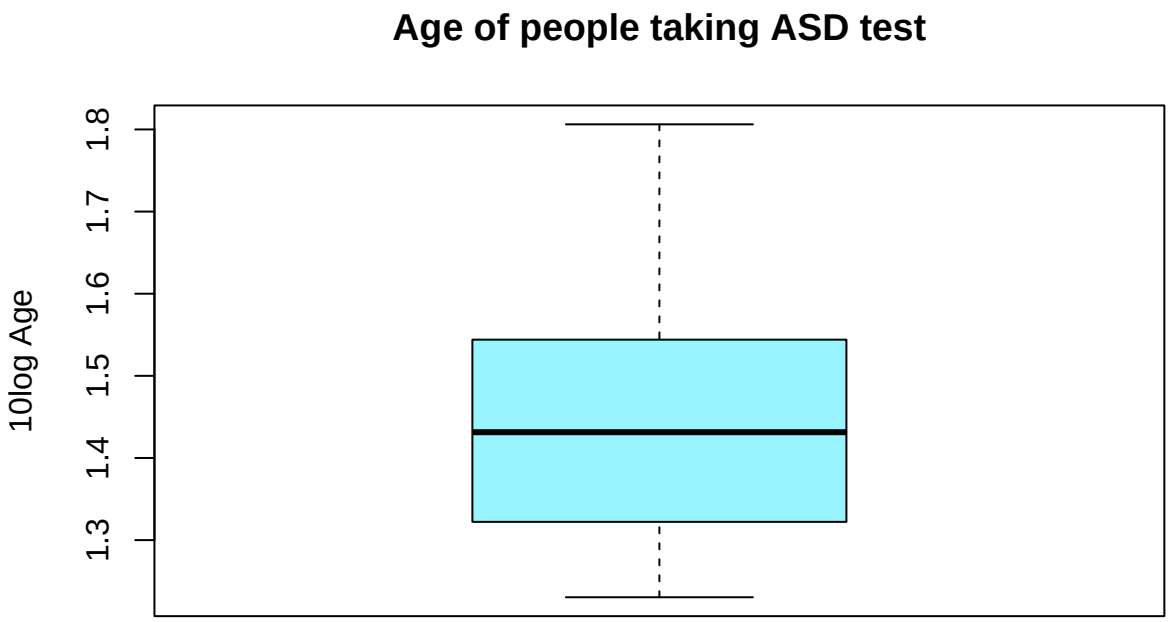


Figure 1B/C, This is the fixed plot by using the 10log function. It shows that the majority of the people taking the test is Mid age. This boxplot shows that there's not much old people (60+) doing the test. The people who take this test are usually mid aged.

Now we will take a look at the test, how much people had what kind of anwser. The goal of this plot is to take a look at what the people scored.

```
df <- data.frame(x = rep(paste0("Q", c("01", "02", "03", "04", "05", "06", "07", "08", "09", "10")), ea
                 y = rep(0, each=20),
                 group = rep(c("YES", "NO"), time = 10))

df$y[1] <- sum(data$a1_score == "YES")
df$y[2] <- sum(data$a1_score == "NO")

df$y[3] <- sum(data$a2_score == "YES")
df$y[4] <- sum(data$a2_score == "NO")

df$y[5] <- sum(data$a3_score == "YES")
df$y[6] <- sum(data$a3_score == "NO")

df$y[7] <- sum(data$a4_score == "YES")
df$y[8] <- sum(data$a4_score == "NO")

df$y[9] <- sum(data$a5_score == "YES")
df$y[10] <- sum(data$a5_score == "NO")

df$y[11] <- sum(data$a6_score == "YES")
df$y[12] <- sum(data$a6_score == "NO")

df$y[13] <- sum(data$a7_score == "YES")
df$y[14] <- sum(data$a7_score == "NO")

df$y[15] <- sum(data$a8_score == "YES")
df$y[16] <- sum(data$a8_score == "NO")

df$y[17] <- sum(data$a9_score == "YES")
df$y[18] <- sum(data$a9_score == "NO")

df$y[19] <- sum(data$a10_score == "YES")
df$y[20] <- sum(data$a10_score == "NO")

ggp <- ggplot(df, aes(x = x, y = y, fill = group, label = y)) +  # Create stacked bar chart
  geom_bar(stat = "identity") + ggtitle("Q10 test Anwsers") + xlab("Question number") + ylab("Amount of
ggp
```

