Exploratory Analysis of the ATXN2 edata

Diksha Jethnani and Jinko Graham

2024-02-28

```
knitr::opts_chunk$set(echo = TRUE)
```

Read in excel data

We'll use the `readxl` tidyverse package to read in Joanna's excel data sheets.

The `excel_sheets()` function allows users to identify the sheets in excel files. To make plots, we use the ggplot2 library. Going further, we will also use the dplyr package in R to group and understand our data and perform smooth exploratory analysis. We will first import all the necessary packages that we might need during our analysis.

```
library(readxl)
library(ggplot2)
library(stringr)
library(RColorBrewer)
library(stats)
library(tidyr)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
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
```

Let's start by looking at Joanna's `ATXN2persons.xlsx` excel file and see the different sheets that we have in the `ATXN2persons.xlsx` data file.

```
excel_sheets("../ATXN2persons.xlsx")
```

```
## [1] "Info" "Data"
```

The name of the sheet with the data is `Data`, so let's read it into R.

```
persons<-read_excel("../ATXN2persons.xlsx", sheet="Data")
head(persons)
```

```
## # A tibble: 6 x 15
##   SampleIndex SampleNumber sampleID `ATXN2ex1 variant1` `ATXN2ex1 variant2`
##   <chr>              <dbl> <chr>    <chr>               <chr>
## 1 EX                   457 EX457    neg                 neg
## 2 EX                   458 EX458    neg                 neg
## 3 EX                   461 EX461    neg                 neg
## 4 EX                   462 EX462    neg                 neg
## 5 EX                   463 EX463    neg                 neg
## 6 EX                   464 EX464    neg                 neg
## # i 10 more variables: `Clinical information` <chr>, Sex <chr>, DOB <dttm>,
## #   `Family ID` <chr>, `Enrichment kits` <chr>, `Created at` <dttm>,
## #   OtherSampleIDs <chr>, ND <dbl>, PatientID <dbl>, relationships <chr>
```

We notice that we have a data frame with 15 variables.The 358 observations correspond to the total number of samples that we have. We will now make a few modifications to our data frame under the pre processing step to facilitate our further analysis. As we will be using the SKAT package (SNP-Set(Sequence) Kernel Association Test) for our analysis,it is important to reshape our data as per the package requirements.

```
persons$ND <- factor(persons$ND, levels = c(1, 0, 3), labels = c("Yes", "No", "Maybe"))
categorical_vars <- subset(persons, select = c("SampleIndex", "Sex", "ND", "Clinical information", "Enr
```

Adding new column to our data frame-

Here, we add a new column to our data frame. This new column tells us if a person has a variant or not. The entries in the column are "yes" or "no" respectively where "yes" signifies that the person has a variant and "no" tells us that the person has none of the variants present.

```
# Update the "Variant" column based on conditions
persons$Variant <- ifelse(persons$`ATXN2ex1 variant1` == "neg" & persons$`ATXN2ex1 variant2` == "neg" ,
head(persons)
```

```
## # A tibble: 6 x 16
##   SampleIndex SampleNumber sampleID `ATXN2ex1 variant1` `ATXN2ex1 variant2`
##   <chr>              <dbl> <chr>    <chr>               <chr>
## 1 EX                   457 EX457    neg                 neg
## 2 EX                   458 EX458    neg                 neg
## 3 EX                   461 EX461    neg                 neg
## 4 EX                   462 EX462    neg                 neg
## 5 EX                   463 EX463    neg                 neg
## 6 EX                   464 EX464    neg                 neg
## # i 11 more variables: `Clinical information` <chr>, Sex <chr>, DOB <dttm>,
## #   `Family ID` <chr>, `Enrichment kits` <chr>, `Created at` <dttm>,
## #   OtherSampleIDs <chr>, ND <fct>, PatientID <dbl>, relationships <chr>,
## #   Variant <chr>
```

We will add another column to our data frame, this column tells us the number of variants that person has

i.e. 0/1/2.

```
# Create the "Total_variants" column
persons$Total_Variants <- ifelse(persons$`ATXN2ex1 variant1` == "neg" & persons$`ATXN2ex1 variant2` ==
```

We will now add another column to our data frame. For our new column, we will calculate the AGE of each person in order to help us in our further analysis where we might want to test if there is any association in the type of disease with respect to the age of the person.

To calculate the age of the person, we will use the date of birth of every person along with the date on which the person was sampled. This will give us the estimate of their age as to when the sample was taken. We can now have the entries for the new column in our data frame specifying the age of each person.

This would be very beneficial for us in order to relate the different characteristics of the person and see if their age acts as a factor in determining the type of disease they suffer from.

```
# Calculate age from date of birth
persons$`Created at` <-  as.Date(persons$`Created at`)
persons$age <- year(persons$`Created at`) - year(persons$DOB)

# Adjust age for leap year cases
persons$age <- ifelse(month(persons$DOB) > month(persons$`Created at`) | (month(persons$DOB) == month(p

# Descriptive statistics for age
summary(persons$age)      # Summary statistics
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   26.00   51.50   44.37   62.00   85.00
```

```
sd(persons$age)           # Standard deviation
```

```
## [1] 23.75052
```

In the data provided to us by Joanna, she warns us about some questionable variants that are colored in "yellow" in the excel sheet. As we are not very confident about them, we would like to check for any specific association or trends that these samples exhibit. If not, it would be ideal to set the variant status as 'missing' for the samples having these variants as we do not trust them. So, to check that, we will add another column to our persons data frame. This column corresponds to the sample indexes that contain a questionable variant.

```
#Questionable variants
persons$'Questionable Variant' <- ifelse(
 persons$SampleNumber%in% c(120, 182, 15, 96, 236, 131, 128, 182, 17, 137, 48),
  "yes",
  "no"
)
```

Univariate Summaries

After adding the necessary details,we move to the next step of exploring the data. We will start by looking at the univariate summaries. i.e. we will create a table summarizing all the unique values along with their counts. These uni-variate summaries would let us have an overview of the data and direct us towards the type of analysis that we could potentially perform on our data.

We will start by looking at the sample index column.

```
  summary_SampleIndex <- persons %>%
    group_by(SampleIndex) %>%
```

```
    summarise(count = n())
 print(summary_SampleIndex)
```

```
## # A tibble: 15 x 2
##    SampleIndex count
##    <chr>       <int>
##  1 EX              8
##  2 P-ALS          93
##  3 P-AMY           3
##  4 P-AX           22
##  5 P-BGW          20
##  6 P-DIV          15
##  7 P-DYT           6
##  8 P-HL            3
##  9 P-MH            4
## 10 P-MY           42
## 11 P-NP           52
## 12 P-SPG          19
## 13 P-SW           3
## 14 P-SY           58
## 15 P-TM           10
```

We see that we have 15 different sample indexes with the frequency of each one of them as stated in the table. ALS(93), which has been categorized as ND has the most frequent occurrence in the list followed by SY(58) and NP(52) that are NON-ND and undetermined respectively.

We will similarly look at the univariate summary table for the gender.

```
summary_Sex <- persons %>%
  group_by(Sex) %>%
  summarise(count= n())
print(summary_Sex)
```

```
## # A tibble: 2 x 2
##   Sex    count
##   <chr>  <int>
## 1 Female   177
## 2 Male     181
```

The table comparing the two sex tells us that we have 177 females and 181 males in the study. This ensures that we have a fair representation of both the genders.

```
summary_Sex_q  <- persons %>% filter(`Questionable Variant`=="yes") %>%
  group_by(Sex) %>%
  summarise(count= n())
print(summary_Sex_q)
```

```
## # A tibble: 2 x 2
##   Sex    count
##   <chr>  <int>
## 1 Female    13
## 2 Male       9
```

Let's now see the distribution of the total persons based on the classification of the disease i.e. how many of them have a disease that is ND and how many of them have a disease that is Non-ND.

```
summary_ND <- persons %>%
  group_by(ND) %>%
  summarise(count= n())
print(summary_ND)
```

```
## # A tibble: 3 x 2
##   ND    count
##   <fct> <int>
## 1 Yes     134
## 2 No      161
## 3 Maybe    63
```

On looking at the uni-variate distribution of the ND column, we see that in our study 168 people are having a disease that is non-neuro degenerative while 134 people have a neuro degenerative disease. A total of 56 subjects have a disease that is yet not classified as ND or Non-ND because of overlapping symptoms.

Let us now try to look at the ND status for the questionable variants-

```
summary_ND_q <- persons %>% filter(`Questionable Variant`=="yes") %>%
  group_by(ND) %>%
  summarise(count= n())
print(summary_ND_q)
```

```
## # A tibble: 3 x 2
##   ND    count
##   <fct> <int>
## 1 Yes       8
## 2 No       12
## 3 Maybe     2
```

So,the questionable variants give rise to both ND and non-ND diseases.

