Austism_screening

Chi Ting Low

6/29/2018

```r
library(mlr) #machine learning
```

```
## Loading required package: ParamHelpers
```

```r
library(foreign) #reading arff file
library(Amelia) #checking missing values
```

```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.5, built: 2018-05-07)
## ## Copyright (C) 2005-2018 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```r
library(tidyverse) #ggplots
```

```
## -- Attaching packages ------------------------------------------------------------------------ tidyve
```

```
## √ ggplot2 2.2.1     √ purrr   0.2.5
## √ tibble  1.4.2     √ dplyr   0.7.5
## √ tidyr   0.8.1     √ stringr 1.3.1
## √ readr   1.1.1     √ forcats 0.3.0
```

```
## -- Conflicts --------------------------------------------------------------------------- tidyverse_c
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(dplyr) #data manipulation
library(knitr) #for pretty table
library(PerformanceAnalytics) #for correlation
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Attaching package: 'xts'
```

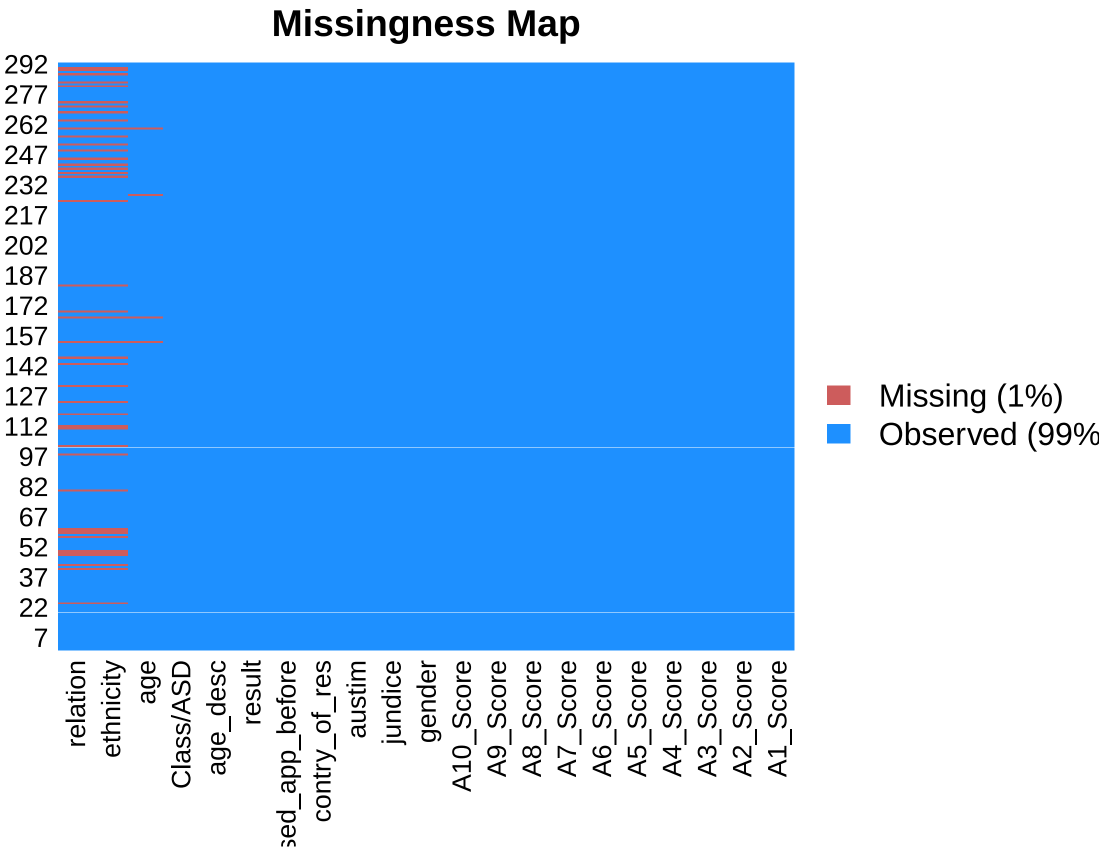
```
## The following objects are masked from 'package:dplyr':
##
##     first, last
```

```
##
## Attaching package: 'PerformanceAnalytics'
## The following object is masked from 'package:graphics':
##
##     legend
```

```
library(corrr) #for correlation ntework
library(rpart.plot)
```

```
## Loading required package: rpart
```

```
set.seed(1234) #reproducible research
#reading data
data <- read.arff('Autism-Child-Data.arff')

#plot missing values
missmap(data)
```



As presented in the result it shows that there is only 1% of the data is missing. Therefore, it will be removed.