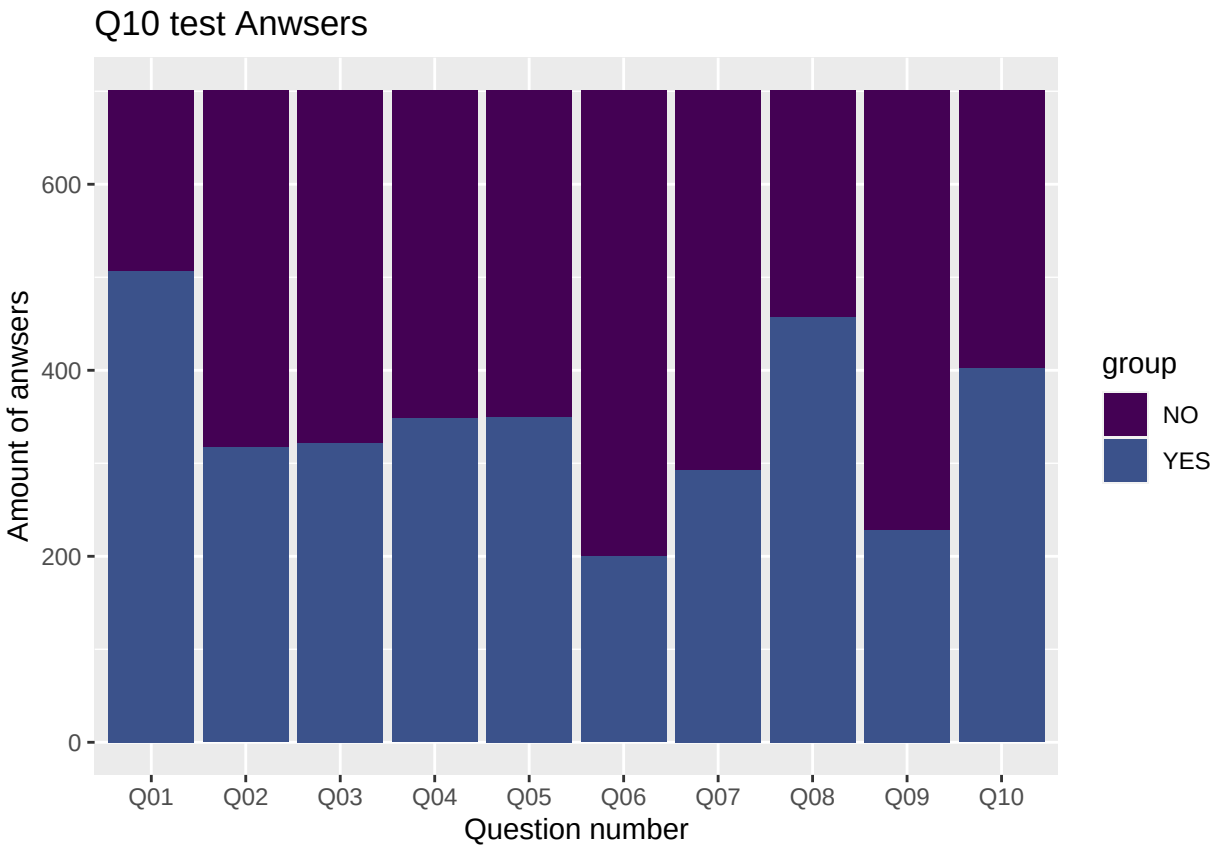


Figure 2, The conclusion of this plot is that most questions are anwsered positively (Without getting a point).