Univaraite summary for the enrichment kits-

```
persons$`Enrichment kits` <- factor(persons$`Enrichment kits`, levels = c("Twist Comprehensive Exome Re
"Twist Comprehensive Exome plus Refseq",
 "Twist Comprehensive Exome plus Refseq, Twist Exome 2.0, Twist Mix (Comprehensive plus 2.0)", "Twist M
), labels = c("TCER vs2", "Exon v7","TCE(RefSeq)","TCE(R,Ex,Mix)", "T Mix"))
```

```
E_kits <- persons %>%
  group_by(`Enrichment kits`) %>%
  summarise(count= n())
print(E_kits)
```

```
## # A tibble: 5 x 2
##   `Enrichment kits` count
##   <fct>             <int>
## 1 TCER vs2            240
## 2 Exon v7               4
## 3 TCE(RefSeq)          79
## 4 TCE(R,Ex,Mix)         3
## 5 T Mix                32
```

There are 5 different kinds of enrichment kits being used where 'Twist Comprehensive Exome Refseq vs2' is the one that is the most frequent (240 times).

Moving to our next variable of Clinical Information, it is a little difficult to look at it if we summarize it based on the different types of clinical information available about the disease, given the number of different

values. So, we will group all the entries together under "Others'' if they occur less than 3 times.

```
ClinicalInfo<- persons %>%
  group_by(`Clinical information`) %>%
  summarise(count = n())

#Merge rows with value <= 3 in the 'count' column.
merged_row <- ClinicalInfo%>%
  filter(count <= 3) %>%
  summarise(column1 = "others",
            column2 = n())
colnames(merged_row) <- colnames(ClinicalInfo)

# Remove rows with value <= 3 in the count column.
filtered_data <-ClinicalInfo %>%
  filter(count > 3)

# Combine the filtered data with the merged row
ClinicalInfo <- rbind(filtered_data, merged_row)
names(ClinicalInfo) <- c("Info", "count")
ClinicalInfo$Label <- str_extract(ClinicalInfo$Info, "\\w+")

# Print the final merged table
print(ClinicalInfo)
```

```
## # A tibble: 10 x 3
##    Info                                                          count Label
##    <chr>                                                         <int> <chr>
##  1 (+) Amyotrophic lateral sclerosis                                74 Amyo~
##  2 (+) Amyotrophic lateral sclerosis, (+) Frontotemporal dementia    5 Amyo~
##  3 (+) Delayed speech and language development, (+) Global developm~     4 Dela~
##  4 (+) Malignant hyperthermia                                       4 Mali~
##  5 (+) Myopathy                                                     4 Myop~
##  6 (+) Peripheral axonal neuropathy, (+) Polyneuropathy             4 Peri~
##  7 (+) Polyneuropathy                                               4 Poly~
##  8 (+) Spastic paraplegia                                           4 Spas~
##  9 trio parent                                                      8 trio
## 10 others                                                         235 othe~
```

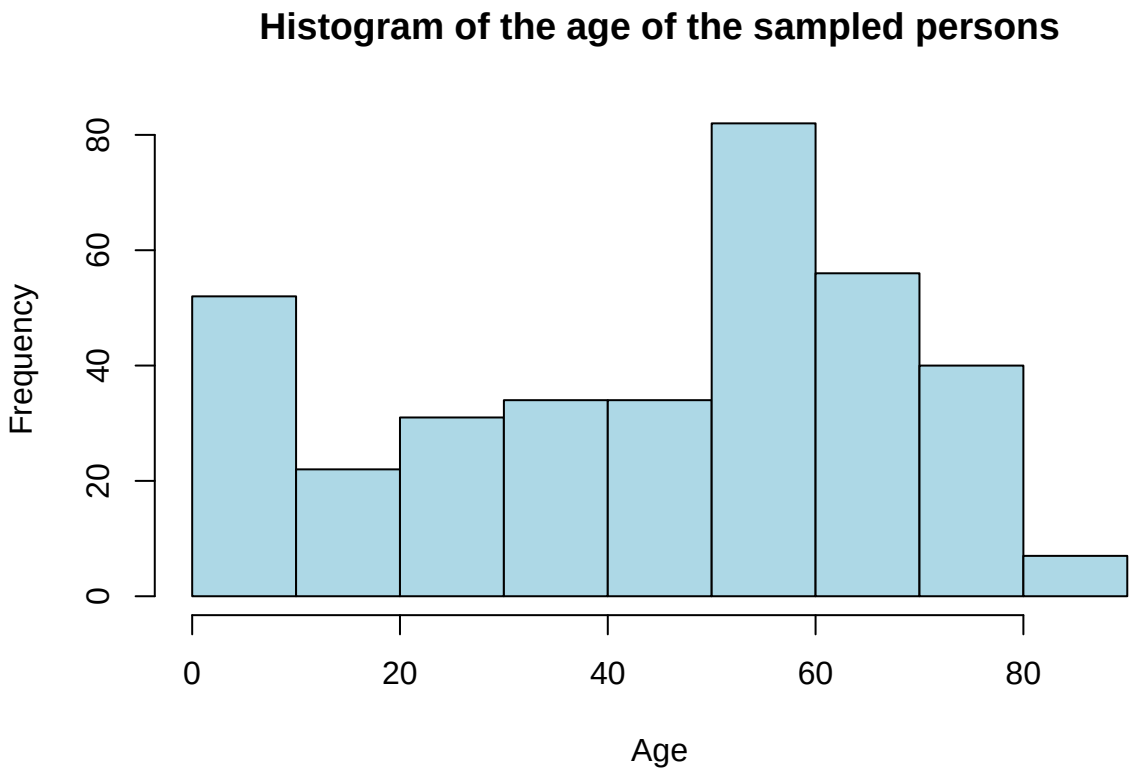
So, we see that there are 9 different unique 'Clinical Information' available about the patients that occur more than 3 times. All the other information have been grouped under the row 'Others'. We notice that the most frequent occurrence is of Amyotrophic lateral sclerosis(74 times). All the others have a frequency of 5 or less.

Here, note that, we have a category of "trio parents" that is just the data of some families that were considered during the data collection as Joanna mentions. We can, (on a later stage) choose and decide if we want to consider these families in our control group or not.

Histogram of Age distribution

Let us now look at how the age of our sampled individuals is distributed. For this, we will plot a histogram of the ages of the persons.

```
hist(persons$age, breaks = 10, col = "lightblue", main = "Histogram of the age of the sampled persons",
```



From the histogram we can see that the majority of persons sampled were from the age group 50-60 years followed by the 60-70 age group. Going a step further, we will later look at the distribution of age based on gender or the disease being ND or Non-ND to see if there is a particular trend in the age distribution.

Bivariate Summaries

Now that we have analyzed the uni variate summaries for our different variables, let's see the trends and relationships (if any) between the different variables. We can start looking at it by the bivariate summaries using contingency tables to summarize the associated counts. We can then use tests like, fisher's exact test/ permutation test/ chi-square test based on the data to analyze if the different variables are independent or not. We will also see if there are any confounding variables and adjust for them before moving to the next step.

Let us start by looking at the Sample Index and the sex of the the sampled individuals.

```
#selected_columns <- persons[, c("SampleIndex", "Sex", "ND", "Clinical information", "Enrichment_kits")]
contingency_table_1 <- table(persons$SampleIndex, persons$Sex)
print(contingency_table_1)
```

```
##
##         Female Male
##   EX         4    4
##   P-ALS     49   44
##   P-AMY      0    3
##   P-AX      13    9
##   P-BGW     16    4
##   P-DIV      6    9
##   P-DYT      3    3
##   P-HL       1    2
##   P-MH       3    1
##   P-MY      19   23
```

```
##   P-NP      22   30
##   P-SPG      9   10
##   P-SW       2    1
##   P-SY      24   34
##   P-TM       6    4
```

```
#Test for association
result <- fisher.test(contingency_table_1, simulate.p.value = TRUE, B = 10000)
p_value <- result$p.value
print(p_value)
```

```
## [1] 0.2277772
```

The table helps us see how the different sample indexes are distributed among the two genders. We see that since the p-value is relatively large (greater than the commonly used significance level of 0.05), we do not have sufficient evidence to reject the null hypothesis of independence. This means that there is no strong evidence to suggest that there is a relationship between the Sample Index and the gender of the patient based on the data.

Hence, we can assure that the gender of a person does not play any significant role in determining the type of disease they suffer from.

Going further, let us look at the Sample Index and the Enrichment kits used.

```
contingency_table_2 <- table(persons$SampleIndex, persons$`Enrichment kits`)
print(contingency_table_2)
```

```
##
##         TCER vs2 Exon v7 TCE(RefSeq) TCE(R,Ex,Mix) T Mix
##   EX           8       0           0             0     0
##   P-ALS       79       0           4             0    10
##   P-AMY        2       0           1             0     0
##   P-AX        10       1           6             2     3
##   P-BGW       14       0           5             0     1
##   P-DIV       11       0           3             0     1
##   P-DYT        2       0           3             0     1
##   P-HL         2       0           1             0     0
##   P-MH         2       0           2             0     0
##   P-MY        24       0          14             0     4
##   P-NP        36       0           9             1     6
##   P-SPG       10       1           7             0     1
##   P-SW         2       0           1             0     0
##   P-SY        35       2          16             0     5
##   P-TM         3       0           7             0     0
```

```
#Test of association
result <- fisher.test(contingency_table_2, simulate.p.value = TRUE, B = 10000)
p_value <- result$p.value
print(p_value)
```

```
## [1] 0.00049995
```

This is quite interesting result. As we obtain such a small p value, it implies that there is a significant relationship between the Sample Index and Enrichment kits being used. One possible explanation could be the fact that the persons were sampled in different labs and that a particular lab useed a particular type of enrichment kit.

```
contingency_table_3 <- table(persons$SampleIndex, persons$ND)
print(contingency_table_3)
```

```
##
##         Yes No Maybe
##   EX      0  8     0
##   P-ALS  93  0     0
##   P-AMY   0  0     3
##   P-AX   22  0     0
##   P-BGW   0 20     0
##   P-DIV   0 13     2
##   P-DYT   0  0     6
##   P-HL    0  3     0
##   P-MH    0  4     0
##   P-MY    0 42     0
##   P-NP    0  0    52
##   P-SPG  19  0     0
##   P-SW    0  3     0
##   P-SY    0 58     0
##   P-TM    0 10     0
```

```
result <- fisher.test(contingency_table_3, simulate.p.value = TRUE, B = 10000)
p_value <- result$p.value
print(p_value)
```

```
## [1] 9.999e-05
```

On applying the fisher test, the obtained p value i.e. (9.999e-05 or 0.00009999) shows us that there is strong evidence to suggest that there is a significant relationship between the Sample Index and the ND status variable for our given data. This is also an intuitive interpretation given that the sample Index has been decided in order to represent the disease and the disease in turn is either ND or Non-ND.