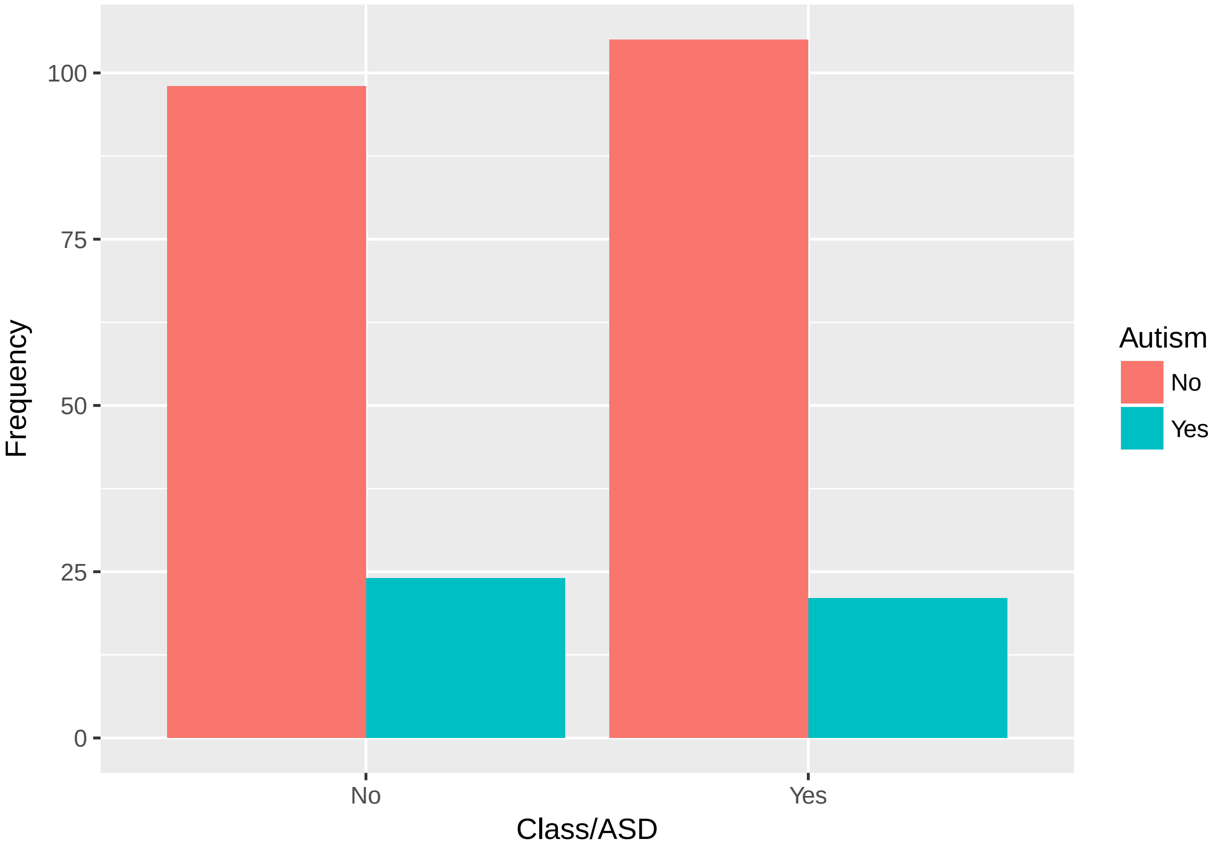
```
#remove missing values
data.na <- na.omit(data)

#rename variables
colnames(data.na) <- c("A1_Score", "A2_Score", "A3_Score", "A4_Score", "A5_Score", "A6_Score", "A7_Scor
```

```
#summarize of the data
kable(summarizeColumns(data.na))
```

name	type	na	mean	disp	median	mad	min	max	nlevs
A1_Score	factor	0	NA	0.3145161	NA	NA	78	170	2

