

Analysis_Rmarkdown

ChloeYS

2024-05-28

More details on the code can be found on https://github.com/ChloeYS/ASynuclein_APD
https://github.com/ChloeYS/ASynuclein_APD (most updated version is kept there)

1. SOURCE PACKAGES AND FUNCTIONS

Source functions created for this analysis. Source the function to QC the data.

```
source('Functions.R') #Source the functions from file in same repository (for now)
```

```
## — Attaching core tidyverse packages ————— tidyverse 2.0.0 —
## ✓ dplyr     1.1.4    ✓ readr     2.1.5
## ✓ forcats   1.0.0    ✓ stringr   1.5.1
## ✓ ggplot2   3.5.0    ✓ tibble    3.2.1
## ✓ lubridate 1.9.3    ✓ tidyrr    1.3.1
## ✓ purrr    1.0.2
## — Conflicts ————— tidyverse_conflicts() —
## ✘ dplyr::filter() masks stats::filter()
## ✘ dplyr::lag()   masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

#tidyverse() loaded in Functions.R

```
source('DataQC_PROCESSING.R') #Source the function taht creates the new dataframe on which analysis is performed
#lubridate() loaded in DataQC_PROCESSING.R
```

Load libraries

```
library(car) #very useful for Anova(), Levene, etc. Please note that base R anova() and car::Anova() differ for:
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

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

```
## The following object is masked from 'package:purrr':  
##  
##      some  
  
##contrast option already set as c("contr.sum", "contr.poly") for car::Anova(). Also,  
##car::Anova() does type II SS  
##method as default, whereas base R anova() does type I SS method (same results as s  
ummary(aov))  
library(emmeans) #estimated marginal means of a model  
library(ggfortify) #plots for QC model  
library(caret)
```

```
## Loading required package: lattice
```

```
##  
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':  
##  
##      lift
```

```
library(ggplot2) #plot figures with stats  
library(ggpubr) #ggscatter  
library(ggsurvfit) #survival analysis  
library(survival) #survival analysis: survdiff()
```

```
##  
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':  
##  
##      cluster
```

```
library(survminer) #ggsurvplot
```

```
##  
## Attaching package: 'survminer'
```

```
## The following object is masked from 'package:survival':  
##  
##      myeloma
```

```
library(performance) #check_normality  
library(rcompanion) #Cramer's V  
library(janitor) #remove empty rows
```

```
##  
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':  
##  
##     chisq.test, fisher.test
```

```
library(nlme) #weighted least square regression
```

```
##  
## Attaching package: 'nlme'
```

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

```
library(lmtest) #bptest()
```

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

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

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

```
library(fmsb) #create_beautiful_radarchart()
```

```
## Registered S3 methods overwritten by 'fmsb':  
##   method    from  
##   print.roc pROC  
##   plot.roc pROC
```

```
library(pscl) #McFadden index
```

```
## Classes and Methods for R originally developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University (2002–2015),
## by and under the direction of Simon Jackman.
## hurdle and zeroinfl functions by Achim Zeileis.
```

```
library(lsr) #eta-squared aov()
library(ggsignif) #annotate stats on figures
```

2. DATAFRAME LOADING

Process raw data to get the dataframe prepped for analysis

```
df <- read.csv("/Users/nikhilbhagwat/Desktop/7_PUBLICATIONS_ONGOING/7.1_APD_MS_2024-01/
7.1.0_Data/APD_Neurology_MS_data.csv", 'df')
```

```
## The file: /Users/nikhilbhagwat/Desktop/7_PUBLICATIONS_ONGOING/7.1_APD_MS_2024-01/7.1.
0_Data/APD_Neurology_MS_data.csv is read into the dataframe: df
```

```
df <- var.func.1(df) #the df for analysis is created
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## This is not a warning. as.numeric() call:  
## Check potential issues when using cbind:  
## 1- duplicate column names;  
## 2- different row numbers, etc.  
## [1] 67  
## [1] 67  
## [1] 67  
## [1] "ID"  
## [2] "Include_APDs"  
## [3] "Name"  
## [4] "Clinic_only"  
## [5] "Sex"  
## [6] "DOB_dd.mmmm.yy"  
## [7] "ID_Date_dd.mmmm.yy"  
## [8] "RTQUIC_2_Date_dd.mmmm.yy"  
## [9] "First_Visit_Date_dd.mmmm.yy"  
## [10] "Last_Visit_Date_dd.mmmm.yy"  
## [11] "ID_Age"  
## [12] "PPA"  
## [13] "Onset_type"  
## [14] "DX_Lifetime"  
## [15] "DX_Lifetime_criteria"  
## [16] "DX_Jabbari"  
## [17] "DX_APD"  
## [18] "Race_ethnicity"  
## [19] "RTQUIC_lifetime"  
## [20] "RTQUIC_conversion"  
## [21] "AD_lifetime ATHENA"  
## [22] "AD_lifetime_notes"  
## [23] "AD_2 ATHENA"  
## [24] "AD_2_notes"  
## [25] "YKL40_2"  
## [26] "LP2_auton_signs"  
## [27] "LP2_auton_signs_n"  
## [28] "LP2_Light_Sensitivity"  
## [29] "LP2_Dysphagia"  
## [30] "LP2_Sexual_dysfunction"  
## [31] "LP2_Constipation"  
## [32] "LP2_Urinary"  
## [33] "LP2_Hyperhidrosis_Thermoregulatory_plus_other"  
## [34] "LP2_Orthostatism_plus_other"  
## [35] "LP2_Bowel_Incontinence"  
## [36] "LP2_RBD_plus_other"  
## [37] "LP2_Anosmia"  
## [38] "Genetics_FamilyHistory"  
## [39] "APOEe4_alleles"  
## [40] "LP2_MOCA_1year"  
## [41] "LP2_Cognitive_Extra"  
## [42] "LP2_MOCA_total_corrected"  
## [43] "LP2_Delusions"  
## [44] "Lifetime_Hallucinations"  
## [45] "LP2_gait"
```

```
## [46] "LP2_falls_PI"
## [47] "LP2_hypomimia"
## [48] "LP2_retropulsion"
## [49] "LP2_tremor"
## [50] "LP2_slowness"
## [51] "LP2_oculomotor"
## [52] "Lifetime_oculomotor"
## [53] "LP2_rigidity"
## [54] "LP2_dystonia"
## [55] "LP2_apraxia"
## [56] "Lifetime_apraxia"
## [57] "LP2_myoclonus"
## [58] "LP2_alien_limb"
## [59] "LP2_PSPRS"
## [60] "Medication_10_Mov"
## [61] "Lifetime_Dopa"
## [62] "Lifetime_Dopa_responder"
## [63] "Lifetime_Dopa_responder_true"
## [64] "X"
## [65] "X.1"
## [66] "X.2"
## [67] "X.3"
## [68] "X.4"
## [69] "X.5"
## [70] "X.6"
## [71] "X.7"
## [72] "X.8"
## [73] "X.9"
## [74] "X.10"
## [75] "X.11"
## [76] "X.12"
## [77] "X.13"
## [78] "X.14"
## [79] "X.15"
## [80] "X.16"
## [81] "X.17"
## [82] "X.18"
## [83] "X.19"
## [84] "X.20"
## [85] "X.21"
## [86] "X.22"
## [87] "X.23"
## [88] "X.24"
## [89] "X.25"
## [90] "X.26"
## [91] "X.27"
## [92] "X.28"
## [93] "DOB"
## [94] "Date"
## [95] "First_Visit"
## [96] "Last_Visit"
## [97] "Education"
```

```
## [98] "ID_Age_TXT"
## [99] "Onset_age"
## [100] "Park_onset"
## [101] "ptau_2"
## [102] "ttau_2"
## [103] "abeta_2"
## [104] "ATI_2"
## [105] "NFL_2"
## [106] "LP2_Cognitive_Z.score"
## [107] "LP2_MOCA_Z.score"
## [108] "LP2_MOCA_total"
## [109] "Lag_hours"
## [110] "ThTmax"
## as.factor() call:
## Check potential issues when using cbind:
## 1- duplicate column names;
## 2- different row numbers, etc.
## [1] 67
## [1] 67
## [1] 67
## [1] "ID"
## [2] "Include_APDs"
## [3] "Name"
## [4] "Clinic_only"
## [5] "DOB_dd.mmmm.yy"
## [6] "ID_Date_dd.mmmm.yy"
## [7] "RTQUIC_2_Date_dd.mmmm.yy"
## [8] "First_Visit_Date_dd.mmmm.yy"
## [9] "Last_Visit_Date_dd.mmmm.yy"
## [10] "ID_Age"
## [11] "DX_Lifetime"
## [12] "DX_Lifetime_criteria"
## [13] "DX_Jabbari"
## [14] "RTQUIC_conversion"
## [15] "AD_lifetime_notes"
## [16] "AD_2 ATHENA"
## [17] "AD_2_notes"
## [18] "YKL40_2"
## [19] "LP2_auton_signs_n"
## [20] "LP2_Light_Sensitivity"
## [21] "LP2_Dysphagia"
## [22] "LP2_Hyperhidrosis_Thermoregulatory_plus_other"
## [23] "LP2_Orthostatism_plus_other"
## [24] "Genetics_FamilyHistory"
## [25] "LP2_MOCA_1year"
## [26] "LP2_Cognitive_Extra"
## [27] "LP2_MOCA_total_corrected"
## [28] "LP2_Delusions"
## [29] "Lifetime_Hallucinations"
## [30] "LP2_hypomimia"
## [31] "LP2_dystonia"
## [32] "LP2_myoclonus"
```

```
## [33] "LP2_alien_limb"
## [34] "LP2_PSPRS"
## [35] "Medication_10_Mov"
## [36] "Lifetime_Dopa_responder"
## [37] "X"
## [38] "X.1"
## [39] "X.2"
## [40] "X.3"
## [41] "X.4"
## [42] "X.5"
## [43] "X.6"
## [44] "X.7"
## [45] "X.8"
## [46] "X.9"
## [47] "X.10"
## [48] "X.11"
## [49] "X.12"
## [50] "X.13"
## [51] "X.14"
## [52] "X.15"
## [53] "X.16"
## [54] "X.17"
## [55] "X.18"
## [56] "X.19"
## [57] "X.20"
## [58] "X.21"
## [59] "X.22"
## [60] "X.23"
## [61] "X.24"
## [62] "X.25"
## [63] "X.26"
## [64] "X.27"
## [65] "X.28"
## [66] "DOB"
## [67] "Date"
## [68] "First_Visit"
## [69] "Last_Visit"
## [70] "Education"
## [71] "ID_Age_TXT"
## [72] "Onset_age"
## [73] "Park_onset"
## [74] "ptau_2"
## [75] "ttau_2"
## [76] "abeta_2"
## [77] "ATI_2"
## [78] "NFL_2"
## [79] "LP2_Cognitive_Z.score"
## [80] "LP2_MOCA_Z.score"
## [81] "LP2_MOCA_total"
## [82] "Lag_hours"
## [83] "ThTmax"
## [84] "Age"
```

```
## [85] "Age_Calculation_SanityCheck"
## [86] "Age_cbind_SanityCheck"
## [87] "Followup_duration"
## [88] "LP2_Disease_Duration"
## [89] "LP2_Park_duration"
## [90] "Age_at_last_visit"
## [91] "Duration_last_visit"
## [92] "ID.factor"
## [93] "Sex"
## [94] "PPA"
## [95] "Onset_type"
## [96] "DX_APD"
## [97] "Race_ethnicity"
## [98] "RTQUIC_lifetime"
## [99] "AD_lifetime ATHENA"
## [100] "LP2_auton_signs"
## [101] "LP2_RBD_plus_other"
## [102] "LP2_Anosmia"
## [103] "LP2_Sexual_dysfunction"
## [104] "LP2_Constipation"
## [105] "LP2_Urinary"
## [106] "LP2_Bowel_Incontinence"
## [107] "APOEe4_alleles"
## [108] "LP2_gait"
## [109] "LP2_falls_PI"
## [110] "LP2_retropulsion"
## [111] "LP2_tremor"
## [112] "LP2_slowness"
## [113] "LP2_oculomotor"
## [114] "Lifetime_oculomotor"
## [115] "LP2_rigidity"
## [116] "LP2_apraxia"
## [117] "Lifetime_apraxia"
## [118] "Lifetime_Dopa"
## [119] "Lifetime_Dopa_responder_true"
```