#Sample index~ Variant

```
contingency_table_4<- table(persons$SampleIndex, persons$Variant)
print(contingency_table_4)
```

```
##
##         no yes
##   EX     8   0
##   P-ALS 80  13
##   P-AMY  2   1
##   P-AX  17   5
##   P-BGW 19   1
##   P-DIV 14   1
##   P-DYT  5   1
##   P-HL   3   0
##   P-MH   3   1
##   P-MY  41   1
##   P-NP  50   2
##   P-SPG 15   4
##   P-SW   1   2
##   P-SY  53   5
##   P-TM  10   0
```

```
result <- fisher.test(contingency_table_4, simulate.p.value = TRUE, B = 10000)
p_value <- result$p.value
print(p_value)
```

```
## [1] 0.01439856
```

A p value of 0.0146 tells us that there is a significant relationship between the Sample index and the presence or absence of a variant.

#Sample Index ~ Questionable Variant

```
contingency_table_5 <- table(persons$SampleIndex, persons$`Questionable Variant`)
print(contingency_table_5)
```

```
##
##         no yes
##   EX     8   0
##   P-ALS 88   5
##   P-AMY  3   0
##   P-AX  19   3
##   P-BGW 19   1
##   P-DIV 14   1
##   P-DYT  6   0
##   P-HL   3   0
##   P-MH   2   2
##   P-MY  37   5
##   P-NP  51   1
##   P-SPG 19   0
##   P-SW   3   0
##   P-SY  54   4
##   P-TM  10   0
```

```
result <- fisher.test(contingency_table_5, simulate.p.value = TRUE, B = 10000)
p_value <- result$p.value
print(p_value)
```

```
## [1] 0.2174783
```

A large p value of 0.21 tells us that Sample Index and Questionable variants do not have a relationship.

#ND Status~Sex

```
contingency_table_6<- table(persons$ND, persons$Sex)
print(contingency_table_6)
```

```
##
##         Female Male
##   Yes       71   63
##   No        80   81
##   Maybe     26   37
```

This contingency table for gender and the ND status helps us see how the ND and Non-Nd diseases are distributed over the two gender. Let us now perform a chi square test to check if these are independent or not.

```
result <- chisq.test(contingency_table_6)
print(result)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table_6
## X-squared = 2.3601, df = 2, p-value = 0.3073
```

The test statistic, is 2,36 As we know that this value represents the discrepancy between the observed frequencies in the contingency table and the frequencies that would be expected under the assumption of independence between the variables. Also, we see that in this case, we obtain a p-value of 0.30, this value is quite high and we can say that we do not have sufficient evidence to reject the null hypothesis of independence or in other words we can say that we do not have significant evidence to conclude that there is a relationship between the gender of a person and the ND status of a disease.

#ND Status~ E_Kits

```
contingency_table_7 <- table(persons$ND, persons$`Enrichment kits`)
print(contingency_table_7)
```

```
##
##         TCER vs2 Exon v7 TCE(RefSeq) TCE(R,Ex,Mix) T Mix
##   Yes          99       2          17             2    14
##   No           99       2          49             0    11
##   Maybe        42       0          13             1     7
```

```
result <- fisher.test(contingency_table_7, simulate.p.value = TRUE, B = 10000)
p_value <- result$p.value
print(p_value)
```

```
## [1] 0.01189881
```

The p-value (0.01) indicates a significant association between the enrichment kits used and the ND status of the disease. We can think about this association, directing us towards thinking if there were a particular type of enrichment kit used in a certain lab and the sampled individuals had a certain type of diagnosis.

#ND Status~Variant

Let us check the association of the ND status with the presence or absence of a variant-

```
contingency_table_8 <- table(persons$ND, persons$Variant)
print(contingency_table_8)
```

```
##
##          no yes
##   Yes   112  22
##   No    151  10
##   Maybe  58   5
```

```
result <- fisher.test(contingency_table_8, simulate.p.value = TRUE, B = 10000)
p_value <- result$p.value
print(p_value)
```

```
## [1] 0.01649835
```

This p value tells us that there is a relationship between the ND status and the presence or absence of a variant.

#ND Status~ Questionable Variant

```
contingency_table_9 <- table(persons$ND, persons$`Questionable Variant`)
print(contingency_table_9)
```

```
##
##          no yes
##   Yes   126   8
##   No    149  12
##   Maybe  61   2
```

```
result <- fisher.test(contingency_table_9, simulate.p.value = TRUE, B = 10000)
p_value <- result$p.value
print(p_value)
```

```
## [1] 0.5412459
```

Clearly, such big p value tells us that there is not any relationship between the ND status of a variant and it being questionable.

#Sex ~Ekits Let us now create a table contrasting the gender with the enrichment kits used.

```
contingency_table_10 <- table(persons$Sex, persons$`Enrichment kits`)
print(contingency_table_10)
```

```
##
##          TCER vs2 Exon v7 TCE(RefSeq) TCE(R,Ex,Mix) T Mix
##   Female      121       2          37             2    15
##   Male        119       2          42             1    17
```

```
result <- fisher.test(contingency_table_10, simulate.p.value = TRUE, B = 10000)
p_value <- result$p.value
print(p_value)
```

```
## [1] 0.9464054
```

Since the permutation p-value (0.950405) is greater than the conventional significance level of 0.05, we do not have sufficient evidence to reject the null hypothesis. The null hypothesis typically states that there is no significant difference between the group means. In this case, we obtain a high p-value and we can say that we do not have sufficient evidence to reject the null hypothesis of independence or in other words we can say that we do not have significant evidence to conclude that there is a relationship between the gender of a person and the enrichment kits.

#Sex~Variant

```
contingency_table_11 <- table(persons$Sex, persons$Variant)
print(contingency_table_11)
```

```
##
##           no yes
##   Female 162  15
##   Male   159  22
```

```
result <- fisher.test(contingency_table_11, simulate.p.value = TRUE, B = 10000)
p_value <- result$p.value
print(p_value)
```

```
## [1] 0.2987605
```

We can see that sex and presence or absence of a variant are independent of each other.

#Sex~Questionable

```
contingency_table_12 <- table(persons$Sex, persons$`Questionable Variant`)
print(contingency_table_12)
```

```
##
##           no yes
##   Female 164  13
##   Male   172   9
```

```
result <- fisher.test(contingency_table_12, simulate.p.value = TRUE, B = 10000)
p_value <- result$p.value
print(p_value)
```

```
## [1] 0.3854749
```

We can see that sex and the questionable variants are independent of each other.

#Enrichment kits~ Presence or absence of a variant

```
contingency_table_13 <- table(persons$Variant, persons$`Enrichment kits`)
print(contingency_table_13)
```

```
##
##       TCER vs2 Exon v7 TCE(RefSeq) TCE(R,Ex,Mix) T Mix
##   no       221       0          67             3    30
##   yes       19       4          12             0     2
```

```
result <- fisher.test(contingency_table_13, simulate.p.value = TRUE, B = 10000)
p_value <- result$p.value
print(p_value)
```

```
## [1] 0.00029997
```

Enrichment kits ~ Questionable

```
contingency_table_14 <- table(persons$'Questionable Variant', persons$`Enrichment kits`)
print(contingency_table_14)
```

```
##
##       TCER vs2 Exon v7 TCE(RefSeq) TCE(R,Ex,Mix) T Mix
##   no       230       3          72             3    28
##   yes       10       1           7             0     4
```

So, we can say that the questionable variants do not really have any association with the enrichment kits being used.

```
result <- fisher.test(contingency_table_14, simulate.p.value = TRUE, B = 10000)
p_value <- result$p.value
print(p_value)
```

```
## [1] 0.06739326
```

Presence or absence of variant and questionable variants

```
contingency_table_15 <- table(persons$'Questionable Variant', persons$Variant)
print(contingency_table_15)
```

```
##
##        no yes
##   no  310  26
##   yes  11  11
```

```
result_7 <- chisq.test(contingency_table_15)
```

```
## Warning in chisq.test(contingency_table_15): Chi-squared approximation may be
```

```
## incorrect
print(result_7)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  contingency_table_15
## X-squared = 35.366, df = 1, p-value = 2.732e-09
```