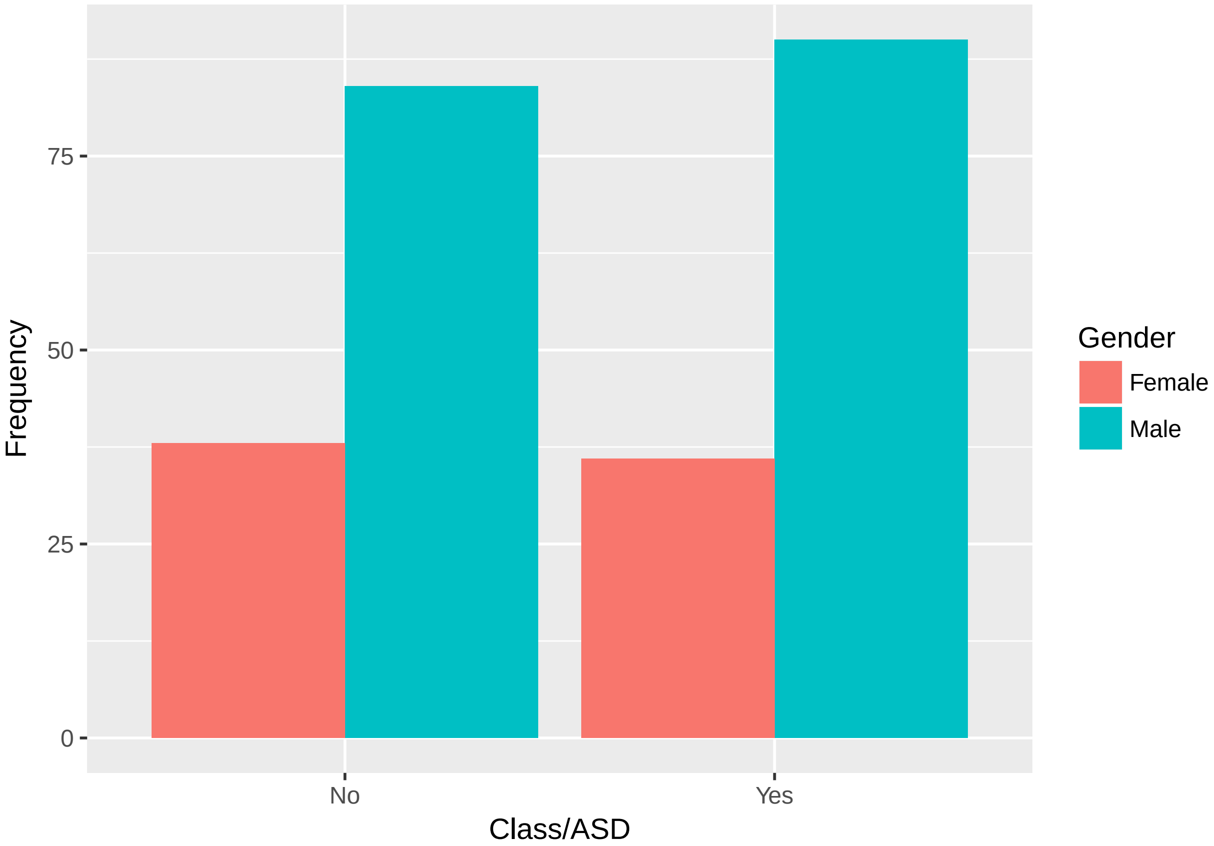
name	type	na	mean	disp	median	mad	min	max	nlevs
A2_Score	factor	0	NA	0.4838710	NA	NA	120	128	2
A3_Score	factor	0	NA	0.2540323	NA	NA	63	185	2
A4_Score	factor	0	NA	0.4274194	NA	NA	106	142	2
A5_Score	factor	0	NA	0.2459677	NA	NA	61	187	2
A6_Score	factor	0	NA	0.2862903	NA	NA	71	177	2
A7_Score	factor	0	NA	0.3750000	NA	NA	93	155	2
A8_Score	factor	0	NA	0.4798387	NA	NA	119	129	2
A9_Score	factor	0	NA	0.4596774	NA	NA	114	134	2
A10_Score	factor	0	NA	0.2661290	NA	NA	66	182	2
age	numeric	0	6.427419	2.3864441	6	2.9652	4	11	0
gender	factor	0	NA	0.2983871	NA	NA	74	174	2
ethnicity	factor	0	NA	0.5645161	NA	NA	2	108	10
jundice	factor	0	NA	0.2459677	NA	NA	61	187	2
austim	factor	0	NA	0.1814516	NA	NA	45	203	2
contry_of_res	factor	0	NA	0.8024194	NA	NA	0	49	46
used_app_before	factor	0	NA	0.0241935	NA	NA	6	242	2
result	numeric	0	6.366936	2.3427110	7	2.9652	0	10	0
age_desc	factor	0	NA	0.0000000	NA	NA	248	248	1
relation	factor	0	NA	0.1411290	NA	NA	1	213	5
Class_ASD	factor	0	NA	0.4919355	NA	NA	122	126	2

```
#recode the varialbe
data.na$gender <- recode(data.na$gender, m = 'Male', f = 'Female')
data.na$Class_ASD <- recode(data.na$Class_ASD, YES = 'Yes', NO = 'No')
data.na$austim <- recode(data.na$austim, yes = 'Yes', no = 'No')
data.na$jundice <- recode(data.na$jundice, yes = 'Yes', no = 'No')

#plot the data for exploratory analysis
autism <- ggplot(data.na, aes(x = Class_ASD, fill = austim)) + geom_bar(stat = 'count', position = 'dodg
autism + labs(x = 'Class/ASD', y = 'Frequency') + guides(fill = guide_legend(title = "Autism"))
```