3. SUBSET DF FOR ASSUMPTION TESTING

Create subsets (practical when testing for assumptions of tests)

```
## CREATE SOME OF THE SUBSETS USED LATER (MOSTLY FOR OUTLIER IDENTIFICATION)
CBSdf <- subset(df, DX_APD=="CBS") #for the description of AD+/- cohort
PSPdf <- subset(df, DX_APD=="PSP") #for the description of AD+/- cohort
RTposdf <- subset(df, RTQUIC=="aSyn-SAA positive") #for the main results on RT+/
RTnegdf <- subset(df, RTQUIC=="aSyn-SAA negative") #for the main results on RT+/
ADposdf <- subset(df, AD=="AD Positive") #for the description of AD+/- cohort
ADnegdf <- subset(df, AD=="AD Negative") #for the description of AD+/- cohort
APOEposdf <- subset(df, APOEe4=="Positive") #for the description of APOE+/- cohort
APOEnegdf <- subset(df, APOEe4=="Negative") #for the description of APOE+/- cohort
YODdf <- subset(df, Early_onset=="Young-onset") #for the description of APOE+/- cohort
LODdf <- subset(df, Early_onset=="Late-onset") #for the description of APOE+/- cohort
```

4. COHORT CHARACTERISTICS

Comparisons shown in: Table 1, Results > Cohort characteristics Count per diagnosis:

```
##   DX_APD n
## 1    CBS 39
## 2    PSP 28
```

4.1. COHORT CHARACTERISTICS: NUMERICAL VARIABLES

4.1.1. AGE

Comparisons shown in: Table 1, eTable 1, eTable 2, Results > Cohort characteristics

AGE STATISTICS: DISTRIBUTION

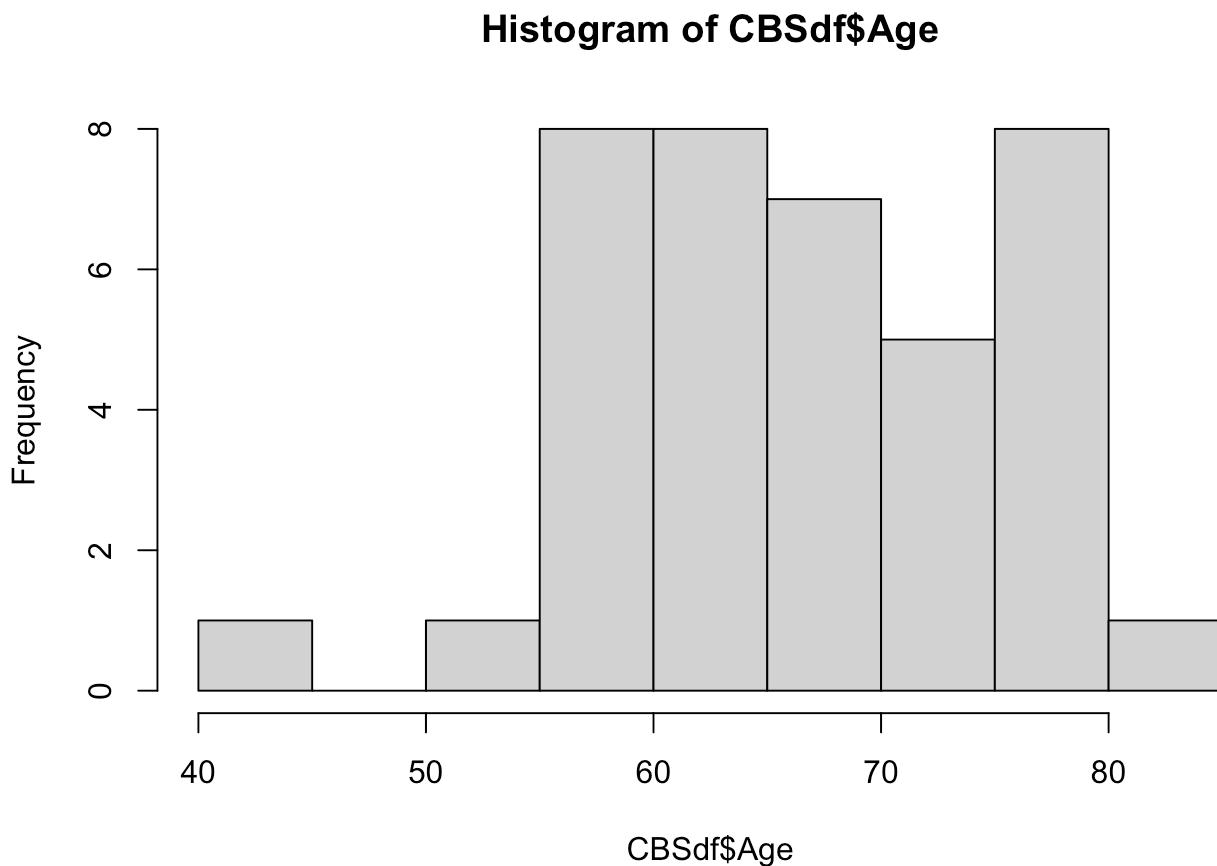
```
shapiro.test(CBSdf$Age) #normal
```

```
##
## Shapiro-Wilk normality test
##
## data: CBSdf$Age
## W = 0.97123, p-value = 0.4092
```

```
shapiro.test(PSPdf$Age) #normal
```

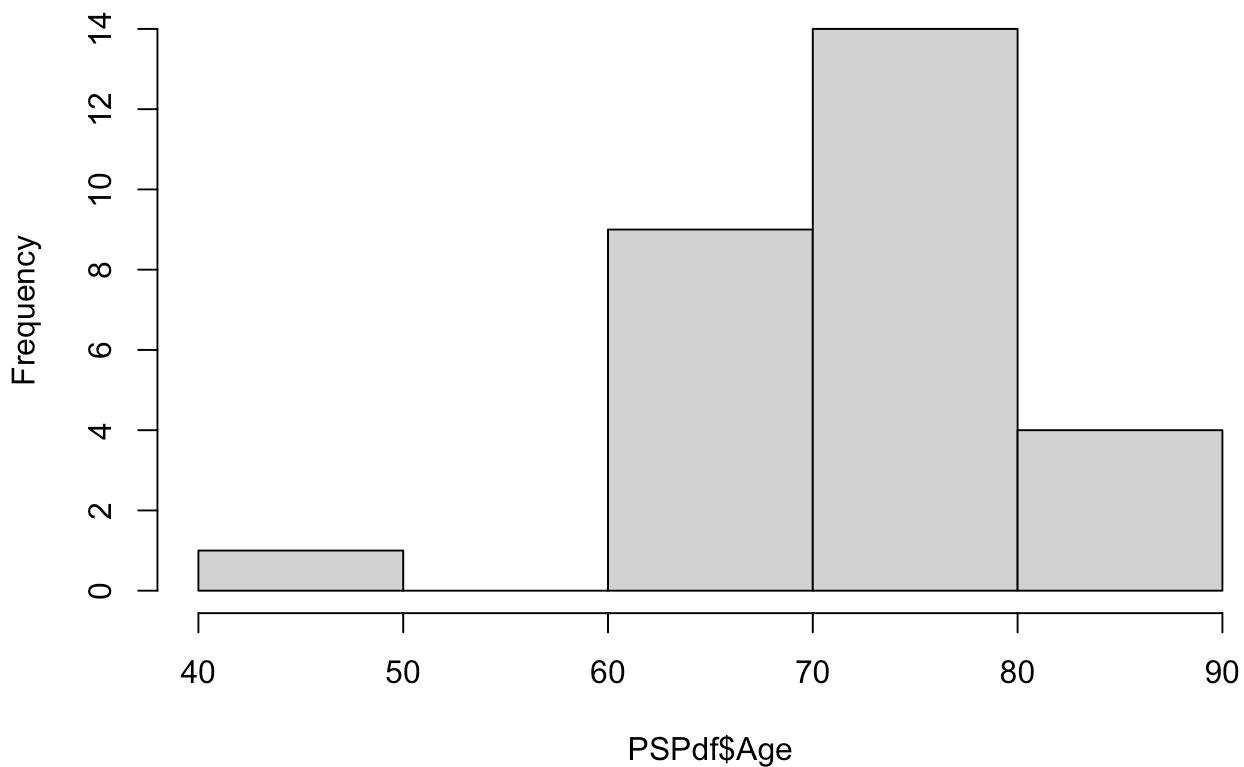
```
##
## Shapiro-Wilk normality test
##
## data: PSPdf$Age
## W = 0.94623, p-value = 0.1591
```

```
hist(CBSdf$Age)
```



```
hist(PSPdf$Age)
```

Histogram of PSPdf\$Age



```
var.test(Age ~ DX_APD, data = df) #homoscedasticity
```

```
## 
## F test to compare two variances
## 
## data: Age by DX_APD
## F = 0.97338, num df = 38, denom df = 27, p-value = 0.924
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.4680448 1.9390187
## sample estimates:
## ratio of variances
## 0.9733823
```

AGE STATISTICS: SUMMARY

```
df %>% group_by(DX_APD) %>% summarize(count=n(), format(round(mean(Age, na.rm=T),2),2),
sd=sd(Age, na.rm=T))
```

```
## # A tibble: 2 × 4
##   DX_APD count `format(round(mean(Age, na.rm = T), 2), 2)`      sd
##   <chr>  <int> <chr>                                         <dbl>
## 1 CBS      39  65.92                                         8.58
## 2 PSP      28  72.51                                         8.70
```

```
df %>% summarize(count=n(), format(round(mean(Age, na.rm=T),2),2), sd=sd(Age, na.rm=T))
```

```
##   count format(round(mean(Age, na.rm = T), 2), 2)      sd
## 1    67                                         68.67 9.171745
```

AGE STATISTICS: STUDENTS TTEST

```
t.test <-t.test(df$Age ~ df$DX_APD, var.equal=TRUE) #var.equal = Student's t-test is performed
if (t.test[3] <= 0.05) {
  cat("There is a significant difference in age at LP between CBS and PSP. p-value:", t.test[3][[1]], "\n")
  cat(t.test[3][[1]])
} else cat("There is no significant difference in age at LP between CBS and PSP.\n")
```

```
## There is a significant difference in age at LP between CBS and PSP. p-value: 0.003007677
## 0.003007677
```

4.1.2. EDUCATION

Comparisons shown in: Table 1

EDUCATION STATISTICS: DISTRIBUTION

```
shapiro.test(CBSdf$Education) #not normal
```

```
##
## Shapiro-Wilk normality test
##
## data: CBSdf$Education
## W = 0.91698, p-value = 0.009044
```

```
shapiro.test(PSPdf$Education) #not normal
```

```
##
## Shapiro-Wilk normality test
##
## data: PSPdf$Education
## W = 0.88686, p-value = 0.00678
```

```
leveneTest(Education ~ DX_APD, data = df) #homoscedasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group  1  0.3371 0.5636
##       62
```

EDUCATION STATISTICS: SUMMARY

```
df %>% summarize(count=n(), format(round(median(Education, na.rm=T),2),2), IQR=IQR(Education, na.rm=T), min=min(Education, na.rm=T), max=max(Education, na.rm=T))
```

```
##   count format(round(median(Education, na.rm = T), 2), 2) IQR min max
## 1     67                      16   6   5   20
```

```
df %>% group_by(DX_APD) %>% summarize(count=n(), format(round(median(Education, na.rm=T),2),2), IQR=IQR(Education, na.rm=T), min=min(Education, na.rm=T), max=max(Education, n.a.rm=T))
```

```
## # A tibble: 2 × 6
##   DX_APD count format(round(median(Education, na.rm = T), 2)...¹ IQR   min   max
##   <chr>  <int> <chr>
## 1 CBS      39  16                      6      5   20
## 2 PSP      28  16                      4.5    7   18
## # i abbreviated name: ¹`format(round(median(Education, na.rm = T), 2), 2)`
```

EDUCATION STATISTICS: WILCOXON

```
wilcox.test(df$Education ~ df$DX_APD, paired=F)
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: df$Education by df$DX_APD
## W = 551.5, p-value = 0.478
## alternative hypothesis: true location shift is not equal to 0
```

4.1.3. ONSET & DURATION

Comparisons shown in: Table 1, eTable 1, eTable 2, Results > Cohort characteristics