Correlations

Let's take a look at the correlations now. To start off, the end score will be measured against people actually having ASD. This will give a good view of the test because the test results will directly be compared to them having ASD.

```
boxplot(data$end_score ~ data$autism, col=c("cadetblue4", "cadetblue3"), xlab="Has ASD", ylab="Q10 end s
```

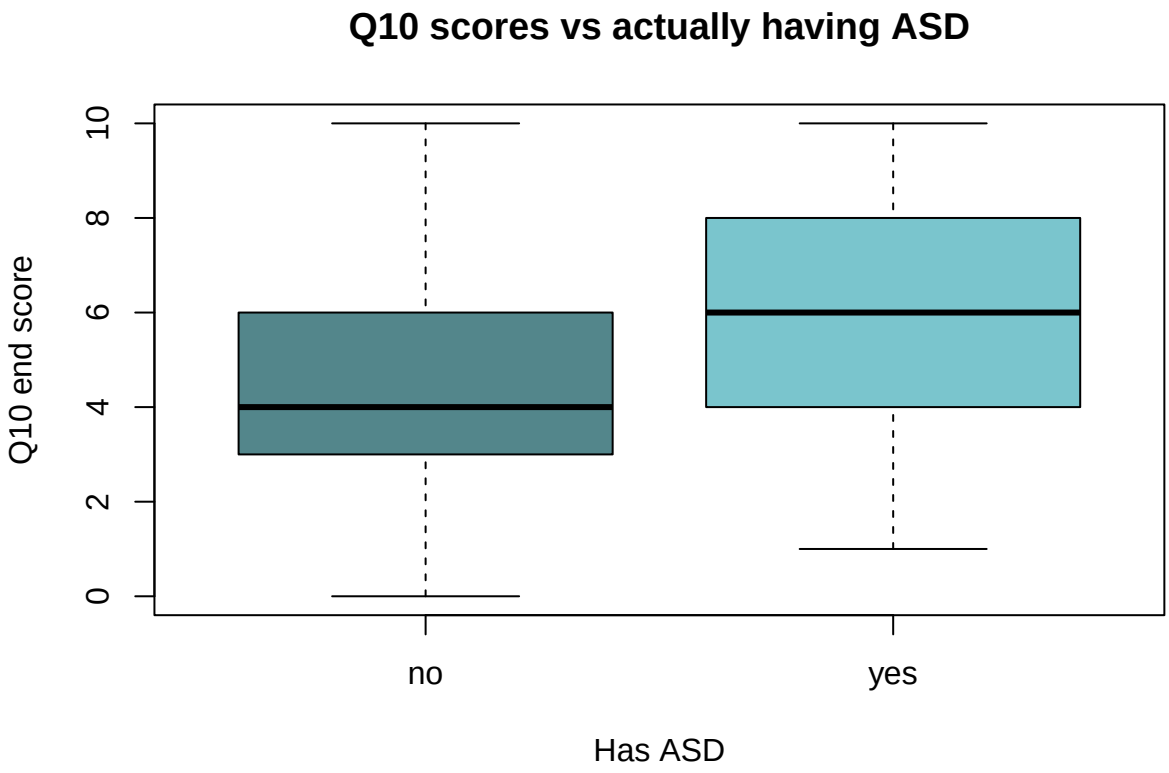


Figure 3, As seen in the plot, the scores do actually correlate with someone having ASD. This is true because the scores of the people having ASD are significantly higher than the other people. Let's confirm this by doing a t-test

```
t.test(data$end_score, subset=data$autism, var.equal = TRUE)
```

```
##
##  One Sample t-test
##
## data:  data$end_score
## t = 51.867, df = 700, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  4.703675 5.073785
## sample estimates:
## mean of x
##   4.88873
```

```r
df <- data.frame(x = rep(1, each=4),
                 y = rep(0, each=4),
                 group = c("ASD and Family member with ASD", "ASD and no Family member with ASD", "no

df$y <- c(sum(data$class_asd == "YES" & data$autism == "yes"), sum(data$class_asd == "NO" & data$autism
          sum(data$class_asd == "YES" & data$autism == "no"), sum(data$class_asd == "NO" & data$autism =

ggp <- ggplot(df, aes(x = x, y = y, fill = group, label = y)) +  # Create stacked bar chart
  geom_bar(stat = "identity") + ggtitle("ASD and family member with ASD") + xlab("") + ylab("amount of p
ggp
```