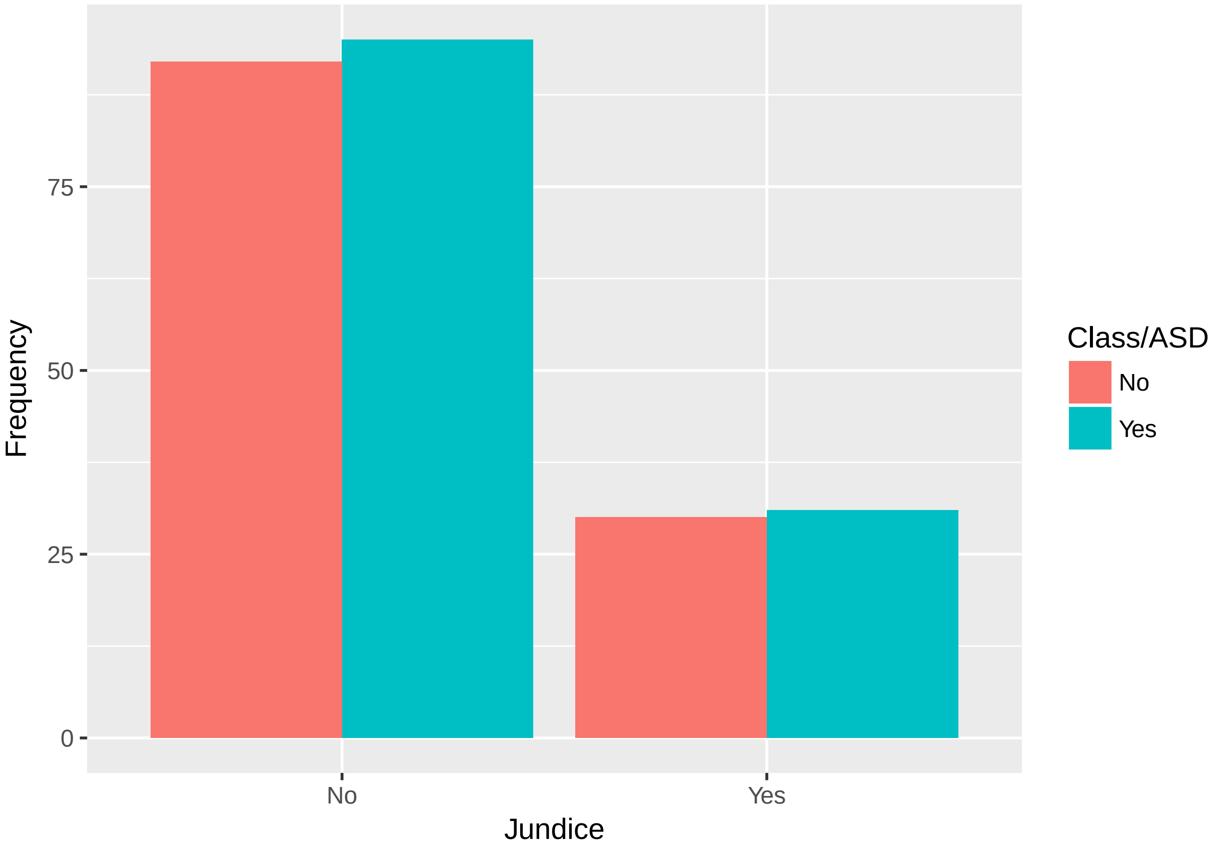
As seen in the graph, it shows that participants with no Autism is also evenly distributed as identified as ASD or not ASD

```
gender <- ggplot(data.na, aes(x = Class_ASD, fill = gender)) + geom_bar(stat = 'count', position = 'dodg
gender + labs(x = 'Class/ASD', y = 'Frequency') + guides(fill = guide_legend(title = "Gender"))
```



The graph shows that male are morel likely to identified as ASD.

```
jundice <- ggplot(data.na, aes(x = jundice, fill = Class_ASD)) + geom_bar(stat = 'count', position = 'd
jundice + labs(x = 'Jundice', y = 'Frequency') + guides(fill = guide_legend(title = "Class/ASD"))
```