ONSET STATISTICS: DISTRIBUTION

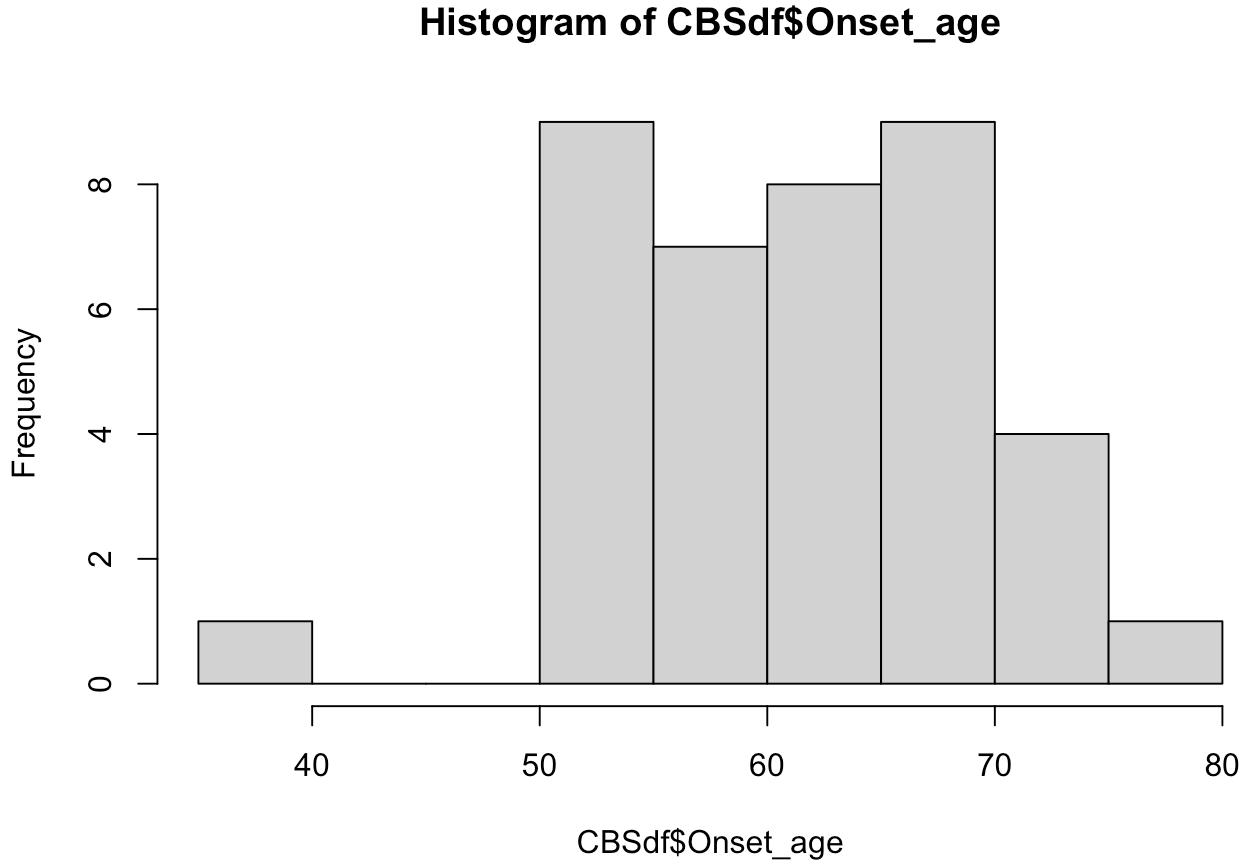
```
shapiro.test(CBSdf$Onset_age) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: CBSdf$Onset_age  
## W = 0.95739, p-value = 0.1458
```

```
shapiro.test(PSPdf$Onset_age) #normal
```

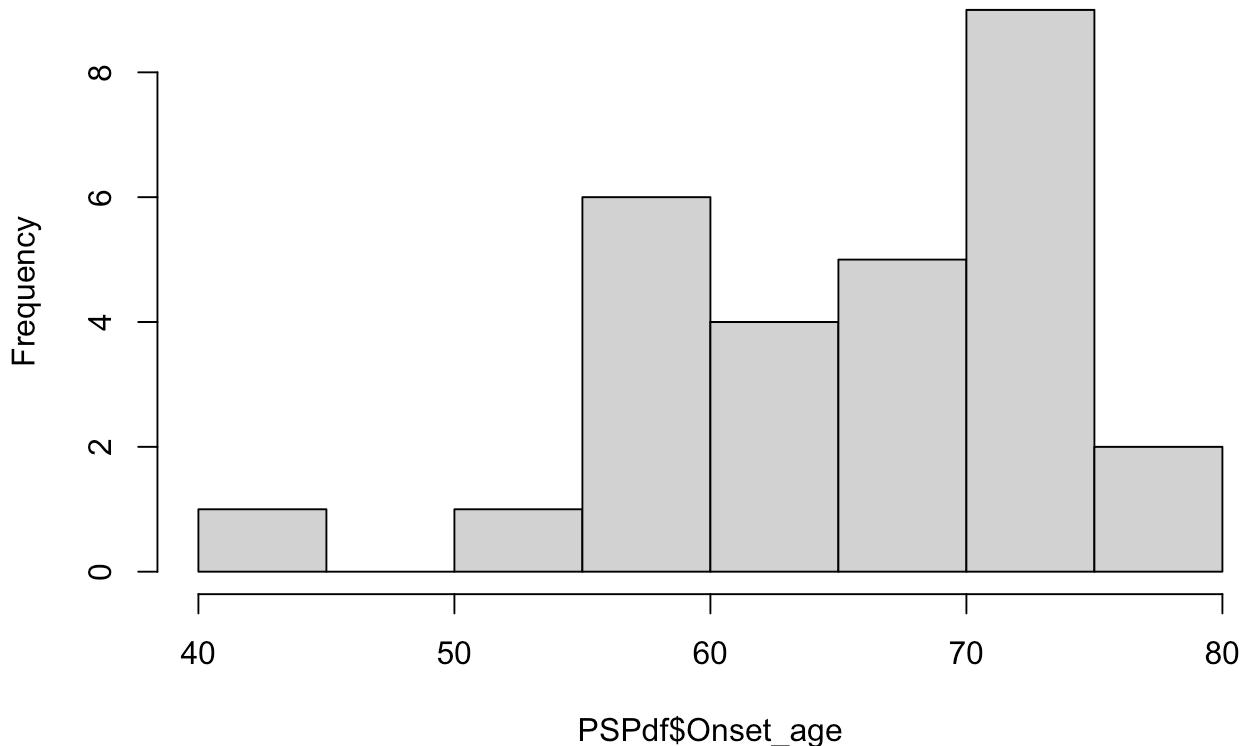
```
##  
## Shapiro-Wilk normality test  
##  
## data: PSPdf$Onset_age  
## W = 0.94965, p-value = 0.1942
```

```
hist(CBSdf$Onset_age)
```



```
hist(PSPdf$Onset_age)
```

Histogram of PSPdf\$Onset_age



```
leveneTest(Onset_age ~ DX_APD, data = df) #homoscedasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group  1     0 0.9954
##       65
```

ONSET STATISTICS: SUMMARY

```
df %>% group_by(DX_APD) %>% summarize(count=n(), format(round(mean(Onset_age, na.rm=T), 2), 2), sd=sd(Onset_age, na.rm=T))
```

```
## # A tibble: 2 × 4
##   DX_APD count `format(round(mean(Onset_age, na.rm = T), 2), 2)`    sd
##   <chr>   <int> <chr>                                         <dbl>
## 1 CBS      39   61.36                                         8.33
## 2 PSP      28   66.29                                         8.33
```

```
df %>% summarize(count=n(), format(round(mean(Onset_age, na.rm=T),2),2), sd=sd(Onset_age, na.rm=T))
```

```
##   count format(round(mean(Onset_age, na.rm = T), 2), 2)      sd
## 1    67                           63.42 8.618425
```

ONSET STATISTICS: STUDENTS TTEST

```
t.test(df$Onset_age ~ df$DX_APD, var.equal=TRUE)
```

```
##
## Two Sample t-test
##
## data: df$Onset_age by df$DX_APD
## t = -2.3887, df = 65, p-value = 0.01982
## alternative hypothesis: true difference in means between group CBS and group PSP is not equal to 0
## 95 percent confidence interval:
## -9.0458769 -0.8076029
## sample estimates:
## mean in group CBS mean in group PSP
##           61.35897           66.28571
```

PARK ONSET STATISTICS: DISTRIBUTION

```
shapiro.test(CBSdf$Park_onset) #borderline
```

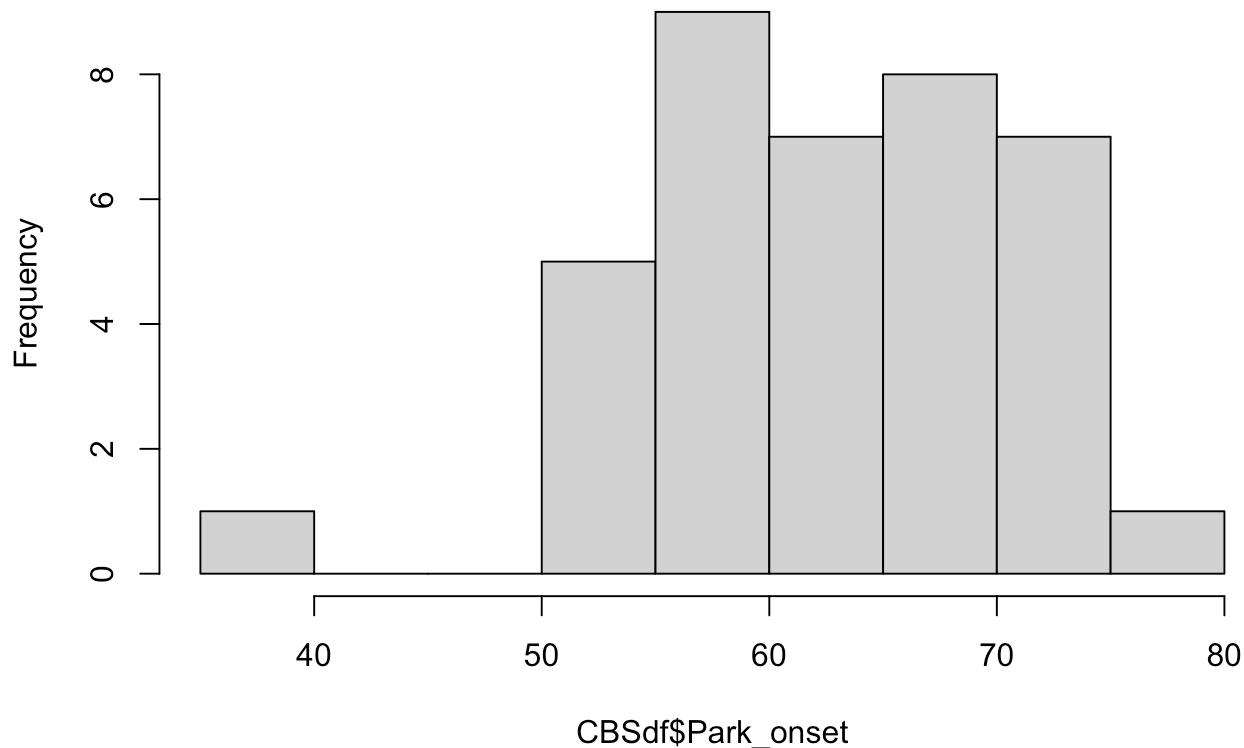
```
##
## Shapiro-Wilk normality test
##
## data: CBSdf$Park_onset
## W = 0.94554, p-value = 0.06351
```

```
shapiro.test(PSPdf$Park_onset) #borderline
```

```
##
## Shapiro-Wilk normality test
##
## data: PSPdf$Park_onset
## W = 0.93769, p-value = 0.09654
```