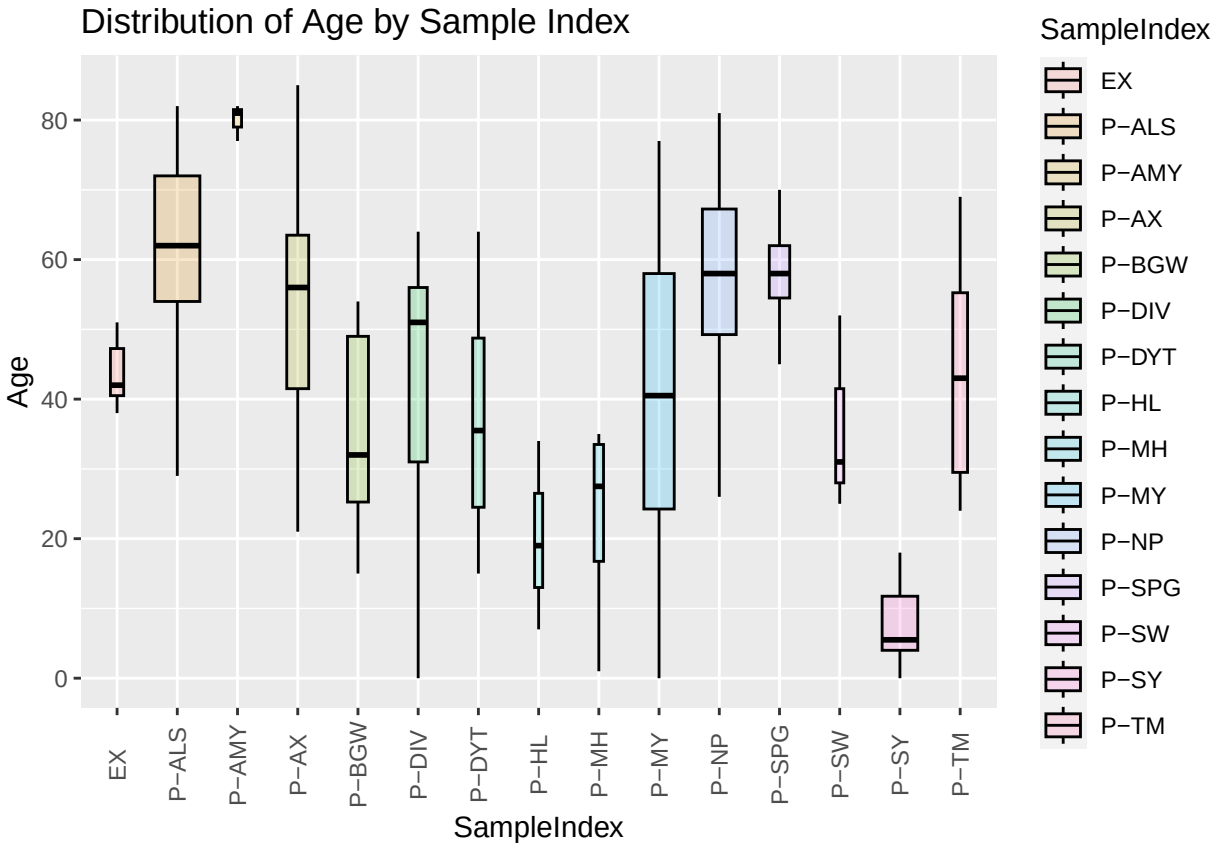
```
result <- fisher.test(contingency_table_15, simulate.p.value = TRUE, B = 10000)
p_value <- result$p.value
print(p_value)
```

```
## [1] 1.029308e-06
```

Bivariate summaries involving age.

Let us now visualize the distribution of age by different variables and see if we can derive some insights from it. We will start by plotting the sample index with the age to see how is the age distributed about the various different sample indices.

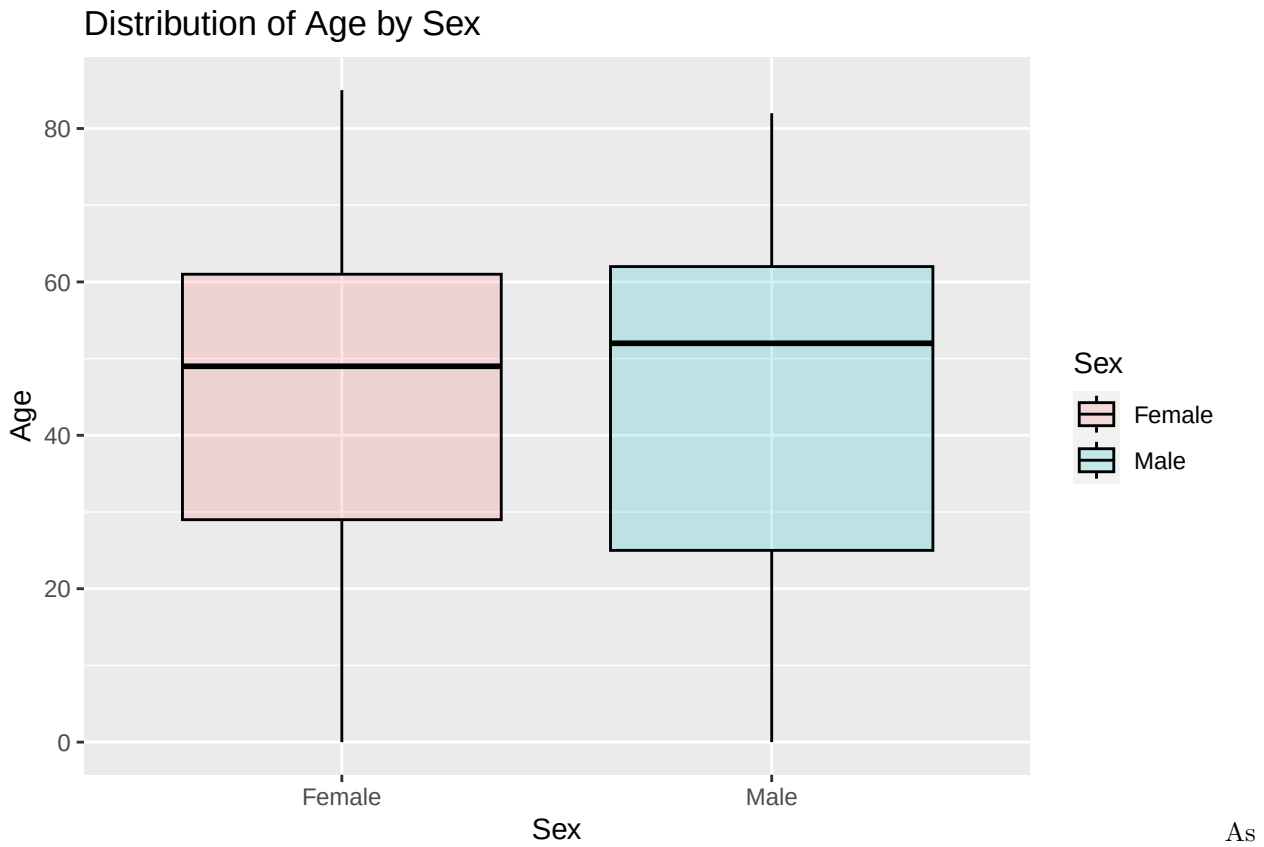
```
ggplot(persons, aes(x = SampleIndex, y = age, fill = SampleIndex)) +
  geom_boxplot(color = "black", outlier.shape = NA, varwidth = TRUE, alpha=0.2) +  labs(x = "SampleIndex
  ggtitle("Distribution of Age by Sample Index") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



The above plot does not convey a lot of information other than patients with some conditions (e.g. HL and SY) tend to be younger.

Let us now look at the distribution of age by sex.

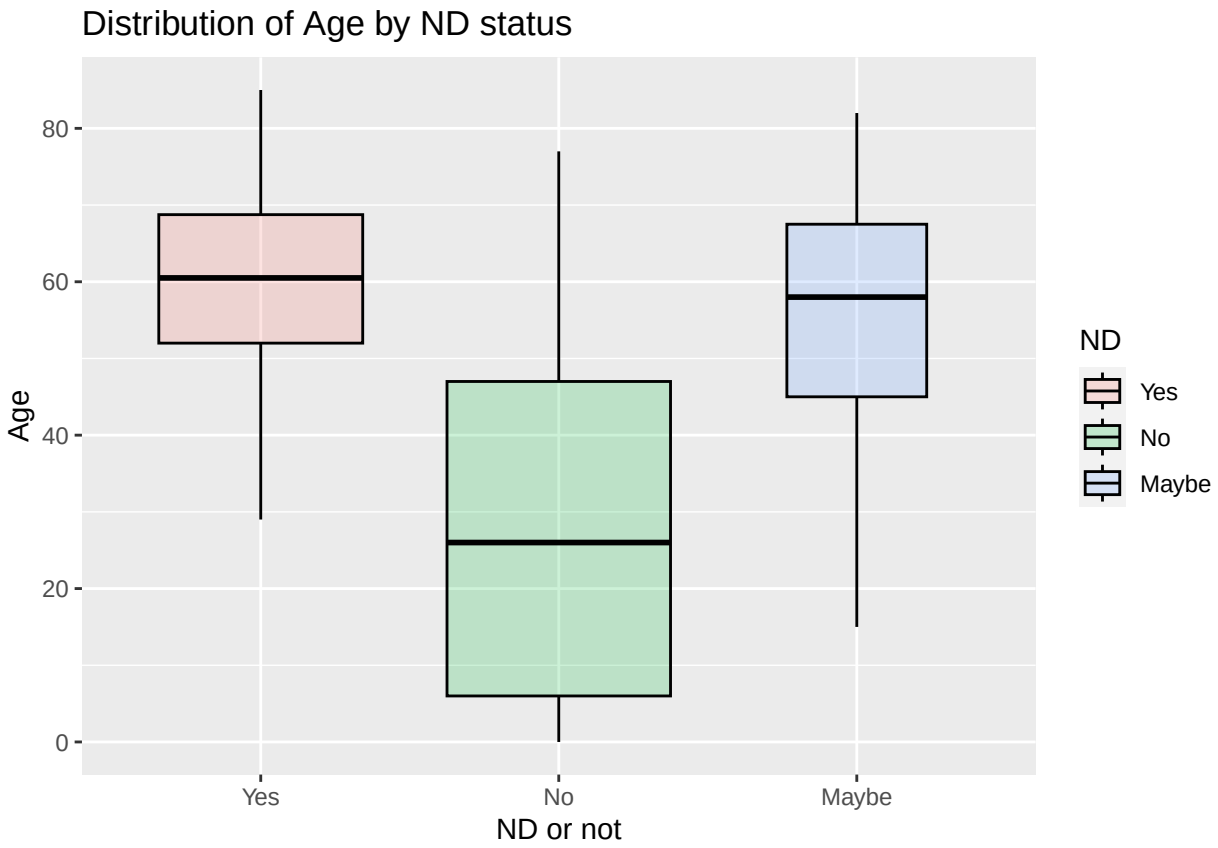
```
ggplot(persons, aes(x =Sex, y = age, fill=Sex)) +
  geom_boxplot(color = "black", outlier.shape = NA, varwidth = TRUE, alpha=0.2) +
  labs(x = "Sex", y = "Age") +
  ggtitle("Distribution of Age by Sex")
```



As is evident from the plot, a similar number of males and females were sampled and the average age of the males is a bit higher than the average age of females.

Going further, let us look at the distribution of age by the ND status of a disease