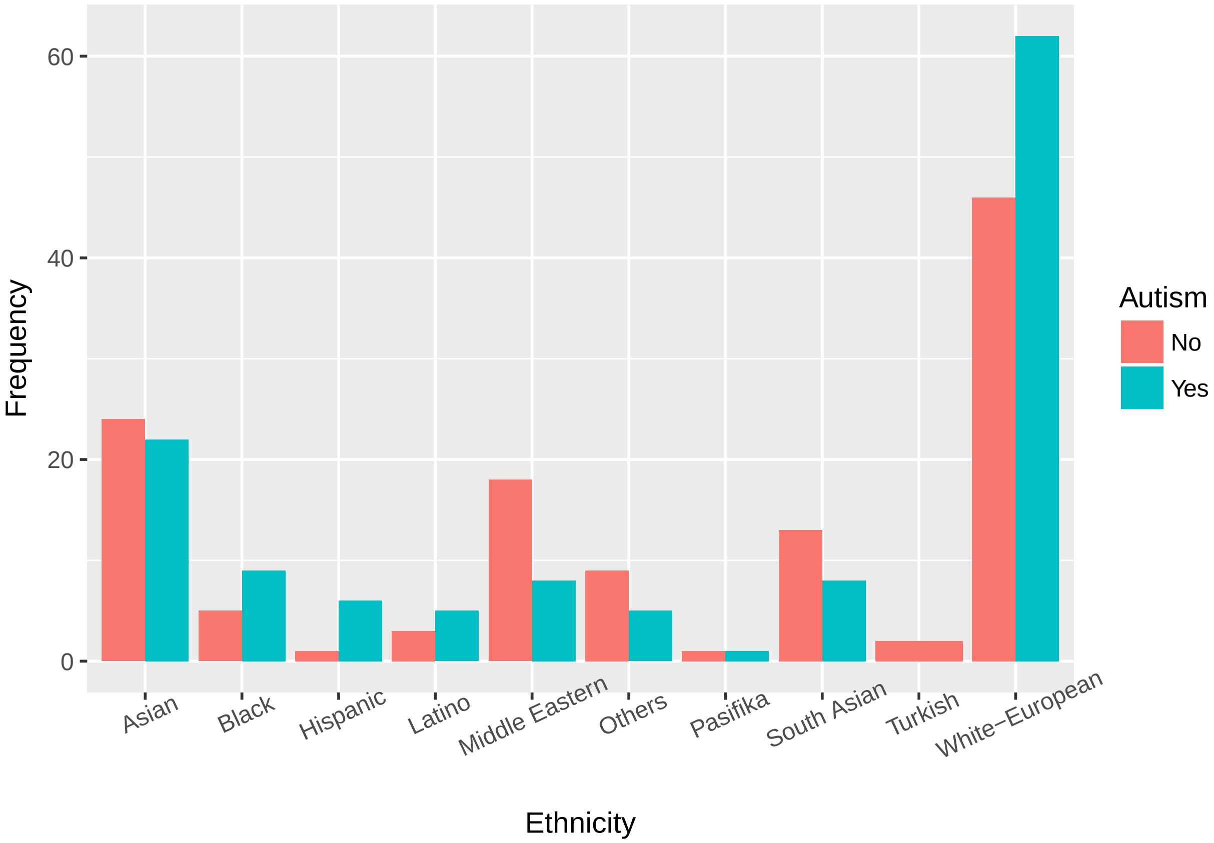
The graph shows that participants identified as ASD does not born with jundice

```
#show the total number of ethnicity
eth = table(data.na$ethnicity)
kable(eth)
```

Var1	Freq
Asian	46
Black	14
Hispanic	7
Latino	8
Middle Eastern	26
Others	14
Pasifika	2
South Asian	21
Turkish	2
White-European	108

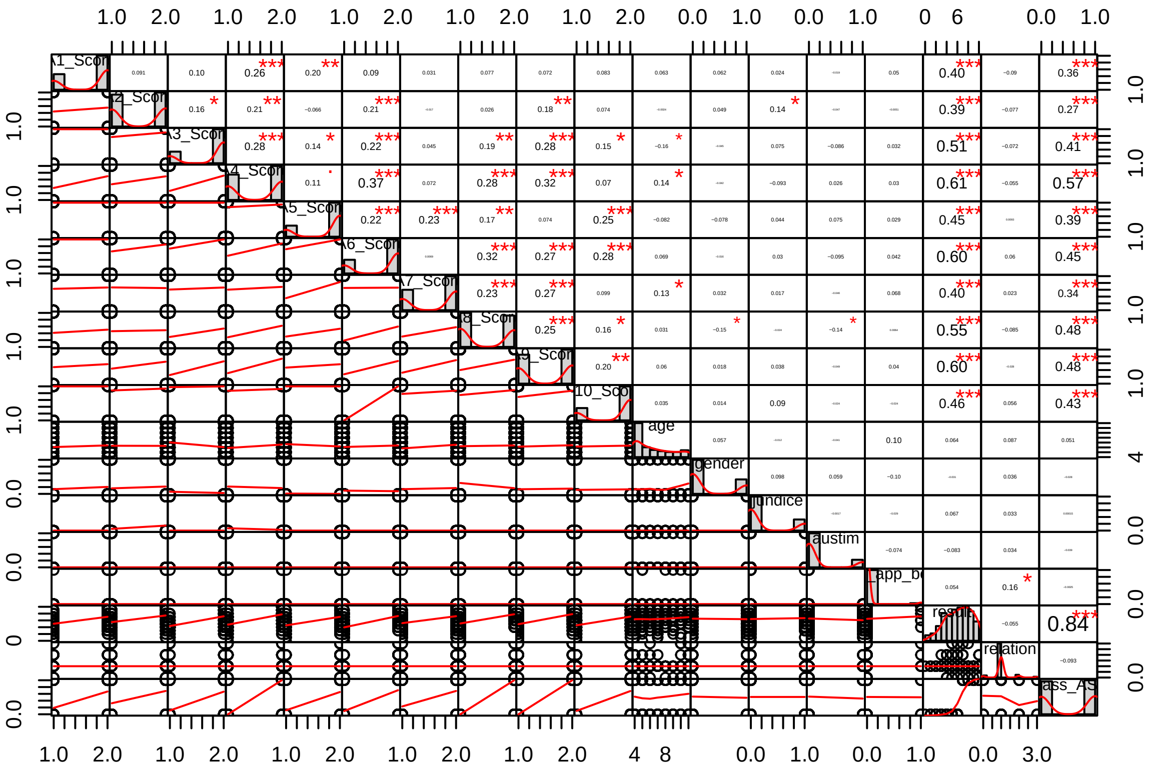
Majority of the particiapant are White European.

```
ethnicity <- ggplot(data.na, aes(x = ethnicity, fill = Class_ASD)) + geom_bar(stat = 'count', position =
ethnicity + labs(x = 'Ethnicity', y = 'Frequency') + guides(fill = guide_legend(title = "Autism")) + the
```