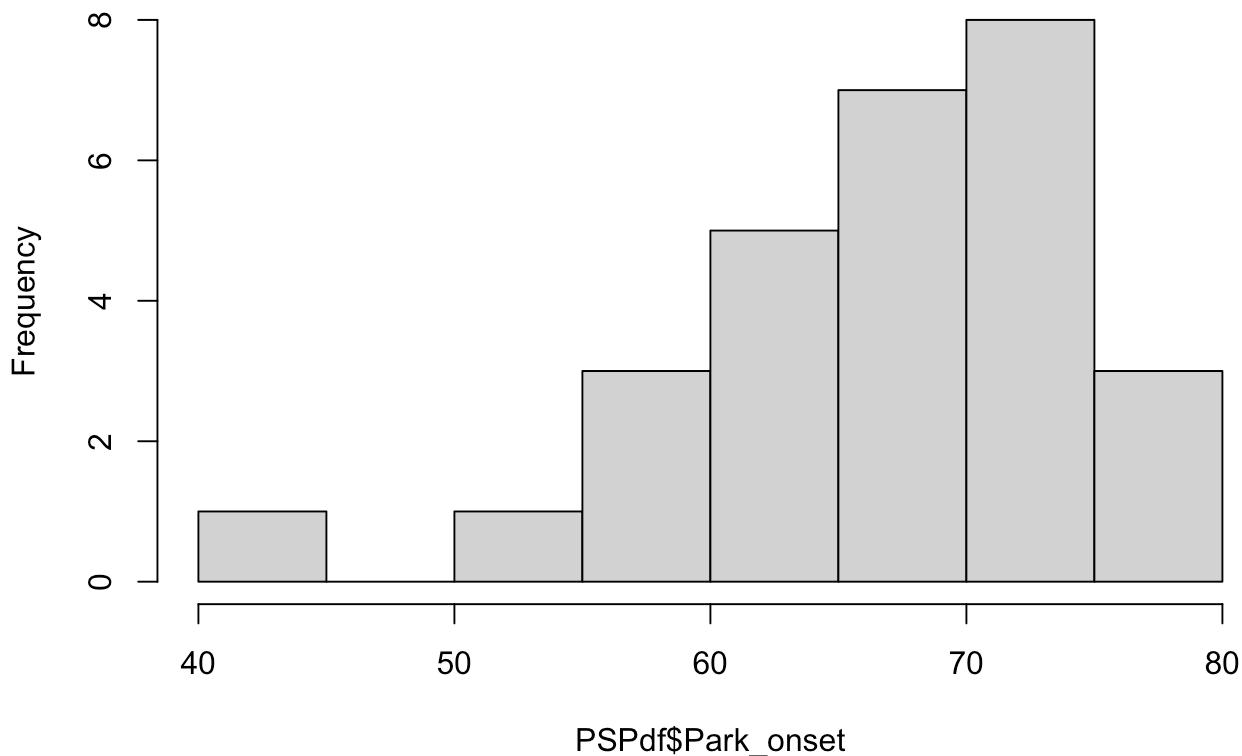
```
hist(CBSdf$Park_onset)
```

Histogram of CBSdf\$Park_onset



```
hist(PSPdf$Park_onset)
```

Histogram of PSPdf\$Park_onset



```
leveneTest(Park_onset ~ DX_APD, data = df) #homoscedasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group  1  0.0483 0.8267
##       64
```

PARK ONSET STATISTICS: SUMMARY

```
df %>% group_by(DX_APD) %>% summarize(count=n(), format(round(mean(Park_onset, na.rm=T), 2), 2), sd=sd(Park_onset, na.rm=T))
```

```
## # A tibble: 2 × 4
##   DX_APD count `format(round(mean(Park_onset, na.rm = T), 2), 2)`    sd
##   <chr>   <int> <chr>                               <dbl>
## 1 CBS      39  63.16                                8.23
## 2 PSP      28  67.14                                8.25
```

```
df %>% summarize(count=n(), format(round(mean(Park_onset, na.rm=T),2),2), sd=sd(Park_onset, na.rm=T))
```

```
##   count format(round(mean(Park_onset, na.rm = T), 2), 2)      sd
## 1    67                           64.85 8.411059
```

PARK ONSET STATISTICS: STUDENTS TTEST

```
t.test(df$Park_onset ~ df$DX_APD, var.equal=TRUE)
```

```
##
## Two Sample t-test
##
## data: df$Park_onset by df$DX_APD
## t = -1.9424, df = 64, p-value = 0.05649
## alternative hypothesis: true difference in means between group CBS and group PSP is not equal to 0
## 95 percent confidence interval:
## -8.0833861 0.1134613
## sample estimates:
## mean in group CBS mean in group PSP
##          63.15789       67.14286
```

DURATION STATISTICS: DISTRIBUTION

```
shapiro.test(CBSdf$LP2_Disease_Duration) #not normal
```

```
##
## Shapiro-Wilk normality test
##
## data: CBSdf$LP2_Disease_Duration
## W = 0.80147, p-value = 9.012e-06
```

```
shapiro.test(PSPdf$LP2_Disease_Duration) #not normal
```

```
##
## Shapiro-Wilk normality test
##
## data: PSPdf$LP2_Disease_Duration
## W = 0.82991, p-value = 0.0003773
```

```
leveneTest(LP2_Disease_Duration ~ DX_APD, data = df) #homoscedasticity but borderline
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  2.9946 0.08829 .
##       65
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

DURATION STATISTICS: SUMMARY

```
df %>% group_by(DX_APD) %>% summarize(count=n(), format(round(median(LP2_Disease_Duration, na.rm=T),2),2), IQR=IQR(LP2_Disease_Duration, na.rm=T), min=min(LP2_Disease_Duration, na.rm=T), max=max(LP2_Disease_Duration, na.rm=T))
```

```
## # A tibble: 2 × 6
##   DX_APD count format(round(median(LP2_Disease_Duration, na.rm = T), 2), 2)    IQR   min   max
##   <chr>  <int> <chr>
## 1 CBS      39  3.9          2.25  1.03  14.8
## 2 PSP      28  4.83         3.93  1.65  17.3
## # i abbreviated name:
## # `¹`format(round(median(LP2_Disease_Duration, na.rm = T), 2), 2)`
```

```
df %>% summarize(count=n(), format(round(median(LP2_Disease_Duration, na.rm=T),2),2), IQR=IQR(LP2_Disease_Duration, na.rm=T), min=min(LP2_Disease_Duration, na.rm=T), max=max(LP2_Disease_Duration, na.rm=T))
```

```
##   count format(round(median(LP2_Disease_Duration, na.rm = T), 2), 2)    IQR
## 1     67                           4.1  2.847945
##   min     max
## 1 1.027397 17.27123
```

DURATION STATISTICS: WILCOXON

```
wilcox.test(df$LP2_Disease_Duration ~ df$DX_APD, paired=F)
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: df$LP2_Disease_Duration by df$DX_APD
## W = 409, p-value = 0.0827
## alternative hypothesis: true location shift is not equal to 0
```

4.1.4. COGNITIVE Z-SCORES

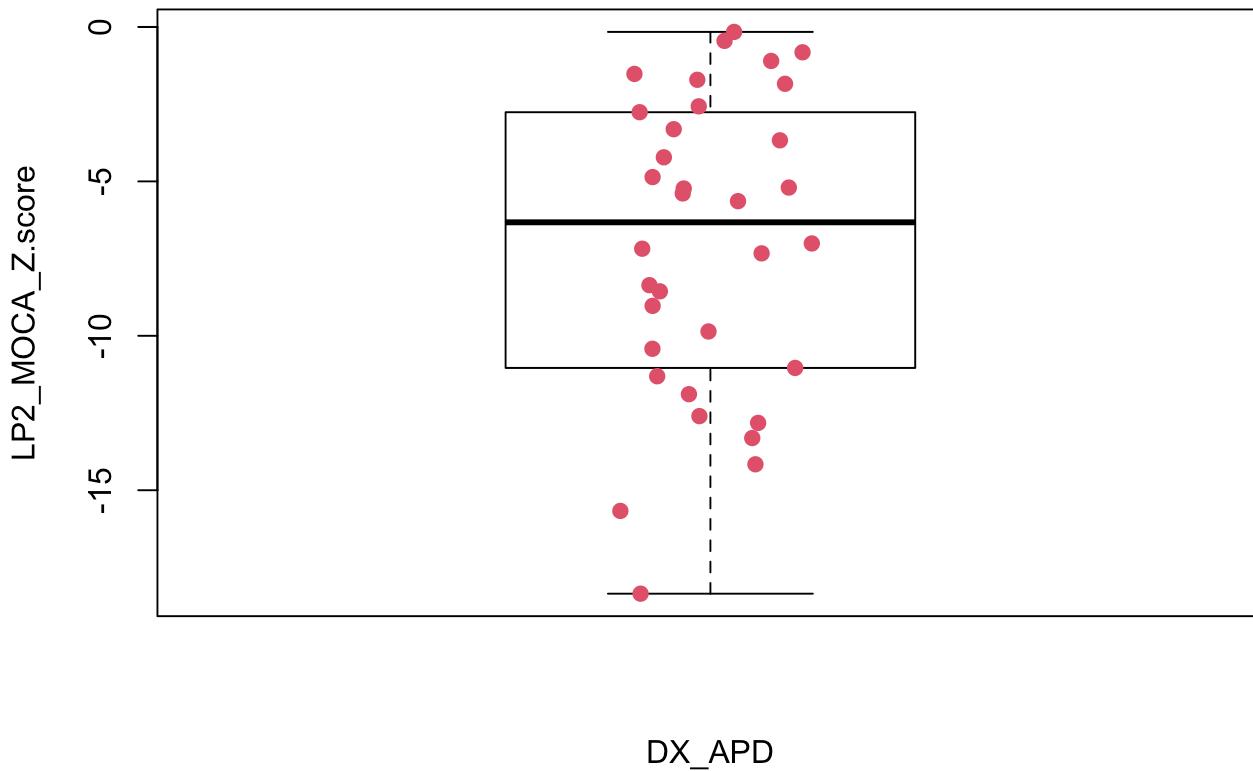
Comparisons shown in: Table 1, eTable 2

MoCA Z-SCORE STATISTICS: DISTRIBUTION

```
boxplot(LP2_MOCA_Z.score ~ DX_APD, data= CBSdf, col = "white")$out #identify outliers in each diagnosis. First, look at CBS: there is one so attribute its value to vector.
```

```
## numeric(0)
```

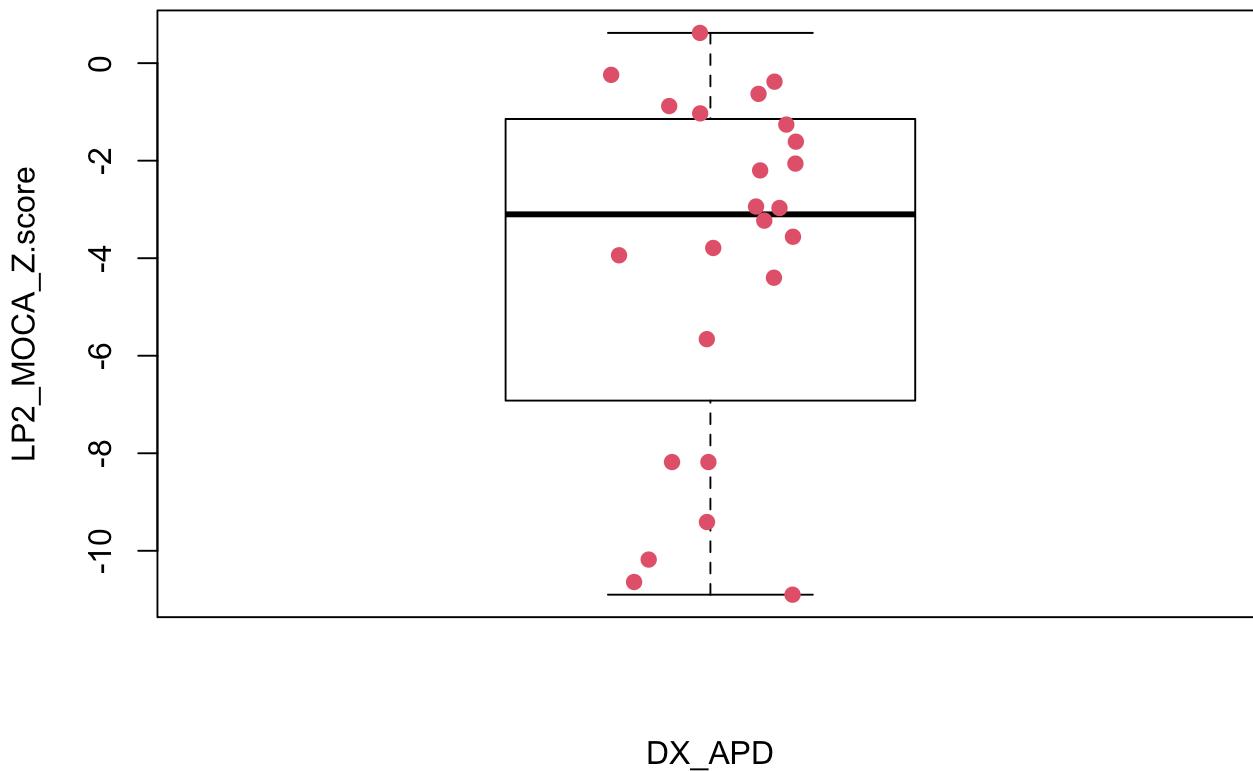
```
stripchart(LP2_MOCA_Z.score ~ DX_APD, data = CBSdf, method = "jitter", pch = 19, col = 2:4, vertical = TRUE, add = TRUE)
```