```
#Age by ND status
ggplot(persons, aes(x =ND, y = age , fill=ND)) +
  geom_boxplot(color = "black", outlier.shape = NA ,varwidth = TRUE, alpha=0.2) +  labs(x = "ND or not"
  ggtitle("Distribution of Age by ND status")
```



The plot suggests that the people who had a neuro-degenerative disease tend to be in the higher age group. In particular, people above the age of 40 years appear to be more susceptible to ND diseases. Later, we will verify this with a bootstrap test on our data.

Looking ahead to the model fitting, let's use a loess data smoother (implemented in the `gam` package) to explore how the probability of ND varies as a function of age. This will help us to see if a linear adjustment for age in the logistic regression modeling will suffice.

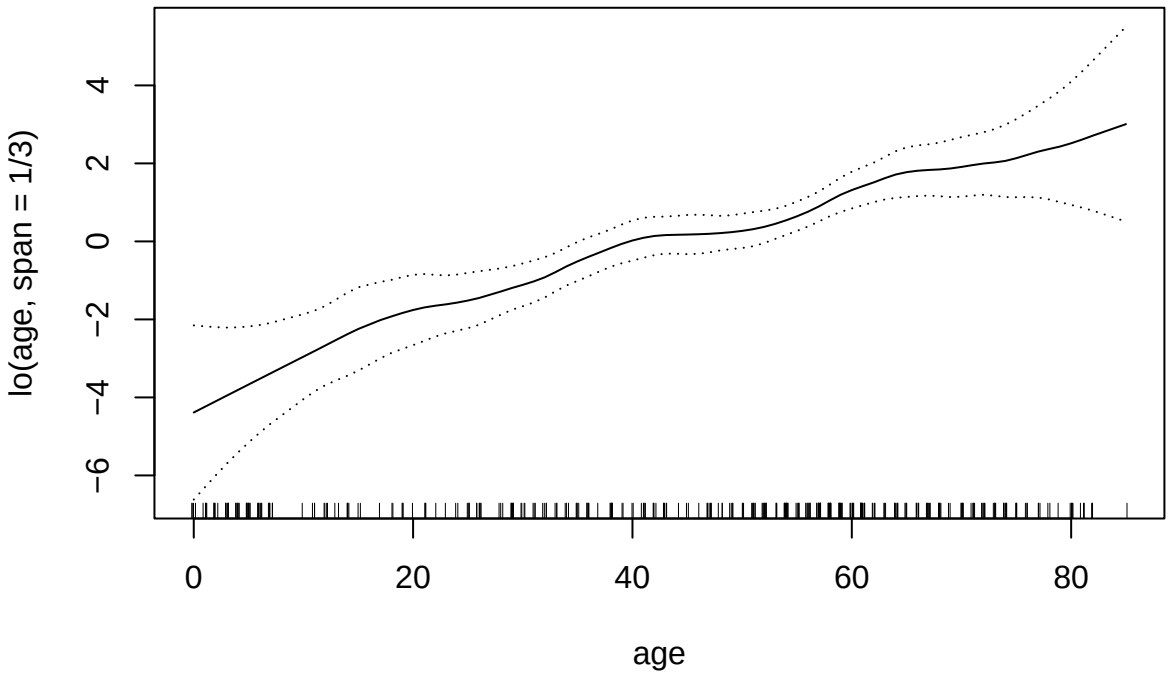
```
newND<-(persons$ND=="Yes"|persons$ND=="Maybe")
library(gam)
```

```
## Loading required package: splines
```

```
## Loading required package: foreach
```

```
## Loaded gam 1.22-2
```

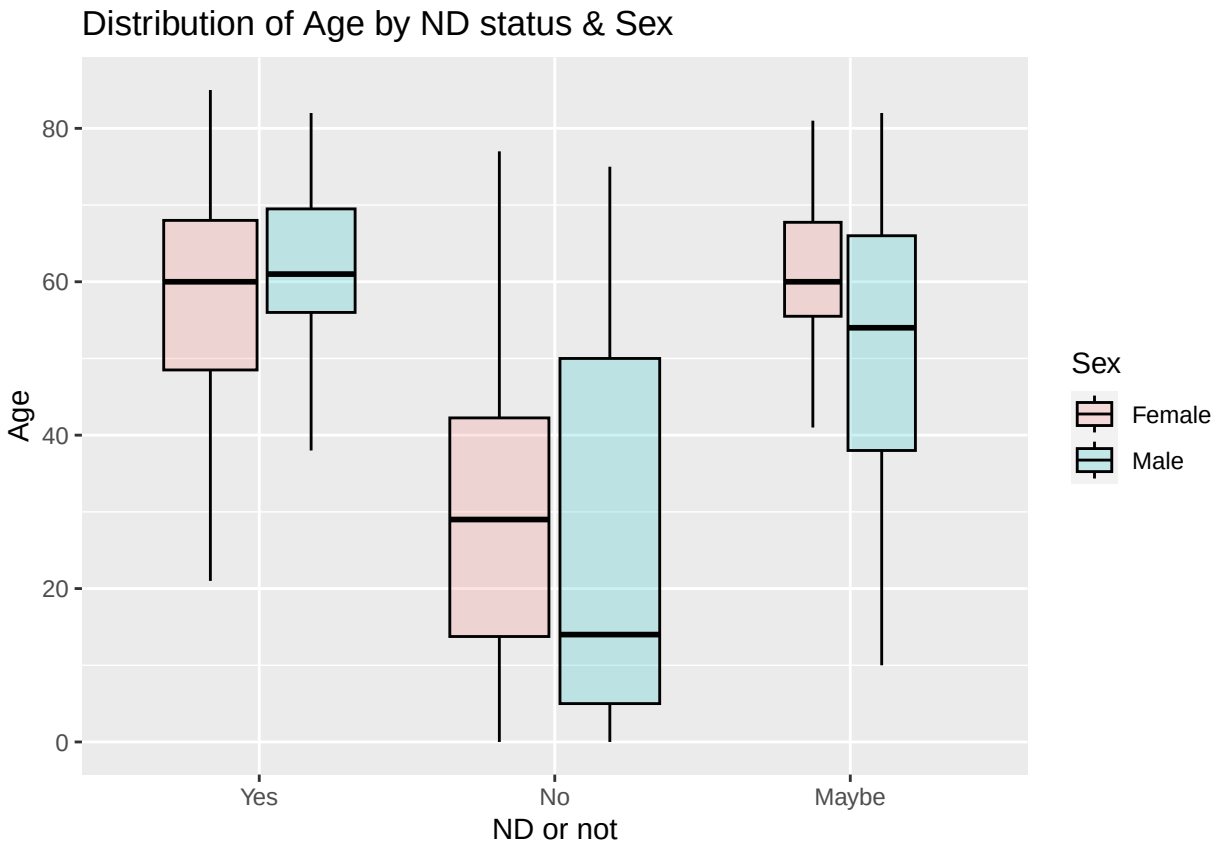
```
gamobj<-gam(newND~lo(age, span=1/3),family=binomial, data=persons)
 #In the above, smooth with a window of 1/3 of the data (the default is 1/2).