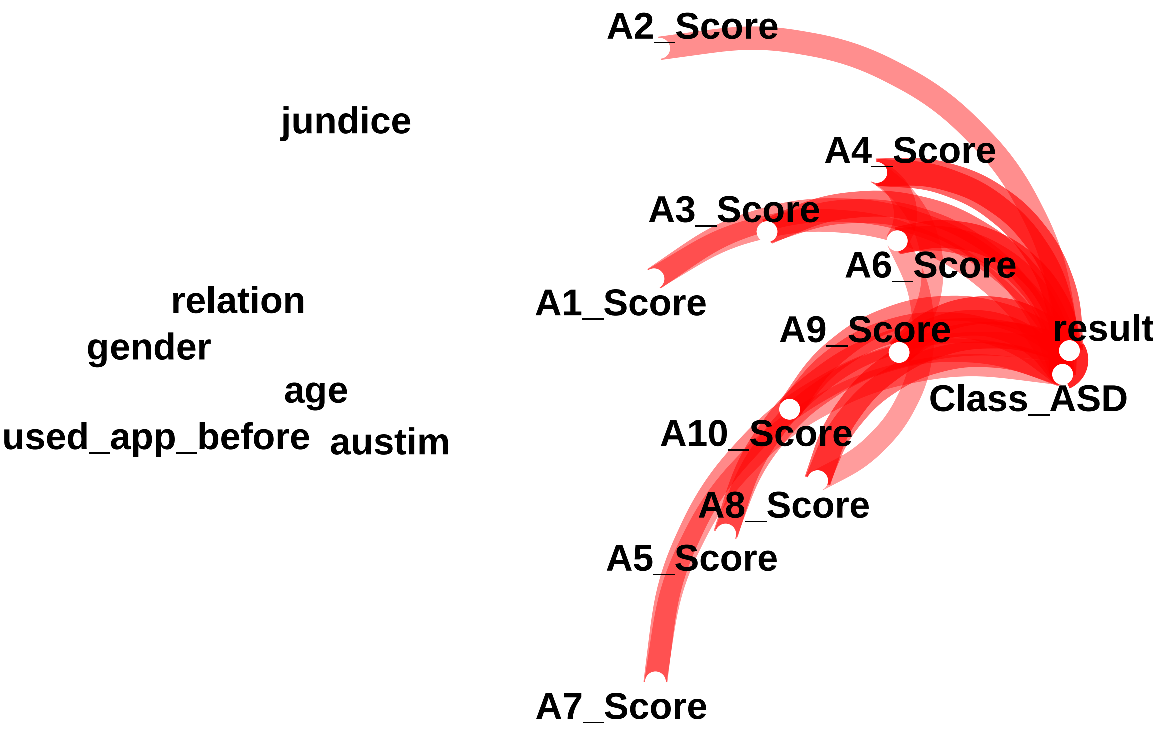


As shown in the graph, Asian and White European have the highest frequenct of identified as Austism.

```
#selecting relevant variable
data.selected <- data.na[,c(1:12,14,15,17,18,20,21)]

data.selected$relation <- as.character(data.selected$relation)
data.selected$relation[data.selected$relation == 'Health care professional'] <- 0
data.selected$relation[data.selected$relation == 'Parent'] <- 1
data.selected$relation[data.selected$relation == 'Relative'] <- 2
data.selected$relation[data.selected$relation == 'self'] <- 3
data.selected$relation[data.selected$relation == 'Self'] <- 3


#recode the factor variable
data.selected$gender <- recode(data.selected$gender, Male = 0, Female = 1)
data.selected$jundice <- recode(data.selected$jundice, No = 0, Yes = 1)
data.selected$austim <- recode(data.selected$austim, No = 0, Yes = 1)
data.selected$used_app_before <- recode(data.selected$used_app_before, no = 0, yes = 1)
data.selected$Class_ASD <- recode(data.selected$Class_ASD, No = 0, Yes = 1)

#unfactor data
data.selected$A1_Score <- as.numeric(data.selected$A1_Score)
data.selected$A2_Score <- as.numeric(data.selected$A2_Score)
data.selected$A3_Score <- as.numeric(data.selected$A3_Score)
data.selected$A4_Score <- as.numeric(data.selected$A4_Score)
data.selected$A5_Score <- as.numeric(data.selected$A5_Score)
data.selected$A6_Score <- as.numeric(data.selected$A6_Score)
data.selected$A7_Score <- as.numeric(data.selected$A7_Score)
data.selected$A8_Score <- as.numeric(data.selected$A8_Score)
data.selected$A9_Score <- as.numeric(data.selected$A9_Score)
data.selected$A10_Score <- as.numeric(data.selected$A10_Score)
data.selected$relation <- as.numeric(data.selected$relation)
data.selected$Class_ASD <- as.numeric(data.selected$Class_ASD)

#correlation
chart.Correlation(data.selected, histogram = T, cex = 30)
```



```
#correlation network
data.selected %>% correlate() %>% network_plot(colors = 'red')
```



As presented in the correlation, it shows that there is highest correlation between the items/questionnaire with ASD. As this can be seen in the correlation network plot.