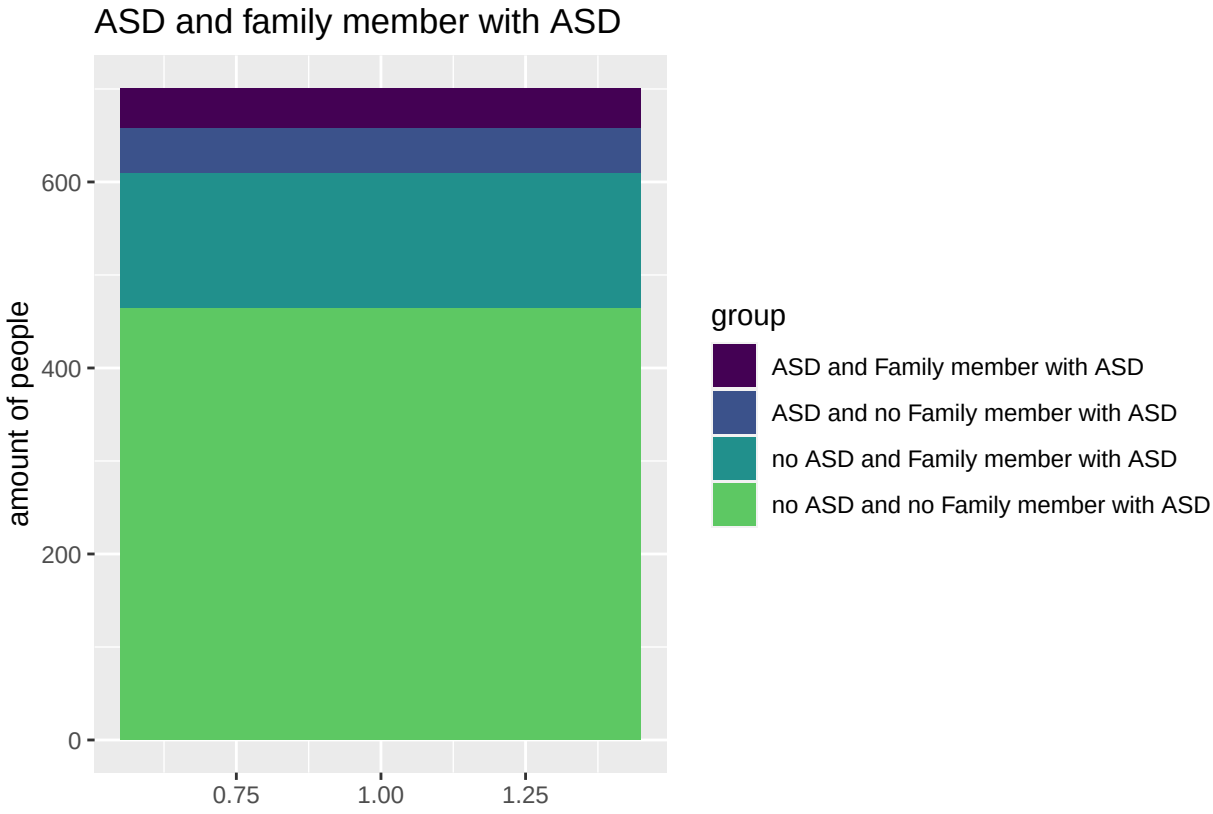


Figure 4, This figure shows that Most people that take the test do not have ASD and no family member with ASD. There's also a big part of the group that do not have ASD but have family members with ASD. This makes sense because these people probably want to take the test because they know they have a family member with ASD and are scared they have it too.

```
grouped <- data %>%
  mutate(positives = case_when(
    autism == "yes" ~ "ASD",
    autism == "no" ~ "no ASD"
  ))
ggplot(grouped, aes(x = end_score, y = age, colour = positives)) +
  geom_point() + geom_jitter() + ggtitle("End score compared to age") + geom_smooth(method="lm")
```