plot(gamobj, se=T)
```




```
#linear looks fine (a straight line can fit between the 1-se error bars)
```

Now, let us move one step ahead and plot the age in relation to both ND status and the gender of the person.

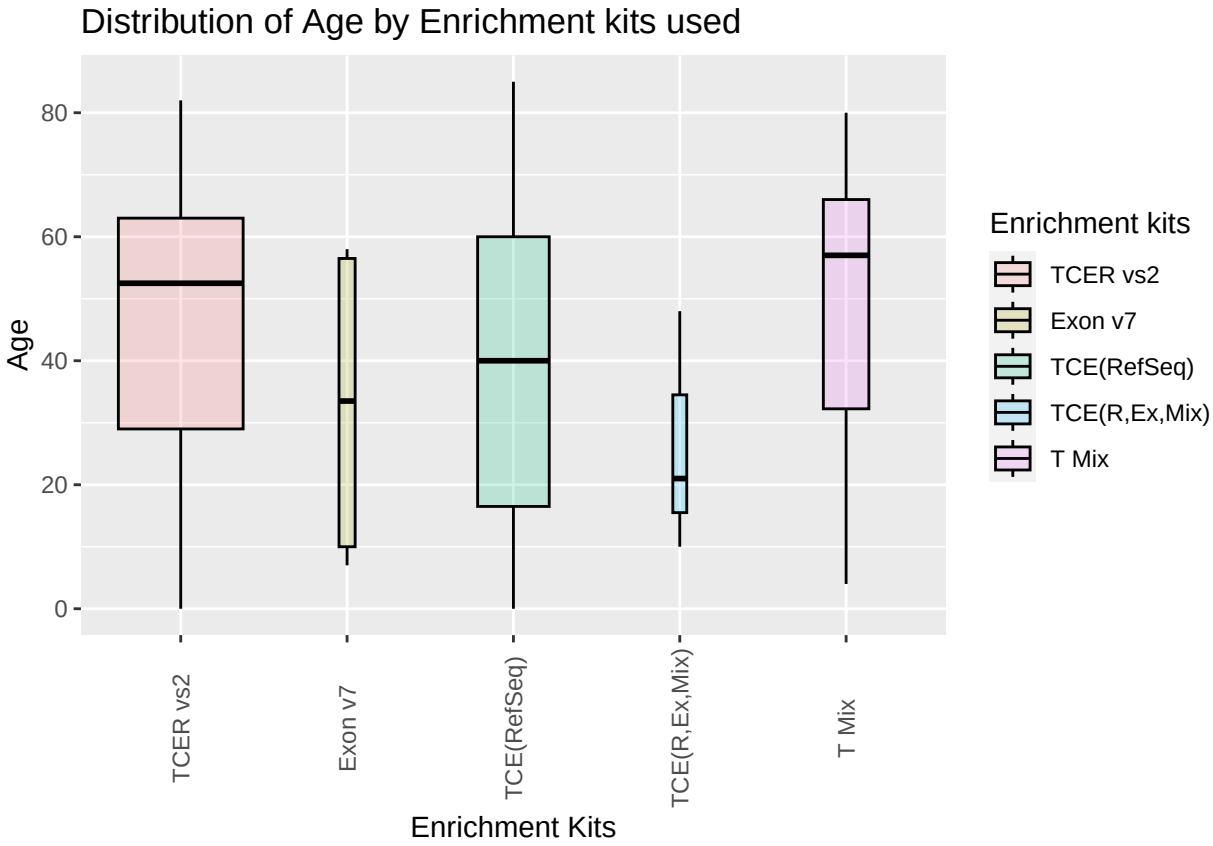
```
#Age by ND status & gender
ggplot(persons, aes(x =ND, y = age , fill=Sex)) +
  geom_boxplot(color = "black", outlier.shape = NA ,varwidth = TRUE, alpha=0.2) +  labs(x = "ND or not"
  ggtitle("Distribution of Age by ND status & Sex")
```



Here again, we see the same trend that people with a ND disease have a higher average age than those with non-ND disease. Plotting the two genders, we can see that the average age for males and females is quite similar for the case of a ND disease but for a non-ND disease, males have slightly lesser average age than females. A bootstrap test on this also would be of great help to see if the gender is associated with the ND status. It would help us look at the real confounding variables so that we can adjust for them beforehand in order to be confident with our results.

Now, let us look at the distribution of age by the E kits used.

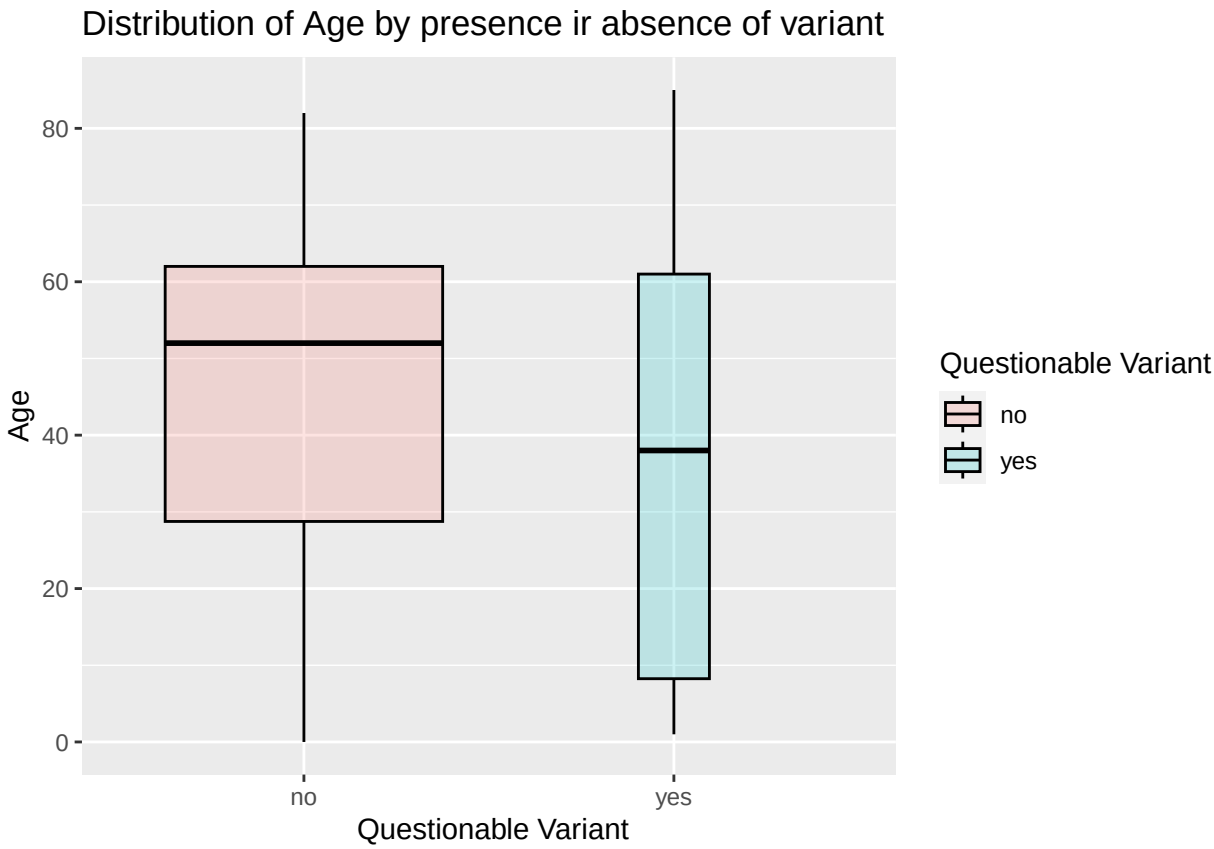
```
#Age by enrichment_kits
ggplot(persons, aes(x =`Enrichment kits`, y = age, fill=`Enrichment kits`)) + geom_boxplot(color = "bla
  labs(x = "Enrichment Kits", y = "Age") +
  ggtitle("Distribution of Age by Enrichment kits used") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



We can see here that the R vs2 kit was the most frequently used and the age doesn't really determine the type of enrichment kit being used.

```
#Presence or absence of variable with age
ggplot(persons, aes(x =Variant, y = age , fill=Variant)) +
  geom_boxplot(color = "black", outlier.shape = NA ,varwidth = TRUE, alpha=0.2) +  labs(x = "Variant", y
  ggtitle("Distribution of Age by presence or absence of variant")
```


```
#Questionable variable with age
ggplot(persons, aes(x =`Questionable Variant`, y = age , fill=`Questionable Variant`)) +
  geom_boxplot(color = "black", outlier.shape = NA ,varwidth = TRUE, alpha=0.2) +  labs(x = " Questional
  ggtitle("Distribution of Age by presence ir absence of variant")
```



Bootstrap code to test the association between variables.