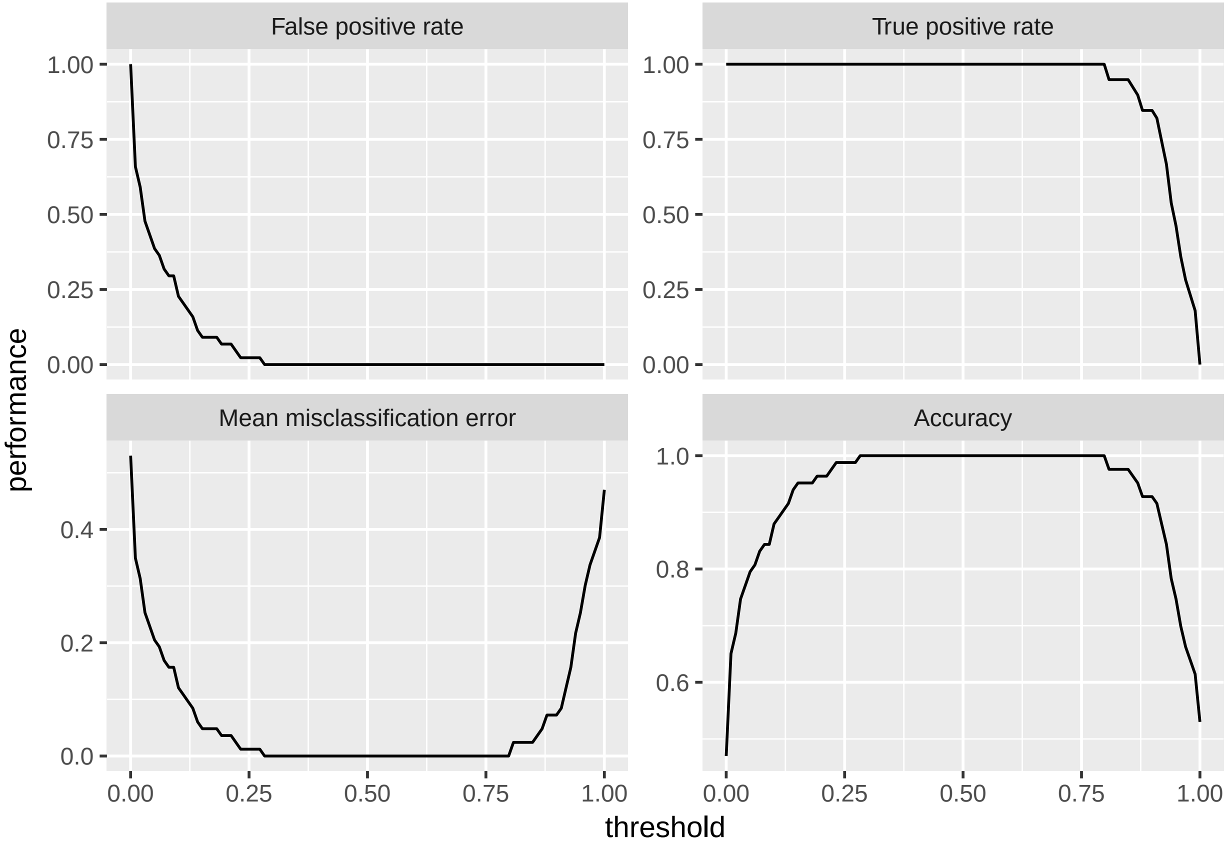
```
#factor data
data.selected$A1_Score <- as.factor(data.selected$A1_Score)
data.selected$A2_Score <- as.factor(data.selected$A2_Score)
data.selected$A3_Score <- as.factor(data.selected$A3_Score)
data.selected$A4_Score <- as.factor(data.selected$A4_Score)
data.selected$A5_Score <- as.factor(data.selected$A5_Score)
data.selected$A6_Score <- as.factor(data.selected$A6_Score)
data.selected$A7_Score <- as.factor(data.selected$A7_Score)
data.selected$A8_Score <- as.factor(data.selected$A8_Score)
data.selected$A9_Score <- as.factor(data.selected$A9_Score)
data.selected$A10_Score <- as.factor(data.selected$A10_Score)
data.selected$gender <- as.factor(data.selected$relation)
data.selected$jundice <- as.factor(data.selected$relation)
data.selected$austim <- as.factor(data.selected$relation)
data.selected$used_app_before <- as.factor(data.selected$relation)
data.selected$relation <- as.factor(data.selected$relation)
data.selected$Class_ASD <- as.factor(data.selected$Class_ASD)


#machine learning classification
#spliting data
n = nrow(data.selected)
train.set = sample(n, size = 2/3*n)
test.set = setdiff(1:n, train.set)

#making ml task
classif.task <- makeClassifTask(data = data.selected, target = 'Class_ASD')
```

```r
#using decision tree algorithm
lrn <- makeLearner('classif.randomForest', predict.type = 'prob')

#train the model
model <- train(lrn, classif.task, subset = train.set)

#predict
pred <- predict(model, classif.task, subset = test.set)

#performance of prediction
performance <- performance(pred, measures = list(fpr, tnr, mmce, acc, mcc))
performance
```