```
boxplot(LP2_MOCA_Z.score ~ DX_APD, data= PSPdf, col = "white")$out #Now identify outliers in PSP: since there is none, no need to attribute to a vector.
```

```
## numeric(0)
```

```
stripchart(LP2_MOCA_Z.score ~ DX_APD, data = PSPdf, method = "jitter", pch = 19, col = 2:4, vertical = TRUE, add = TRUE)
```



DX_APD

```
shapiro.test(CBSdf$LP2_MOCA_Z.score) #normal
```

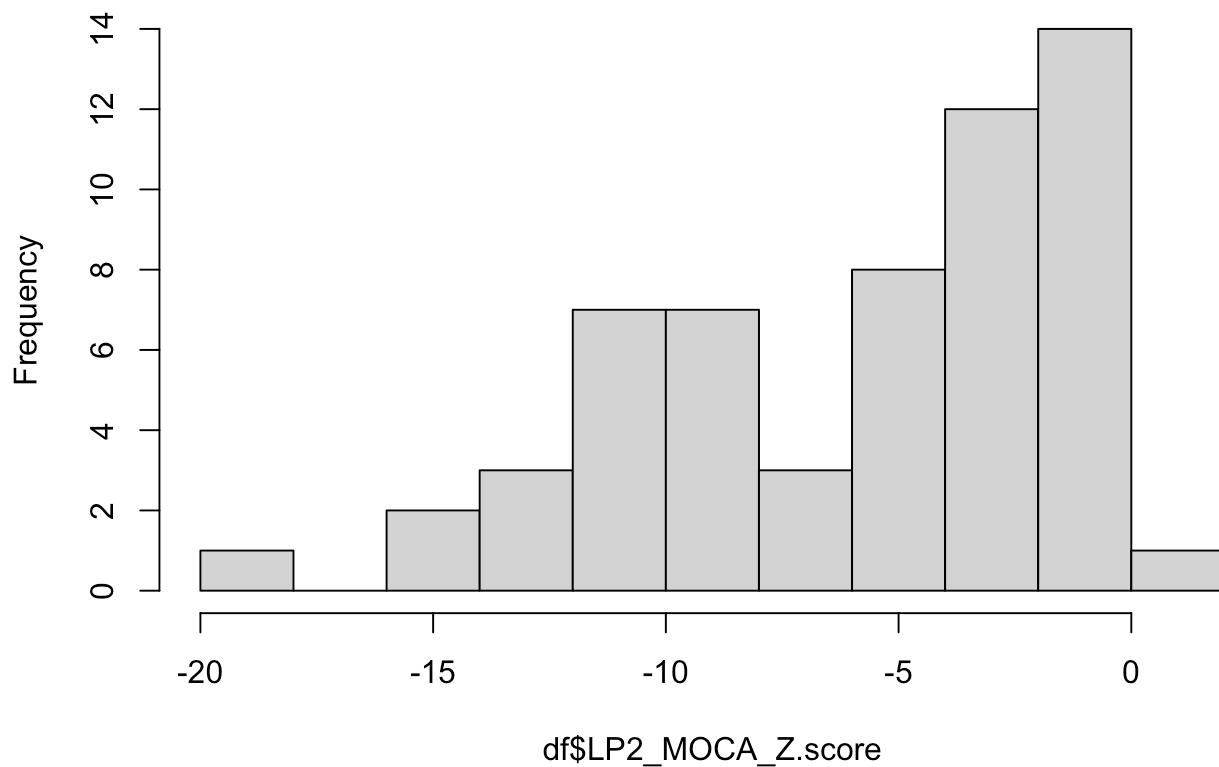
```
##  
## Shapiro-Wilk normality test  
##  
## data: CBSdf$LP2_MOCA_Z.score  
## W = 0.95558, p-value = 0.1802
```

```
shapiro.test(PSPdf$LP2_MOCA_Z.score) #not normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: PSPdf$LP2_MOCA_Z.score  
## W = 0.88622, p-value = 0.01111
```

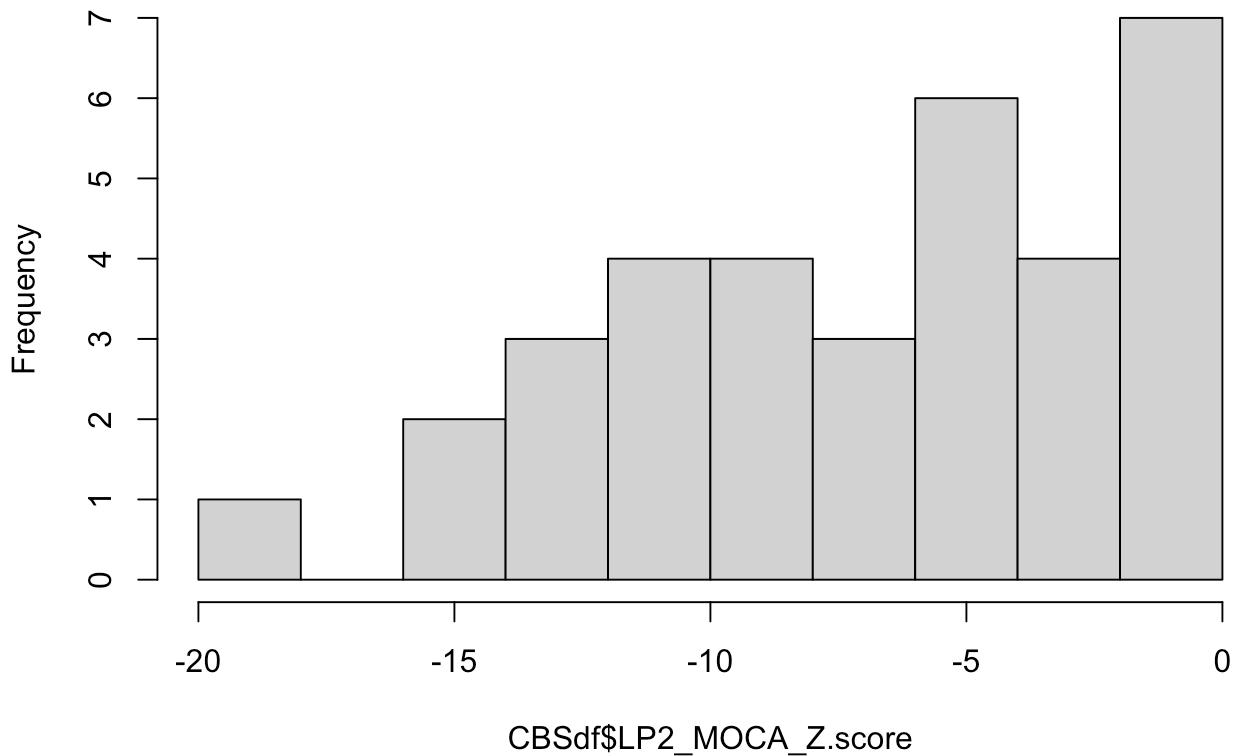
```
hist(df$LP2_MOCA_Z.score)
```

Histogram of df\$LP2_MOCA_Z.score



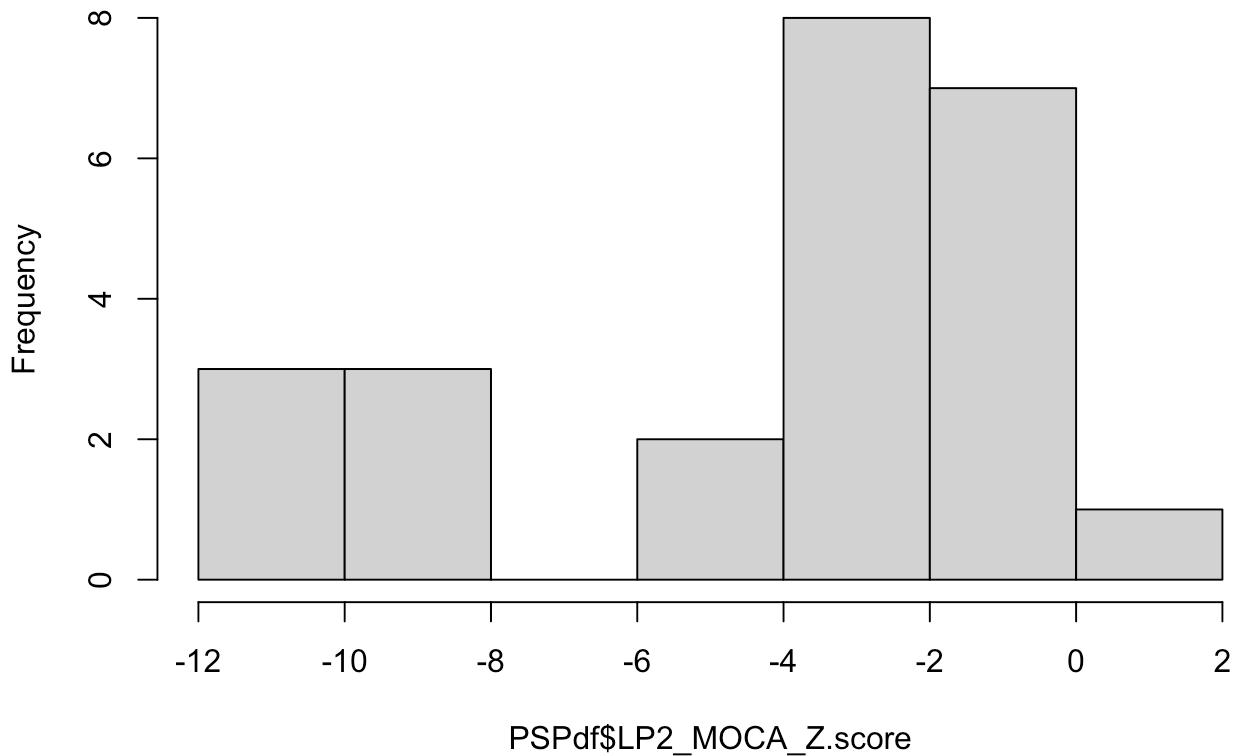
```
hist(CBSdf$LP2_MOCA_Z.score)
```

Histogram of CBSdf\$LP2_MOCA_Z.score



```
hist(PSPdf$LP2_MOCA_Z.score)
```

Histogram of PSPdf\$LP2_MOCA_Z.score



```
leveneTest(LP2_MOCA_Z.score ~ DX_APD, data = df) #heterodasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group  1 3.5999 0.06294 .
##      56
## -----
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

MoCA Z-SCORE STATISTICS: SUMMARY

```
df %>% group_by(DX_APD) %>% summarize(count=n(), format(round(median(LP2_MOCA_Z.score, na.rm=T),2),2), IQR=IQR(LP2_MOCA_Z.score, na.rm=T), min=min(LP2_MOCA_Z.score, na.rm=T), max=max(LP2_MOCA_Z.score, na.rm=T))
```

```
## # A tibble: 2 × 6
##   DX_APD count format(round(median(LP2_MOCA_Z.score, na.rm =...¹    IQR      min      max
##   <chr>  <int> <chr>                               <dbl> <dbl> <dbl>
## 1 CBS      39 -6.32                           7.99 -18.4 -0.16
## 2 PSP      28 -3.1                            5.09 -10.9  0.62
## # i abbreviated name:
## #   ¹`format(round(median(LP2_MOCA_Z.score, na.rm = T), 2), 2)`
```

```
df %>% summarize(count=n(), format(round(median(LP2_MOCA_Z.score, na.rm=T),2),2), IQR=IQR(LP2_MOCA_Z.score, na.rm=T), min=min(LP2_MOCA_Z.score, na.rm=T), max=max(LP2_MOCA_Z.score, na.rm=T))
```

```
##   count format(round(median(LP2_MOCA_Z.score, na.rm = T), 2), 2) IQR      min
## 1     67                               -4.63  7.42 -18.35
##   max
## 1 0.62
```