Let us now write a generic bootstrap code to test the association between a continuous variable such as age and categorical variables such as ND status and clinical information. We start by defining two functions, calc_f for calculating the F statistic and bsF for performing the bootstrap F-test.

```
bsF <- function(cts,cat,B) {
  obs_stat <- calc_f(cts,cat) # calc_f is my own function defined below
  resids <- residuals(aov(cts~cat))
  # Perform bootstrap iterations
  resids_dat <- data.frame(resids=resids,cat=cat)
  bootstrap_stats <- replicate(B, {
    bs_data <- resids_dat %>%
      group_by(cat) %>%
      sample_n(size = n(), replace = TRUE) %>%
      ungroup()
    calc_f(bs_data$resids,bs_data$cat)
  })
  return(mean(bootstrap_stats >= obs_stat))
}

calc_f <- function(cts,cat) {
  fit <- aov(cts~cat) #Use R's built-in anova function to get f-stat
  fit_sum <- summary(fit)
  F_stat <- fit_sum[[1]]$`F value`[1] #Extract the f-stat value
  return(F_stat)
}
```

Let's try out the bootstrap F test function to test whether age and ND status are associated.

```
B= 1000
bsF(persons$age,persons$ND,B)
```

```
## [1] 0
```

We can see that they are, backing up the strong visual impression we get from the box plots!

Let us now use our bootstrap code to test the association between age and clinical information

```
bsF(persons$age,persons$`Clinical information`,B)
```

```
## [1] 0
```

This extremely low p-value (0) suggests that the observed differences in age across categories of clinical information are clearly unlikely to have occurred by random chance alone. Therefore, it can be concluded that there is a significant association between age and the clinical information variable, indicating that age varies systematically across the different clinical information.

#Sex~age

```
bsF(persons$age , persons$Sex,B)
```

```
## [1] 0.623
```

This large value exactly resonates with the results that the boxplot was giving us i.e. the data does not provide strong evidence to conclude that age varies significantly based on gender. The high p-value indicates that any observed differences in age across gender categories are likely due to random chance rather than a systematic effect of gender. Therefore, based on the results of the 1000 bootstrap sampling, it is reasonable to conclude that age and gender are not strongly associated in the data. We can say that any differences in age between male and female individuals may not be meaningful from a statistical standpoint.

#Variant~ age

```
bsF(persons$age,persons$Variant,B)
```

```
## [1] 0.282
```

#Enrichment kits~ age

```
bsF(persons$age, persons$`Enrichment kits`,B)
```

```
## [1] 0.038
```

#Questionable Variant ~ age

```
bsF(persons$age,persons$`Questionable Variant`,B)
```

```
## [1] 0.204
```

We can see from the above plot that there is a large majority of patients of ALS, which is ND disease. An interesting finding here is about the two sample indexes DIV and DYT. Here, half of the times, the two indexes are classified as non-ND but the other half of the times, we see that due to some overlapping characteristics with the ND disease, we are unable to classify them and hence they come under the 'Maybe' category.

Variants Data File- Exploratory Analysis

Now read in the data from Joanna's excel file `ATXN2variants.xlsx`. First let's look at the names of the sheets in this excel file

```
excel_sheets("../ATXN2variants.xlsx")
```

```
## [1] "info"   "Sheet1"
```

The name of the sheet with the data is `Sheet1`, so let's read it into R.

```
variants<-read_excel("../ATXN2variants.xlsx", sheet="Sheet1")
View(variants)
```

Adding a new column to our data frame to classify the entries in yellow. As we mention that according to Joanna, these are Questionable variants and might be faulty to use them. So, we create a new column named .

```
variants$'Questionable variant' <- ifelse(
 variants$SampleNumber%in% c(120, 182, 15, 96, 236, 120, 131, 128, 182, 17, 137, 48),
  "yes",
  "no"
)
```

On the same lines, the way we did uni variate summaries for the variables in the persons file, we will now do similar analysis for the variants file. Lets look at the different types of variants using the func. column.

```
Func <- variants%>%
  group_by(`Func`) %>%
  summarise(count = n())
print(Func)
```

```
## # A tibble: 5 x 2
##   Func                                                  count
##   <chr>                                                 <int>
## 1 disruptive_inframe_deletion                               5
## 2 disruptive_inframe_insertion, direct_tandem_duplication  14
## 3 frameshift_elongation                                     1
## 4 frameshift_truncation                                    15
## 5 frameshift_variant                                       18
```

As Joanna mentions that there are 5 types which could potentially be used as variable in place of the actual cDNA variant (i.e. instead of checking for enrichment of each variant we could check for enrichment of variants of specific type). The most frequent one is 'frameshift_variant' appearing 18 times.

Moving forward,if we want to look at the LocalFound variable, a simple univariate summary won't give us a lot of insight as LocalFound is already a count in itself i.e. the number of times the variant was found in the local database, hence, we will look at the two columns(cDNA- that tells us about the different variants and the corresponding LocalFound)

```
# Extract the two columns cDNA and LocalFound
LocalFound <- variants[, c("cDNA", "LocalFound")]

# Keeping only the unique values
unique_cDNA <- unique(LocalFound)
print(unique_cDNA)
```

```
## # A tibble: 19 x 2
##    cDNA                              LocalFound
##    <chr>                                  <dbl>
##  1 c.176_190dup                               1
##  2 c.39_40del                                 8
##  3 c.42_58del                                 5
##  4 c.42del                                   10
##  5 c.48_58del                                 4
##  6 c.51_56dup                                 3
##  7 c.54_58del                                 1
```

```
##  8 c.56_57insGC                                2
##  9 c.57_58insG                                 1
## 10 c.57_59del                                  1
## 11 c.57_80del                                  1
## 12 c.60del                                    11
## 13 c.65_66insACAGCA                            1
## 14 c.68_85del                                  1
## 15 c.71_72insACAGCAGCAGCA                     13
## 16 c.71_72insACAGCAGCAGCAGCAGCA                2
## 17 c.74_75insACAGCAGCAGCAGCA                   1
## 18 c.80_85del                                  3
## 19 c.83_84insACAGCAGCAGCAGCAGCAGCAGCA          1
```

```
unique_cDNA <- unique_cDNA %>%
  arrange(LocalFound) %>% filter(LocalFound >1)
print(unique_cDNA)
```

```
## # A tibble: 10 x 2
##    cDNA                          LocalFound
##    <chr>                              <dbl>
##  1 c.56_57insGC                           2
##  2 c.71_72insACAGCAGCAGCAGCAGCA           2
##  3 c.51_56dup                             3
##  4 c.80_85del                             3
##  5 c.48_58del                             4
##  6 c.42_58del                             5
##  7 c.39_40del                             8
##  8 c.42del                               10
##  9 c.60del                               11
## 10 c.71_72insACAGCAGCAGCA                13
```

We can see that here, we have 19 different variants. This also helps us verify that our initial consideration that we have 19 different variants has been cross verifies. The arrangement shows us that the most frequent variants is "c.71_72insACAGCAGCAGCA" that was found 13 times and "c.60del" that was found 11 times in the local database.

Joint Exploratory Analysis

```
persons_with_var <- persons %>%
  filter(`ATXN2ex1 variant1` != "neg" | `ATXN2ex1 variant2` != "neg")
```

We can see that, out of the total 358 samples persons, there were 37 who had a variant.

Let us see if the information in the "cDNA" column of the variants file match the information in the "ATXN2ex1 variant1" and "ATXN2ex1 variant2" columns of the persons file for these persons who had a variant.

For this, we have the new data framed named persons_with_var, this df eliminates all the entries of the persons df that do not have any variant i.e. the entry in both ATXN2ex1 var1 and ATXN2ex1 var2 are negative.

Now that we have this df of persons with variants. For each row, we will see the corresponding entry in the variants data frame that has the same SampleIndex and same Sample Number. We will then compare the variant that they have to the cDNA column in the corresponding row of the variants data file.

```
# Loop through each row in persons
for (i in 1:nrow(persons_with_var)) {
```

```
  row1 <- persons_with_var[i, ]
  # Loop through each row in variants
  for (j in 1:nrow(variants)) {
    row2 <- variants[j, ]

    # Check if the values of SampleIndex and SamleNumber match
    if (row1$SampleIndex == row2$SampleIndex && row1$SampleNumber == row2$SampleNumber) {
      #When the entry matches (i.e. the person is the same), we check if the entry in the ATXN2ex1 vari
      if (row1$`ATXN2ex1 variant1` == row2$cDNA) {
        # If cDNA matches variant, store the match in a new row in the persons df.
    persons_with_var$Match = "True"
      }
    }
  }
}

View(persons_with_var)
```

We can clearly see in the newly added column that for each row the entry in the Match column comes out to be true. With this test, we can be assured that there in no discrepancy in the information of variants in the two excel file. The entries are the same for the two files.

As Joanna mentions, we had some questionable variants(the variable indicating the samples highlighted in yellow in the variants file). When we started looking at the excel file and imported the data, we created a new column in our data set for all the samples that contained any of these questionable variants, the column was named as .