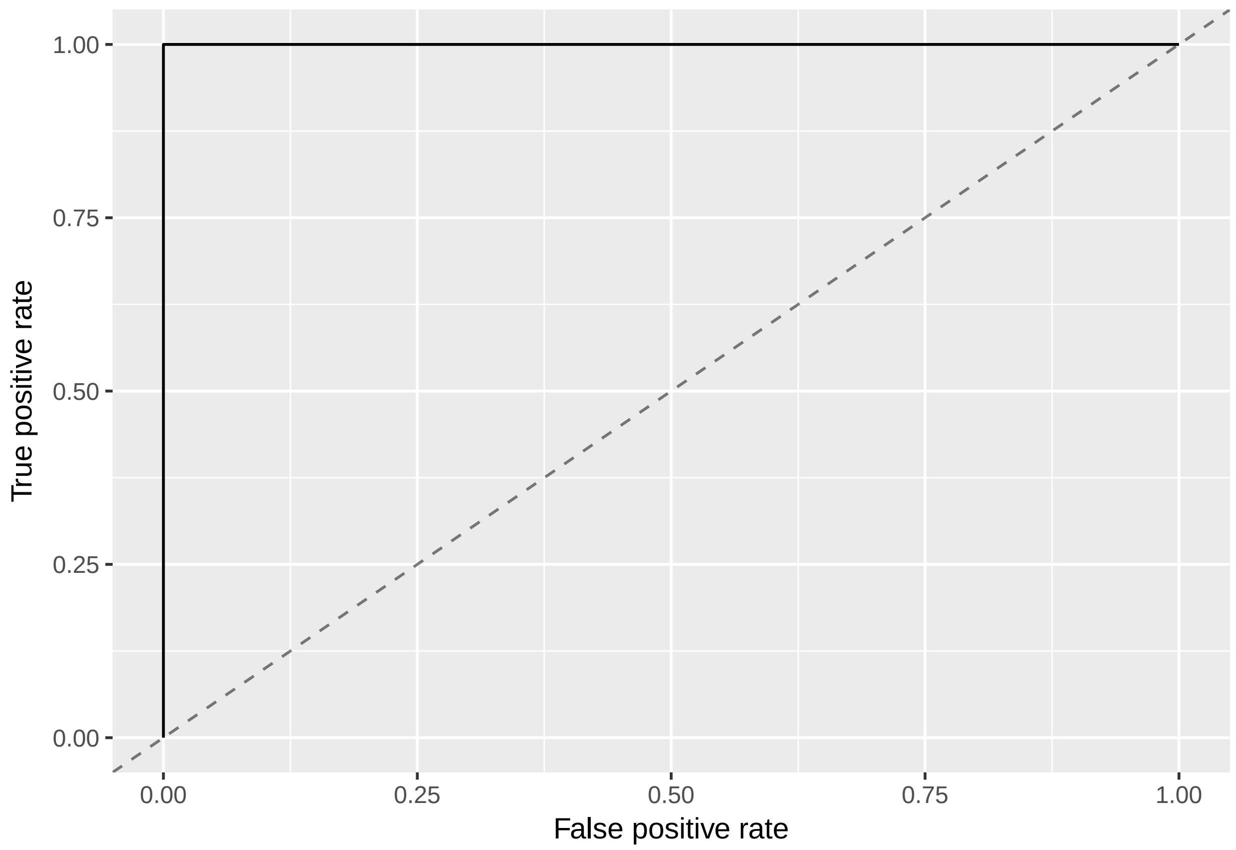
```
##  fpr  tnr mmce  acc  mcc
##    0    1    0    1    1
```

By using random forest algorithm, it shows that it can perfectly predicted all the variable in the test set. This may be occurence of overfitting because the sample size is small.

```r
df = generateThreshVsPerfData(pred, measures = list(fpr, tpr, mmce, acc))
plotThreshVsPerf(df)
```



```r
plotROCCurves(df)
```



```
calculateConfusionMatrix(pred)
```

```
##          predicted
## true       0  1 -err.-
##   0       39  0      0
##   1        0 44      0
##   -err.-  0  0      0
```