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

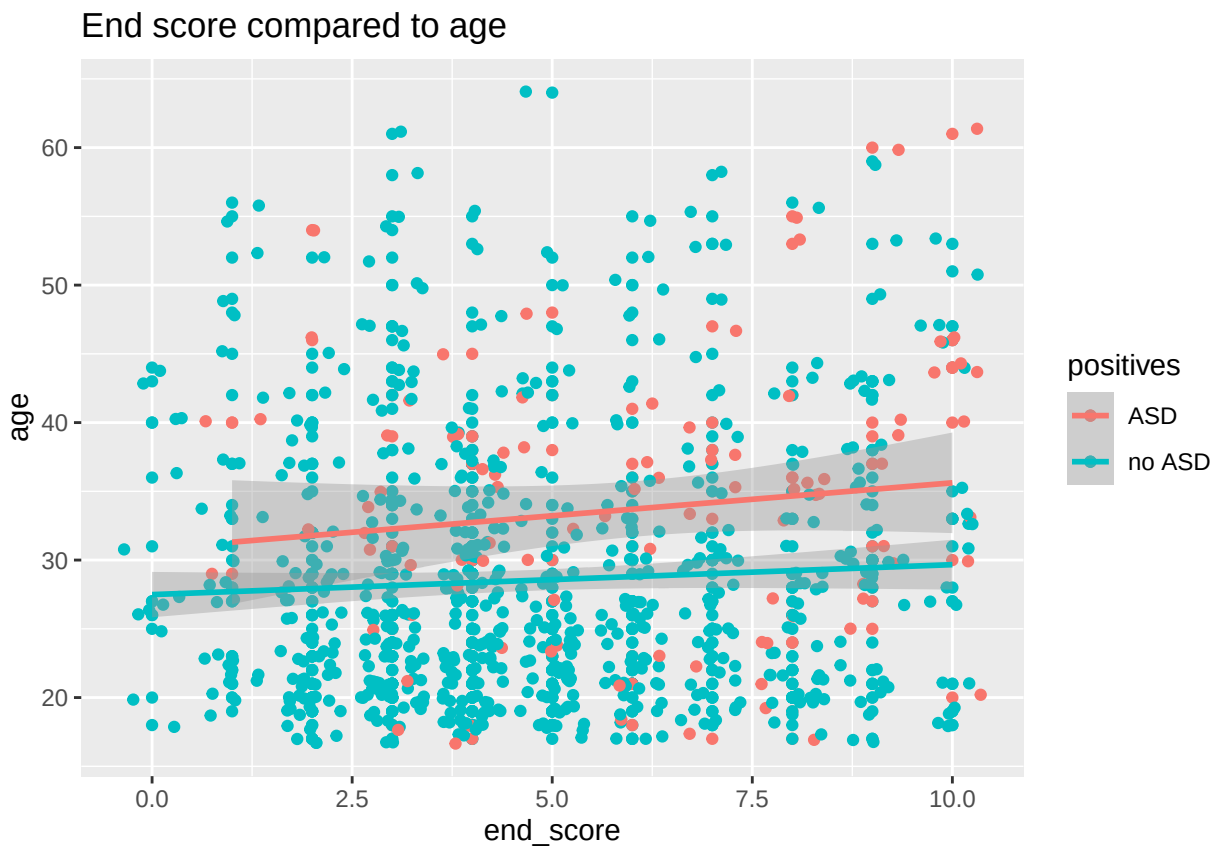


Figure 5, Some correlation that would not be good would be the correlation of age and the end score. Something that could be possible is that the older you are, the worse you make the test. This would be bad because then the AQ-10 test would be based on age and not on people actually having ASD. It does not look like this is the case though. Let's confirm this by doing a correlation test!

```
cor.test(data$end_score, data$age, method="pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$end_score and data$age
## t = 2.5981, df = 699, p-value = 0.009571
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.02392249 0.17061353
```