Let us try to summarize the variants and see if we have any trend.

```
variants %>% filter(`Questionable variant`=="yes") %>% group_by(cDNA) %>%  count(cDNA)
```

```
## # A tibble: 3 x 2
## # Groups:   cDNA [3]
##   cDNA           n
##   <chr>      <int>
## 1 c.39_40del     7
## 2 c.42del        9
## 3 c.80_85del     2
```

So, here we see that all the questionable variants have one one of these 3 cDNAs- c.39_40del, c.42del, c.80_85del

Now, lets create our new data with all the unique cDNAs.

```
variants_new = left_join(variants, persons, by= "SampleIndex") %>%
  distinct(SampleIndex, ND)
```

```
## Warning in left_join(variants, persons, by = "SampleIndex"): Detected an unexpected many-to-many rela
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 9 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
variants_new = merge(variants, variants_new, by = "SampleIndex")

variants_new <- variants_new %>%
  select(-c(SampleIndex, SampleNumber)) %>%
```

```
  select(cDNA, everything())
variants_new <-variants_new %>%
  distinct(cDNA, .keep_all = TRUE)
View(variants_new)
print(variants_new)
```

```
##                                  cDNA     Start       End  Gene   OMIM
## 1                       c.176_190dup 112036648 112036648 ATXN2 601517
## 2                         c.39_40del 112036798 112036800 ATXN2 601517
## 3                            c.60del 112036778 112036779 ATXN2 601517
## 4                         c.57_80del 112036758 112036782 ATXN2 601517
## 5                         c.80_85del 112036753 112036759 ATXN2 601517
## 6                         c.42_58del 112036780 112036797 ATXN2 601517
## 7                            c.42del 112036796 112036797 ATXN2 601517
## 8           c.74_75insACAGCAGCAGCAGCA 112036764 112036764 ATXN2 601517
## 9        c.71_72insACAGCAGCAGCAGCAGCA 112036767 112036767 ATXN2 601517
## 10 c.83_84insACAGCAGCAGCAGCAGCAGCAGCA 112036755 112036755 ATXN2 601517
## 11                   c.65_66insACAGCA 112036773 112036773 ATXN2 601517
## 12             c.71_72insACAGCAGCAGCA 112036767 112036767 ATXN2 601517
## 13                         c.57_58insG 112036781 112036781 ATXN2 601517
## 14                        c.68_85del 112036753 112036771 ATXN2 601517
## 15                       c.56_57insGC 112036782 112036782 ATXN2 601517
## 16                        c.54_58del 112036780 112036785 ATXN2 601517
## 17                        c.57_59del 112036779 112036782 ATXN2 601517
## 18                        c.48_58del 112036780 112036791 ATXN2 601517
## 19                        c.51_56dup 112036782 112036782 ATXN2 601517
##              AAChange                                                Func
## 1  p.(Val59_Ser63dup) disruptive_inframe_insertion, direct_tandem_duplication
## 2   p.(Gln14Alafs*75)                                  frameshift_truncation
## 3   p.(Gln20Hisfs*26)                                     frameshift_variant
## 4  p.(Gln21_Gln28del)                           disruptive_inframe_deletion
## 5  p.(Gln27_Gln28del)                           disruptive_inframe_deletion
## 6   p.(Gln15Alafs*69)                                  frameshift_truncation
## 7   p.(Gln14Hisfs*32)                                     frameshift_variant
## 8  p.(Gln24_Gln28dup) disruptive_inframe_insertion, direct_tandem_duplication
## 9  p.(Gln23_Gln28dup) disruptive_inframe_insertion, direct_tandem_duplication
## 10 p.(Gln21_Gln28dup) disruptive_inframe_insertion, direct_tandem_duplication
## 11 p.(Gln27_Gln28dup) disruptive_inframe_insertion, direct_tandem_duplication
## 12 p.(Gln25_Gln28dup) disruptive_inframe_insertion, direct_tandem_duplication
## 13  p.(Gln20Alafs*70)                                     frameshift_variant
## 14 p.(Gln23_Gln28del)                           disruptive_inframe_deletion
## 15  p.(Gln20Hisfs*27)                                  frameshift_elongation
## 16  p.(Gln19Alafs*69)                                  frameshift_truncation
## 17       p.(Gln28del)                           disruptive_inframe_deletion
## 18  p.(Gln17Alafs*69)                                  frameshift_truncation
## 19 p.(Gln27_Gln28dup) disruptive_inframe_insertion, direct_tandem_duplication
##    LocalFound  GnomVariantFrequency GnomAlleleNumber Questionable variant    ND
## 1           1 2.5139999999999999E-4           111366                   no   Yes
## 2           8             1.506E-3               NA                  yes   Yes
## 3          11 3.6982999999999999E-3           141686                   no   Yes
## 4           1 5.2320000000000003E-4           135706                   no   Yes
## 5           3 4.0030000000000003E-4               NA                  yes   Yes
## 6           5 1.3408999999999999E-3           140202                   no   Yes
## 7          10 2.7476000000000002E-3               NA                  yes   Yes
```

```
## 8           1              4.751E-4          141018                   no   Yes
## 9           2              1.062E-3          141240                   no   Yes
## 10          1 1.8980000000000001E-4          126468                   no   Yes
## 11          1 5.9559999999999995E-4          142710                   no   Yes
## 12         13 7.4625000000000004E-3          141240                   no Maybe
## 13          1 5.4049999999999996E-4          140606                   no   Yes
## 14          1         not in GnomAD           72282                   no   Yes
## 15          2 4.0840000000000001E-4           26932                   no   Yes
## 16          1 4.9930000000000005E-4          140202                   no Maybe
## 17          1 2.2796000000000001E-3          133358                   no Maybe
## 18          4 8.7020000000000001E-4          140202                   no   Yes
## 19          3         not in GnomAD           26932                   no    No
```

We will start by summarizing the func variable.

```
func <- variants_new %>%
  group_by(Func) %>%
  summarize(UniqueVariantsCount = n_distinct(cDNA))
print(func)
```

```
## # A tibble: 5 x 2
##   Func                                                  UniqueVariantsCount
##   <chr>                                                               <int>
## 1 disruptive_inframe_deletion                                             4
## 2 disruptive_inframe_insertion, direct_tandem_duplication                 7
## 3 frameshift_elongation                                                   1
## 4 frameshift_truncation                                                   4
## 5 frameshift_variant                                                      3
```

Of the 19 variants, we can see the distribution over the func variable where 7 of them have 'disruptive_inframe_insertion, direct_tandem_duplication' followed by 4 of them having 'disruptive_inframe_deletion' and 'frameshift_truncation' respectively.

```
contingency_table_v1<- table(variants_new$cDNA,variants_new$ND)
print(contingency_table_v1)
```

```
##
##                                    Yes No Maybe
##   c.176_190dup                       1  0     0
##   c.39_40del                         1  0     0
##   c.42_58del                         1  0     0
##   c.42del                            1  0     0
##   c.48_58del                         1  0     0
##   c.51_56dup                         0  1     0
##   c.54_58del                         0  0     1
##   c.56_57insGC                       1  0     0
##   c.57_58insG                        1  0     0
##   c.57_59del                         0  0     1
##   c.57_80del                         1  0     0
##   c.60del                            1  0     0
##   c.65_66insACAGCA                   1  0     0
##   c.68_85del                         1  0     0
##   c.71_72insACAGCAGCAGCA             0  0     1
##   c.71_72insACAGCAGCAGCAGCAGCA       1  0     0
##   c.74_75insACAGCAGCAGCAGCA          1  0     0
##   c.80_85del                         1  0     0
```

```
##   c.83_84insACAGCAGCAGCAGCAGCAGCAGCA   1  0     0
```

Jotting this table gives us some important insights like - 1. The variants <c.57_59del> and <c.54_58del> giving rise to a type of disease where we can not identify if it is ND or Non-ND because of some overlapping characteristics. 2. Out of the 19 variants, 15 of them give rise to a disease that is ND.

```
result <- fisher.test(contingency_table_v1, simulate.p.value = TRUE, B = 10000)
p_value <- result$p.value
print(p_value)
```

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

To look at the ND status of the questionable variants, lets see the contingency table with ND status of these variants.

```
subset_data <- variants_new[variants_new$`Questionable variant` == "yes", ]
contingency_table_v2 <- table(subset_data$cDNA, subset_data$ND)
print(contingency_table_v2)
```

```
##
##              Yes No Maybe
##   c.39_40del   1  0     0
##   c.42del      1  0     0
##   c.80_85del   1  0     0
```

```
rm(subset_data)
```

So, we see that all these 3 variants give rise to a disease that is ND.

Modifying and updating the persons and the variants data frames to make them fit for our analysis.

Let us first update our "persons" data file to use for our further analysis. For this, we will add new columns to our data frame.Here, the additional columns would be each of the different variants detected.

For this, let us first verify if any variant is appearing twice in any person or not

```
duplicate_rows <- persons %>% filter(`ATXN2ex1 variant1` == `ATXN2ex1 variant2` & `ATXN2ex1 variant1` !=
rm(duplicate_rows)
```

We can see that there is no data in this table. Hence, we can be assured that there is no such person for which the same variant is detected twice.

```
#Create the new variants data frame.
variants_new <- variants %>%
  select(3:13) %>%
  filter(!duplicated(`cDNA`)) %>%
  arrange(`cDNA`) %>%
  select(`cDNA`, everything())
```

#skat function -study

#For questionable var. add missing in the variant status. #Ques Var are not Associated with ND Status. #Associated with p/a of a variant.

#Data- Pathway to figure out what we are gonna do in the formal analysis. #About the trio patients- #How do we choose if we take in parents or the children in our final analysis data. #We look at the association between age and ND status #Skat function in the skat package # DATA matrix- #Rows- people #Columns- Variables we want to adjust for(age and enrichment kits) #Columns for each different variable