MoCA Z-SCORE STATISTICS: WILCOXON

```
wilcox.test(df$LP2_MOCA_Z.score ~ df$DX_APD, paired=F) #Since looking at z-score, no need to correct by age for this comparison
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: df$LP2_MOCA_Z.score by df$DX_APD
## W = 258, p-value = 0.01826
## alternative hypothesis: true location shift is not equal to 0
```

ALL COGNITIVE SCORES DISTRIBUTION/SUMMARY/STATISTICS: NO NEED IN TABLE 1 (REDUNDANT)

```
shapiro.test(CBSdf$LP2_Cognitive_Z.score) #not normal
```

```
##
## Shapiro-Wilk normality test
##
## data: CBSdf$LP2_Cognitive_Z.score
## W = 0.94653, p-value = 0.06848
```

```
shapiro.test(PSPdf$LP2_Cognitive_Z.score) #not normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: PSPdf$LP2_Cognitive_Z.score  
## W = 0.90904, p-value = 0.02506
```

```
leveneTest(LP2_Cognitive_Z.score ~ DX_APD, data = df) #heterodasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to  
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)  
##      Df F value Pr(>F)  
## group  1 5.0982 0.02748 *  
##       62  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
wilcox.test(df$LP2_Cognitive_Z.score ~ df$DX_APD, paired=F)
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot  
## compute exact p-value with ties
```

```
##  
## Wilcoxon rank sum test with continuity correction  
##  
## data: df$LP2_Cognitive_Z.score by df$DX_APD  
## W = 324, p-value = 0.0205  
## alternative hypothesis: true location shift is not equal to 0
```

```
df %>% summarize(count=n(), format(round(median(LP2_Cognitive_Z.score, na.rm=T),2),2), IQR=IQR(LP2_Cognitive_Z.score, na.rm=T), min=min(LP2_Cognitive_Z.score, na.rm=T), max=max(LP2_Cognitive_Z.score, na.rm=T))
```

```
##   count format(round(median(LP2_Cognitive_Z.score, na.rm = T), 2), 2)      IQR  
## 1     67                           -4.85 7.4775  
##   min  max  
## 1 -18.35 0.62
```

```
df %>% group_by(DX_APD) %>% summarize(count=n(), format(round(median(LP2_Cognitive_Z.score, na.rm=T),2),2), IQR=IQR(LP2_Cognitive_Z.score, na.rm=T), min=min(LP2_Cognitive_Z.score, na.rm=T), max=max(LP2_Cognitive_Z.score, na.rm=T))
```

```
## # A tibble: 2 × 6
##   DX_APD count format(round(median(LP2_Cognitive_Z.score, na.rm = TRUE), 2)) IQR min max
##   <chr>  <int> <chr>
## 1 CBS      39  -6.32          8.46 -18.4 -0.16
## 2 PSP      28  -3.4           5.99 -10.9  0.62
## # i abbreviated name:
## #   `format(round(median(LP2_Cognitive_Z.score, na.rm = TRUE), 2), 2)`
```

4.1.5. BIOMARKERS: ABETA42

Comparisons shown in: Table 1

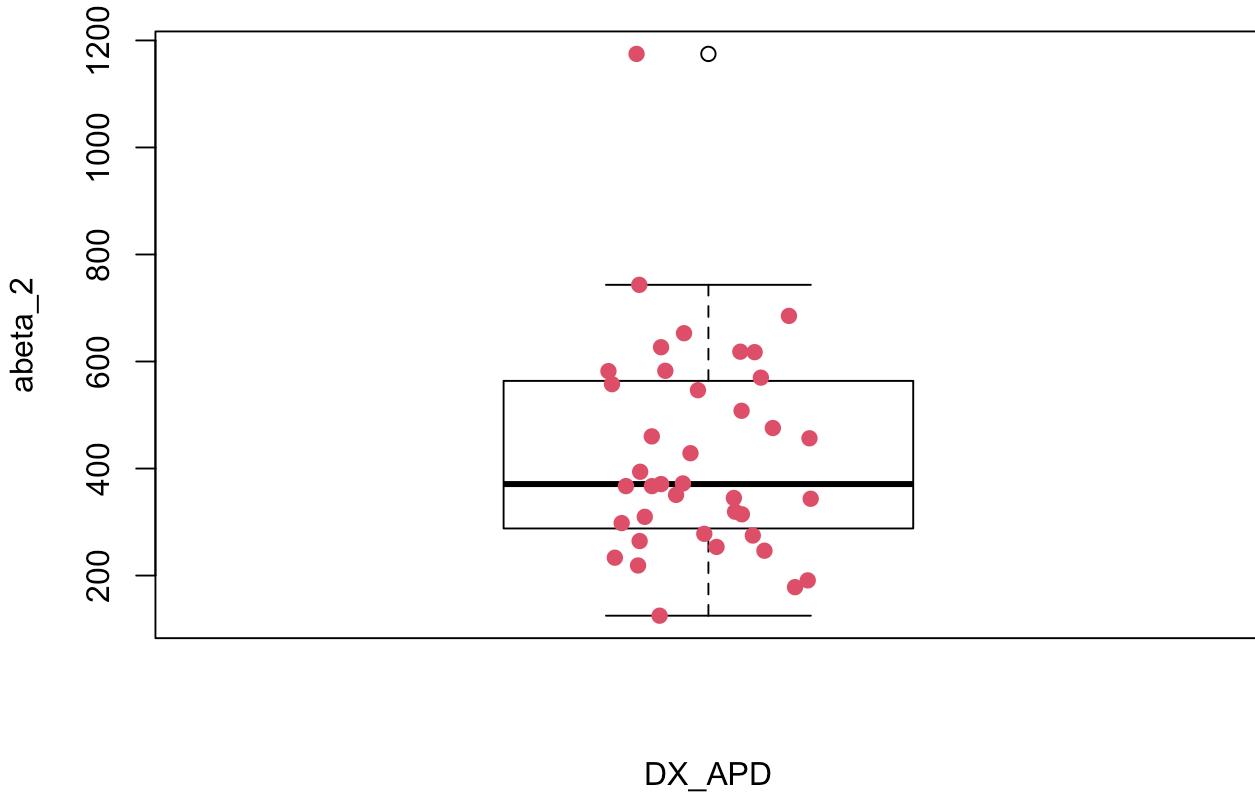
Biomarkers: The goal here is to share data that can be compared to other datasets as reported values for CBS and FTLD can have a wide range. Therefore, mean of raw values are presented here, even if analyses are on logged values. Mean and sd are calculated after removing outliers, calculated based on Tukey method (Q1 - 1.5IQR or Q3 + 1.5IQR). Exception for NfL which has a strong positive skew so instead the threshold is calculated by 3*IQR (Delaby et al 2020 on similar cohort). Outliers are shown in notes of Tables every time. Analyses were done without outlier exclusion as they incorporated multiple variables and were complex models so given the heterogeneity of the dataset, it would lead to groups with small sample sizes being possibly under-represented. Instead, models were run on dataset minus the outlier, 1 outlier at a time. This allowed us to see how different outliers could affect the results of the main model selected.

Abeta42 STATISTICS: OUTLIERS

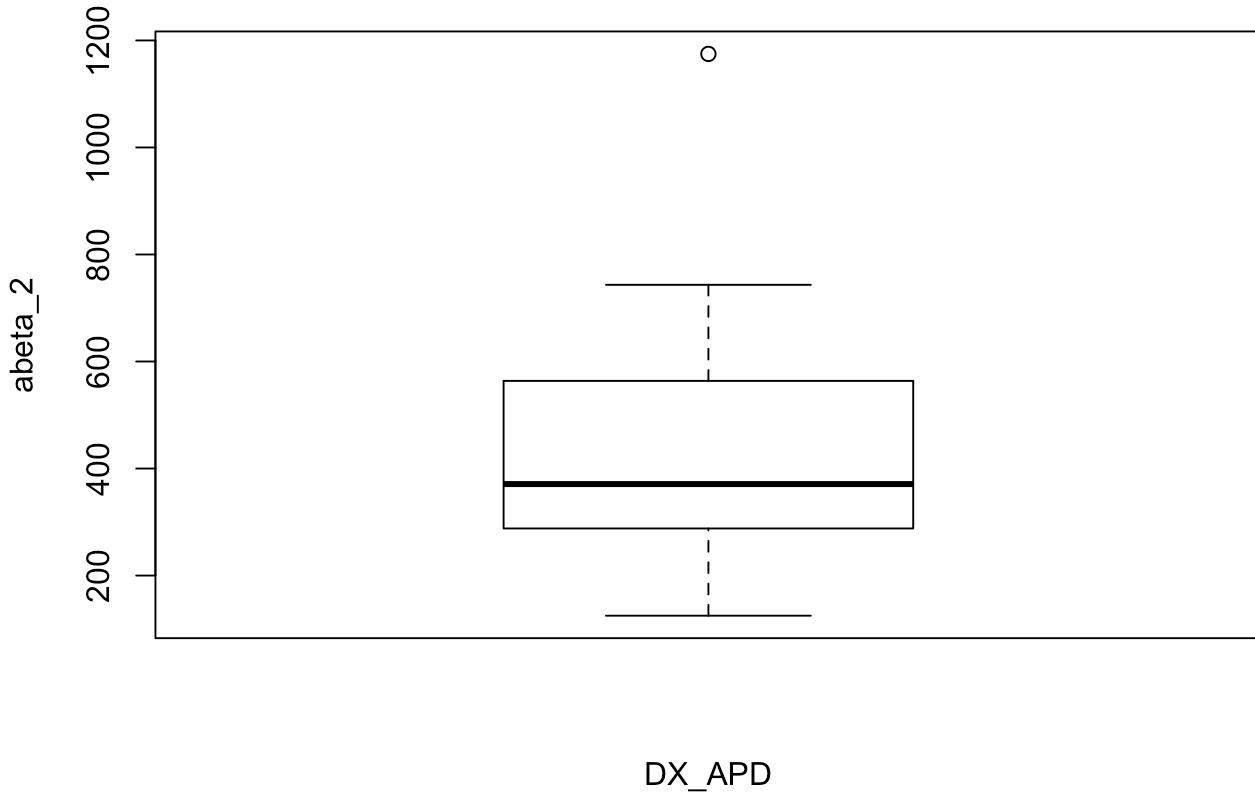
```
# First, display the data points of one group. boxplot() will identify outliers in given df but stripchart function can be funky hence convoluted script.
# Code below plots (boxplot() + assigns outliers to diagnosis-specific vector)
# Outliers are removed before calculating the values that are presented in table.
boxplot(abeta_2 ~ DX_APD, data= CBSdf, col = "white")$out #identify outliers in each diagnosis. First, look at CBS: there is one so attribute its value to vector.
```

```
## [1] 1174.82
```

```
stripchart(abeta_2 ~ DX_APD, data = CBSdf, method = "jitter", pch = 19, col = 2:4, vertical = TRUE, add = TRUE)
```



```
CBSvec.outliers <- boxplot(abeta_2 ~ DX_APD, data= CBSdf, col = "white")$out #list of outlier (Tukey method 1.5 IQR in each RTQUIC_DX combination)
```