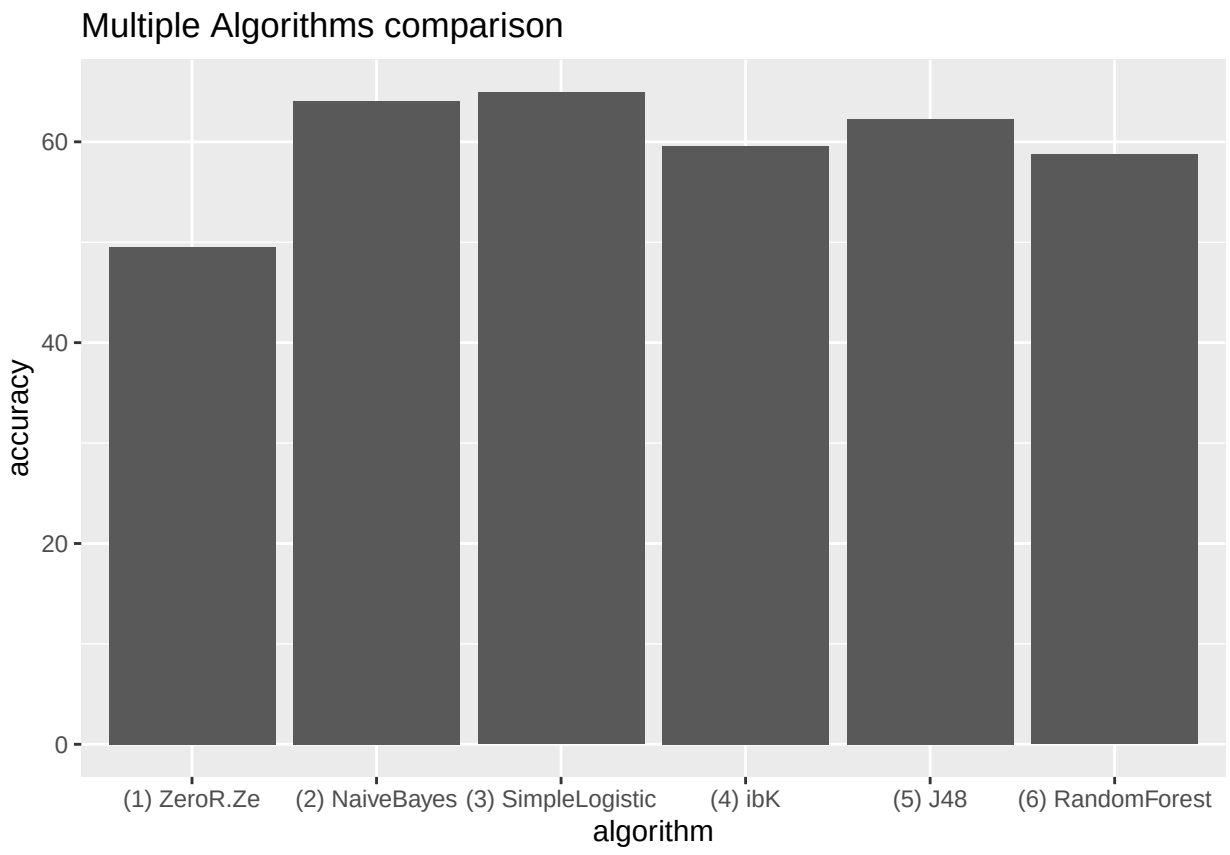
```
## sample estimates:
##        cor
## 0.09779918
```

Supervised Learning

There needs to be a algorithm made in Weka. After entering the data and removing the unused parts (Country of birth, Country of residence, Used the app before) multiple algorithms were tested. The function algorithm "SimpleLogistic" is chosen. This one fits the most because the different variables do not have a correlation therefore a tree or decision algorithm would not work. This will be confirmed by comparing it to different algorithms!