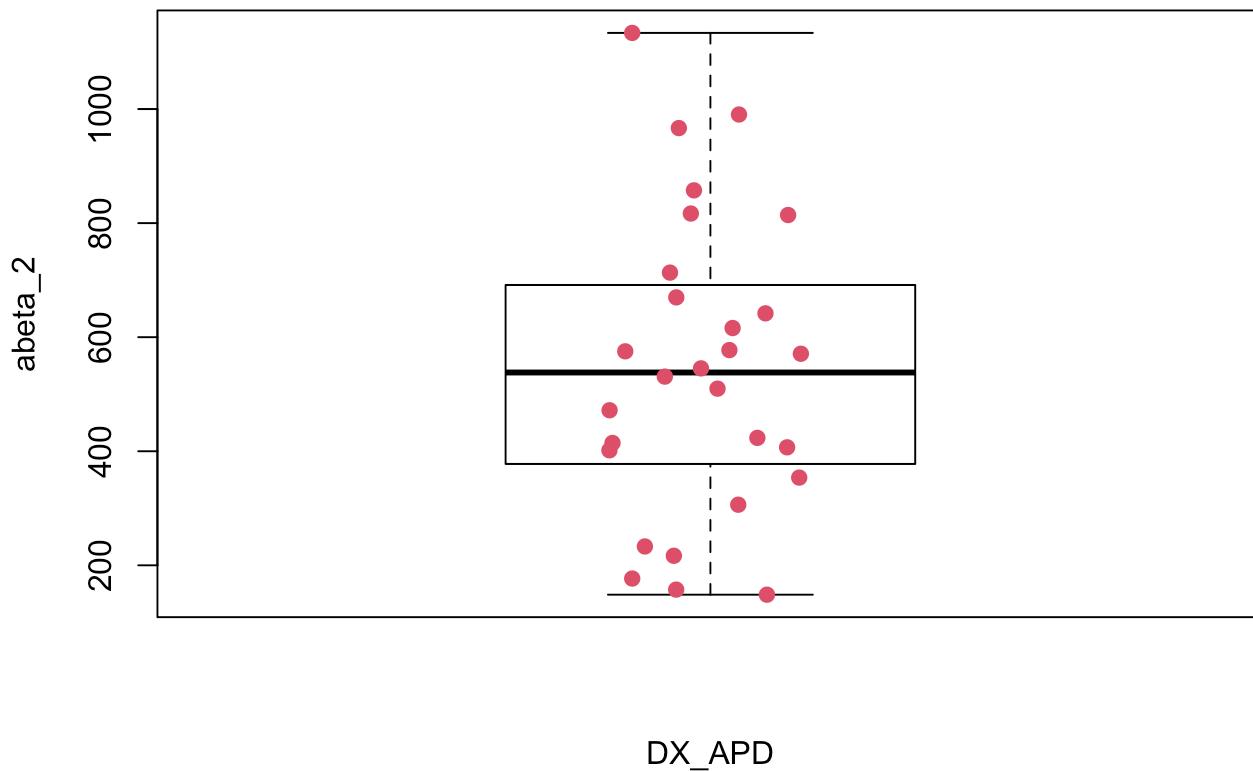


DX_APD

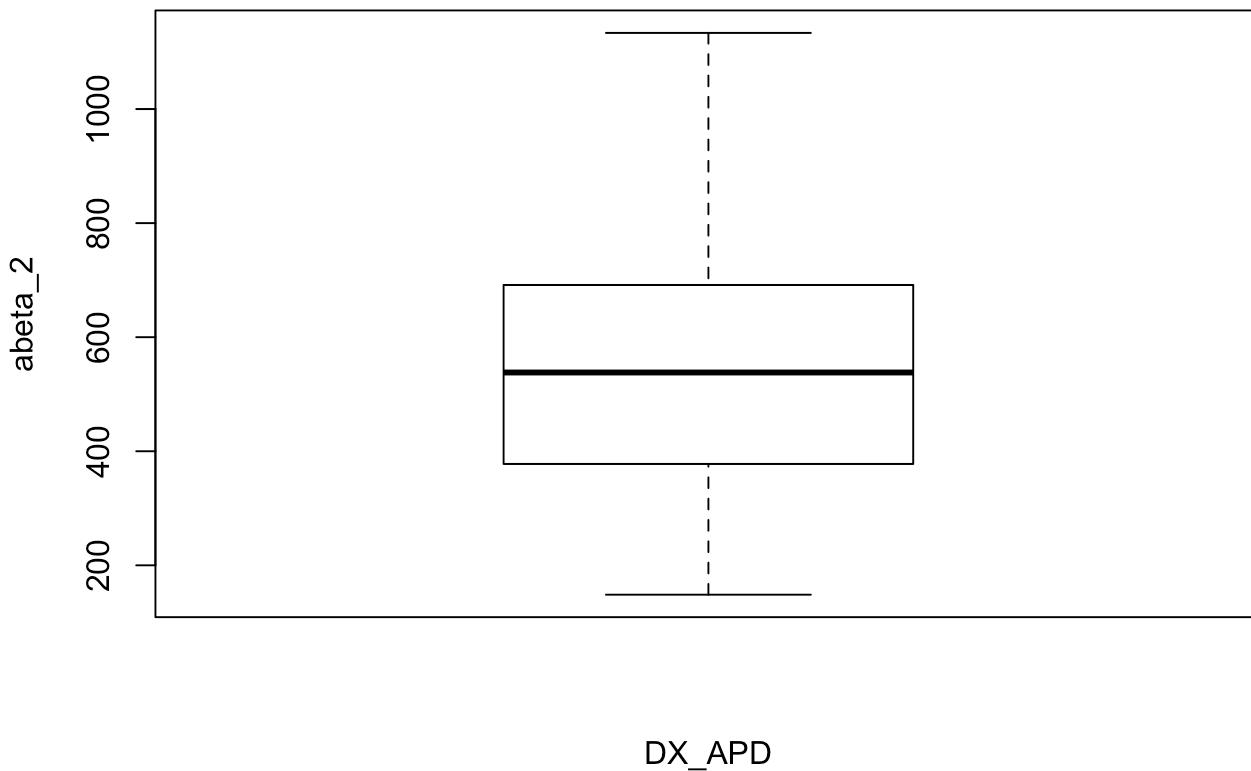
```
boxplot(abeta_2 ~ DX_APD, data= PSPdf, col = "white")$out #Now identify outliers in PSP:  
since there is none, no need to attribute to a vector.
```

```
## numeric(0)
```

```
stripchart(abeta_2 ~ DX_APD, data = PSPdf, method = "jitter", pch = 19, col = 2:4, v  
ertical = TRUE, add = TRUE)
```



```
PSPvec.outliers <- boxplot(abeta_2 ~ DX_APD, data= PSPdf, col = "white")$out
```



DX_APD

```
dfabeta<- df[!(df$DX_APD=="CBS" & (df$abeta_2 %in% CBSvec.outliers)), ] %>% remove_empty("rows") %>% data.frame() #Removes all subjects who are CBS and either have no NFL value or one over the threshold
dfabeta<- dfabeta[!(dfabeta$DX_APD=="PSP" & (dfabeta$abeta_2 %in% PSPvec.outliers)), ] %>% remove_empty("rows") %>% data.frame() #Removes all subjects who are CBS and either have no NFL value or one over the threshold
dfabeta[, c("DX_APD", "abeta_2")]
```

```
##   DX_APD    abeta_2
## 1   CBS 372.1540
## 2   CBS 569.8260
## 3   CBS 233.3620
## 4   PSP 530.8380
## 5   CBS 582.6150
## 6   CBS 393.9600
## 8   PSP 1133.5210
## 9   PSP 990.4640
## 10  PSP 157.3680
## 11  PSP 857.4720
## 12  PSP 616.0428
## 13  PSP 401.6630
## 14  PSP 423.4450
## 15  PSP 406.8770
## 16  PSP 471.8610
## 17  PSP 575.1950
## 18  PSP 353.7820
## 19  PSP 216.7000
## 20  CBS 274.8000
## 21  CBS 178.2770
## 22  CBS 557.7000
## 23  CBS 367.1650
## 24  CBS 617.5000
## 25  CBS 460.1000
## 26  CBS 314.4800
## 27  PSP 966.6420
## 28  PSP 641.7790
## 30  CBS 277.9730
## 31  PSP 814.2000
## 32  CBS 743.2790
## 33  CBS 626.6690
## 34  PSP 816.8230
## 35  PSP 669.8000
## 36  CBS 475.7000
## 37  PSP 148.4680
## 38  PSP 713.2370
## 39  PSP 233.0530
## 40  CBS 582.0360
## 41  CBS 367.1500
## 42  CBS 264.5000
## 43  CBS 190.9850
## 44  CBS 343.5230
## 45  PSP 176.8050
## 46  CBS 319.3500
## 47  CBS 218.8500
## 48  CBS 253.5370
## 49  CBS 652.9000
## 50  CBS 344.9500
## 51  PSP 545.3210
## 52  CBS 685.2000
## 53  CBS 618.3500
```

```
## 54   CBS  507.9095
## 55   CBS  246.4260
## 56   CBS  546.4000
## 57   CBS  456.5250
## 59   CBS  370.9500
## 60   PSP  570.9500
## 61   PSP  306.2360
## 62   CBS  428.6500
## 63   CBS  298.0500
## 64   CBS  125.0000
## 65   CBS  350.7500
## 66   PSP  509.7530
## 67   CBS  309.7500
## 68   PSP  577.3770
## 69   PSP  414.4840
```

```
if (nrow(dfabeta) == nrow(df)) {cat("No outliers were removed for Abeta42 comparisons\n")} else {cat("The following outliers were removed from the CBS:", CBSvec.outliers, "from the PSP:", PSPvec.outliers, "\n")}
```

```
## The following outliers were removed from the CBS: 1174.82 from the PSP: .
```

Abeta42 STATISTICS: DISTRIBUTION

```
shapiro.test(dfabeta[(dfabeta$DX_APD == "CBS"), ]$logabeta) #normal
```

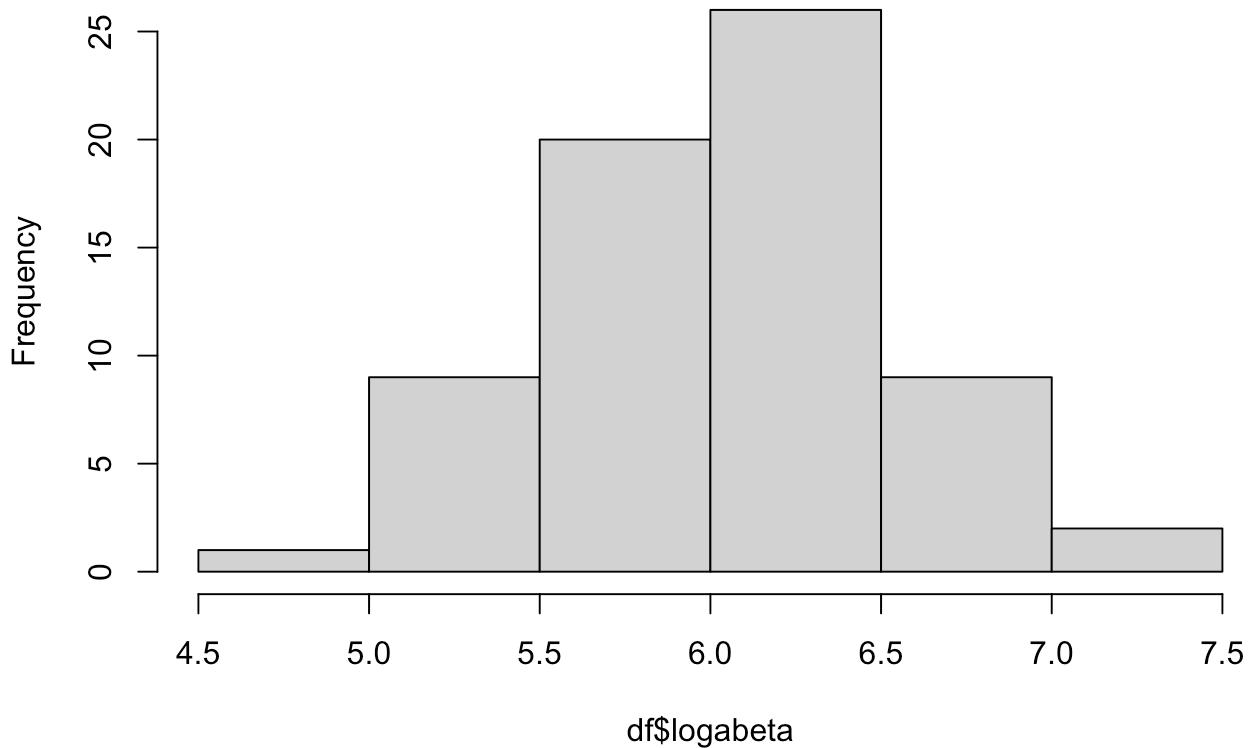
```
##
## Shapiro-Wilk normality test
##
## data: dfabeta[(dfabeta$DX_APD == "CBS"), ]$logabeta
## W = 0.96823, p-value = 0.3462
```

```
shapiro.test(dfabeta[(dfabeta$DX_APD == "PSP"), ]$logabeta) #normal
```

```
##
## Shapiro-Wilk normality test
##
## data: dfabeta[(dfabeta$DX_APD == "PSP"), ]$logabeta
## W = 0.94239, p-value = 0.1272
```