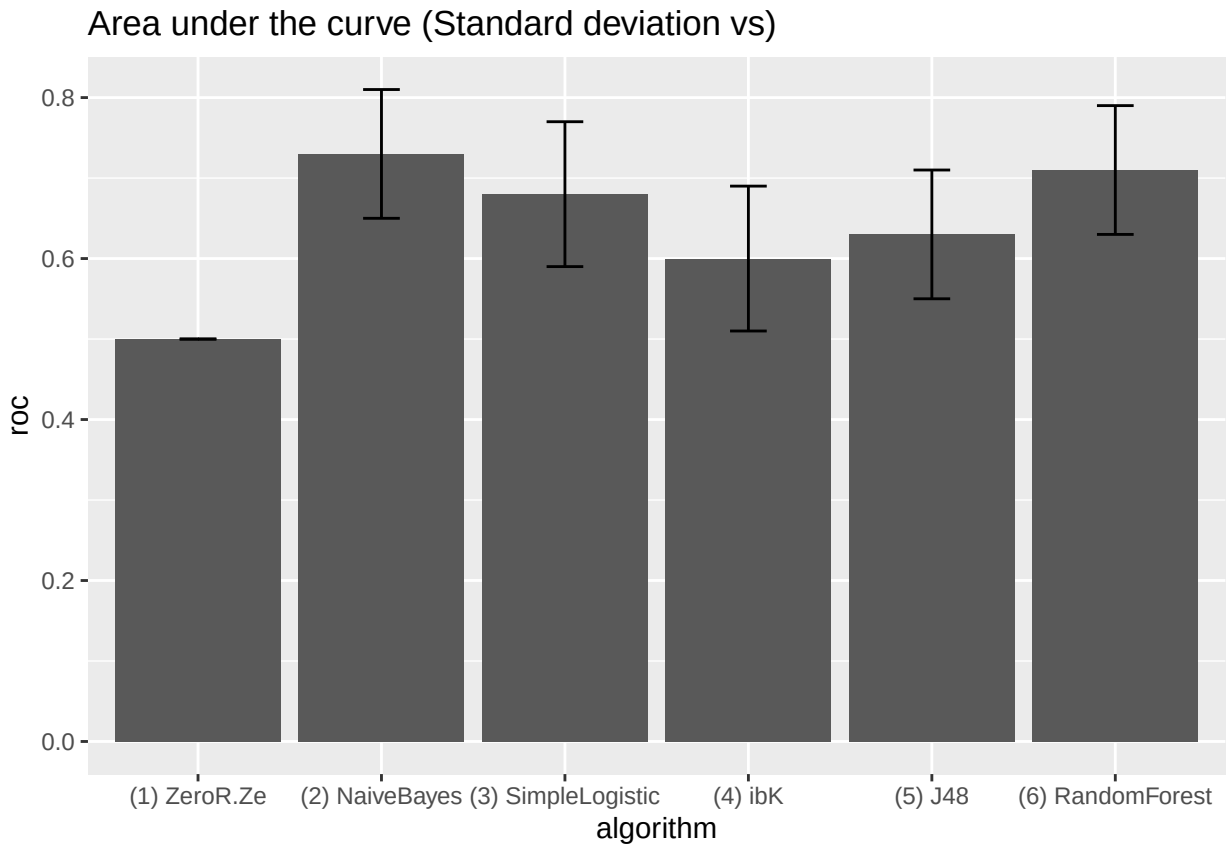
After having tested this algorithm the results were dissapointing. Even though the % predicted right was a high 75%, the algorithm almost always predicted that people didn't have autism. Which seems good because not a lot of people have it. But the goal is to know when people have ASD. After concluding this i realized that the data was imbalanced. This was fixed by using a ClassBalancer supervised filter. Now the ASD values are 350.5 350.5 instead of 507 194. Now that the algorithm is correct some statistical tests will be applied.

```
algorithms <- read.csv("algorithms.csv")

ggp <- ggplot(data=algorithms, aes(x=algorithm, y=accuracy), label = y, fill=group) +
  geom_col(position = "dodge", show.legend = FALSE) +
  ggtitle("Multiple Algorithms comparison")

ggp
```



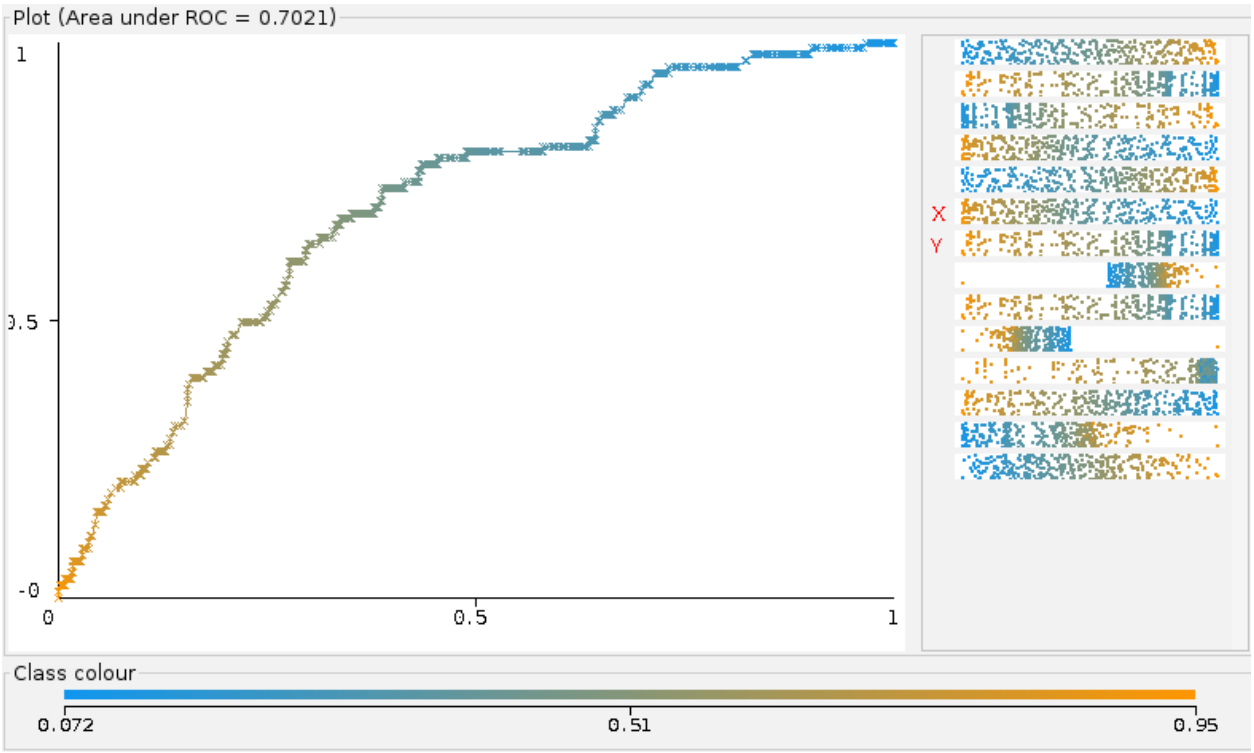
The algorithm with the highest score here is SimpleLogistic. It's slightly higher than then NaiveBayes which is the second option.This shows that Simplelogistic was indeed the best option for accuracy. Now the ROC will be tested. If SimpleLogistic is really low here maybe NaiveBayes should be used instead.

```
ggp <- ggplot(data=algorithms, aes(x=algorithm, y=roc), label = y, fill=group, ylim=1.5) +
  geom_col(position = "dodge", show.legend = FALSE) +
  ggtitle("Area under the curve (Standard deviation vs)") +
  geom_errorbar(aes(ymin = roc-error, ymax= roc+error), width = 0.2)
ggp
```



This graph shows that NaiveBayes has the highest area under the curve. All of them are significant to ZeroR though. The error rate shows that naivebayes could also have the same ROC as SimpleLogistic. Which is why SimpleLogistic is still the chosen classifier!

Threshold curve



This is the Threshold curve of simplelogistic. It has a nice shape showing the ROC is quite good.

After all of this is done A java wrapper will be made in order to use the balanced simplelogistic classifier.