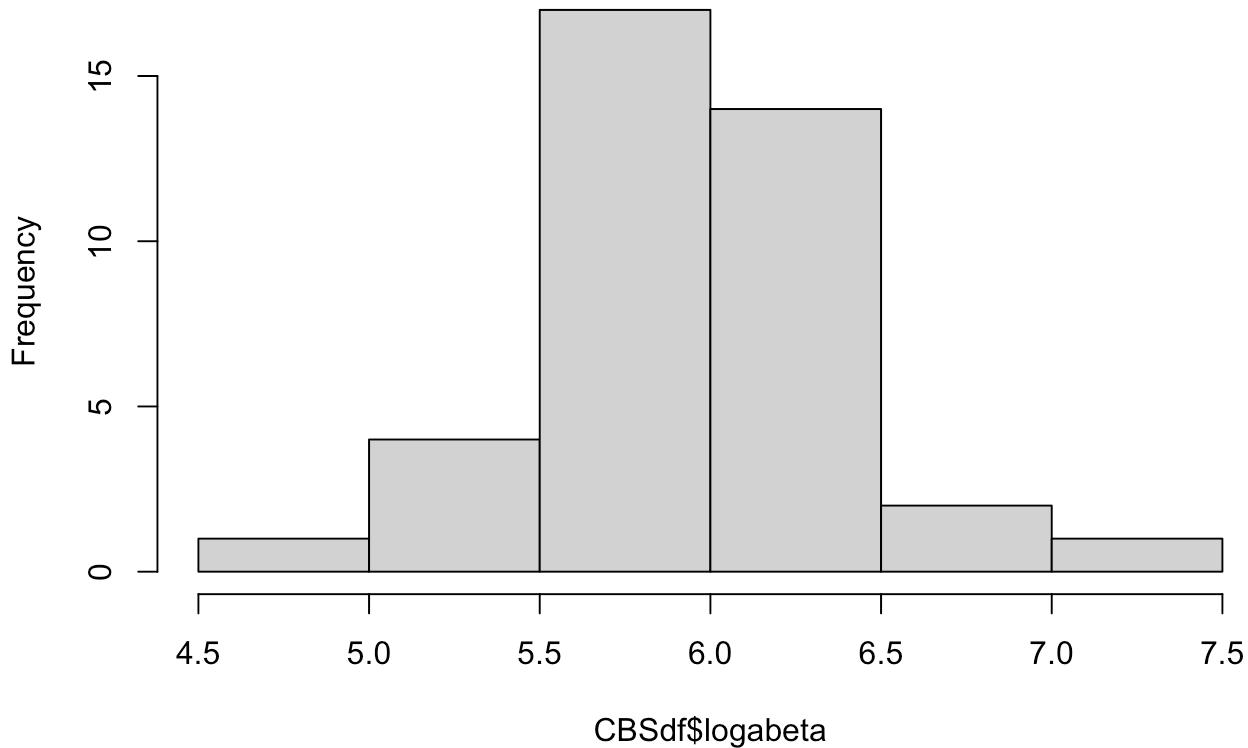
```
hist(df$logabeta)
```

Histogram of df\$logabeta



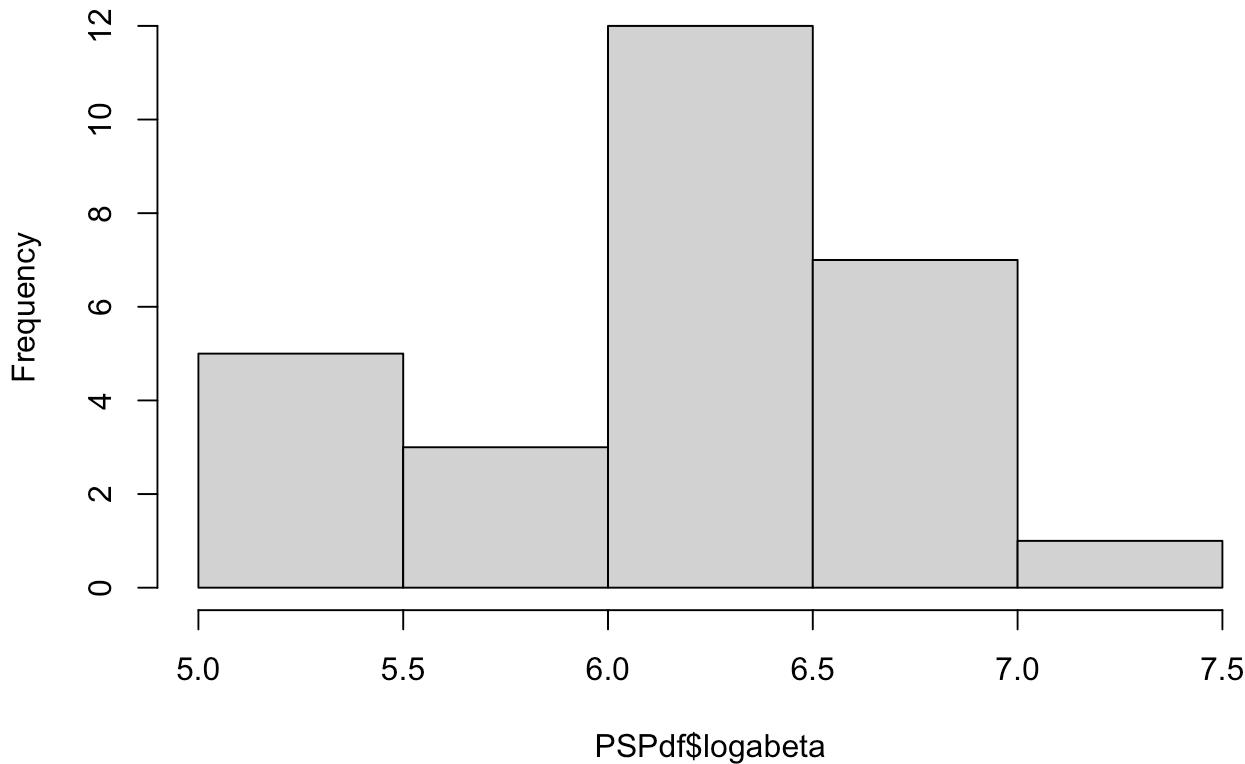
```
hist(CBSdf$logabeta)
```

Histogram of CBSdf\$logabeta



```
hist(PSPdf$logabeta)
```

Histogram of PSPdf\$logabeta



```
leveneTest(logabeta ~ DX_APD, data = dfabeta) #homoscedasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  1.2898 0.2603
##       64
```

Abeta42 STATISTICS: DISTRIBUTION

```
dfabeta %>% summarize(count=n(), format(round(mean(abeta_2, na.rm=T),2),2), sd=sd(abeta_2, na.rm=T)) #Rounds up the sd for some reason
```

```
##   count format(round(mean(abeta_2, na.rm = T), 2), 2)      sd
## 1    66                      466.17 217.8691
```

```
dfabeta %>% group_by(DX_APD) %>% summarize(count=n(), format(round(mean(abeta_2, na.rm=T),2),2), sd=sd(abeta_2, na.rm=T)) #Rounds up the sd for some reason
```

```
## # A tibble: 2 × 4
##   DX_APD count `format(round(mean(abeta_2, na.rm = T), 2), 2)`    sd
##   <chr>  <int> <chr>
## 1 CBS      38  408.61          159.
## 2 PSP      28  544.29          262.
```

```
sd(dfabeta[(dfabeta$DX_APD == "CBS"), ]$abeta_2, na.rm=T)
```

```
## [1] 159.4879
```

```
sd(PSPdf$abeta_2, na.rm=T)
```

```
## [1] 261.5791
```

Abeta42 STATISTICS: ANCOVA

```
t.test(df$logabeta ~ df$DX_APD, var.equal=TRUE) #before moving to more complex model, run simple t-test
```

```
##
## Two Sample t-test
##
## data: df$logabeta by df$DX_APD
## t = -1.6716, df = 65, p-value = 0.09941
## alternative hypothesis: true difference in means between group CBS and group PSP is not equal to 0
## 95 percent confidence interval:
## -0.45313178  0.04020584
## sample estimates:
## mean in group CBS mean in group PSP
##           5.961194           6.167657
```

```
aov <- aov(logabeta ~ Age + DX_APD, df)
Anova(aov, type="II") #no interaction in the model so choosing type II
```

```
## Anova Table (Type II tests)
##
## Response: logabeta
##             Sum Sq Df F value    Pr(>F)
## Age          1.7724  1  7.8833 0.006606 ***
## DX_APD       0.0918  1  0.4085 0.524998
## Residuals 14.3888 64
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
check_normality(aov)
```

```
## OK: residuals appear as normally distributed (p = 0.684).
```

```
etaSquared(aov)
```

```
##          eta.sq eta.sq.part
## Age      0.105147362 0.109667556
## DX_APD  0.005448988 0.006342786
```

4.1.6. BIOMARKERS: PTAU181

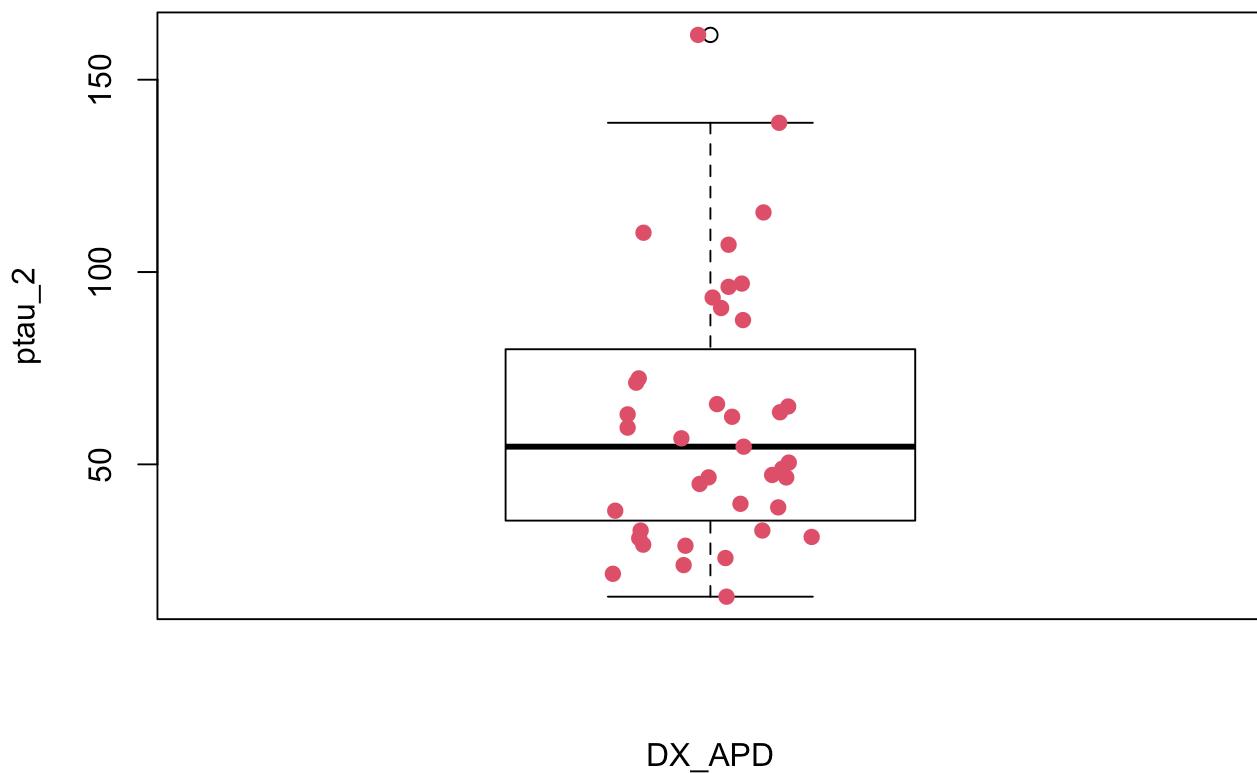
Comparisons shown in: Table 1

PTAU181 STATISTICS: OUTLIERS

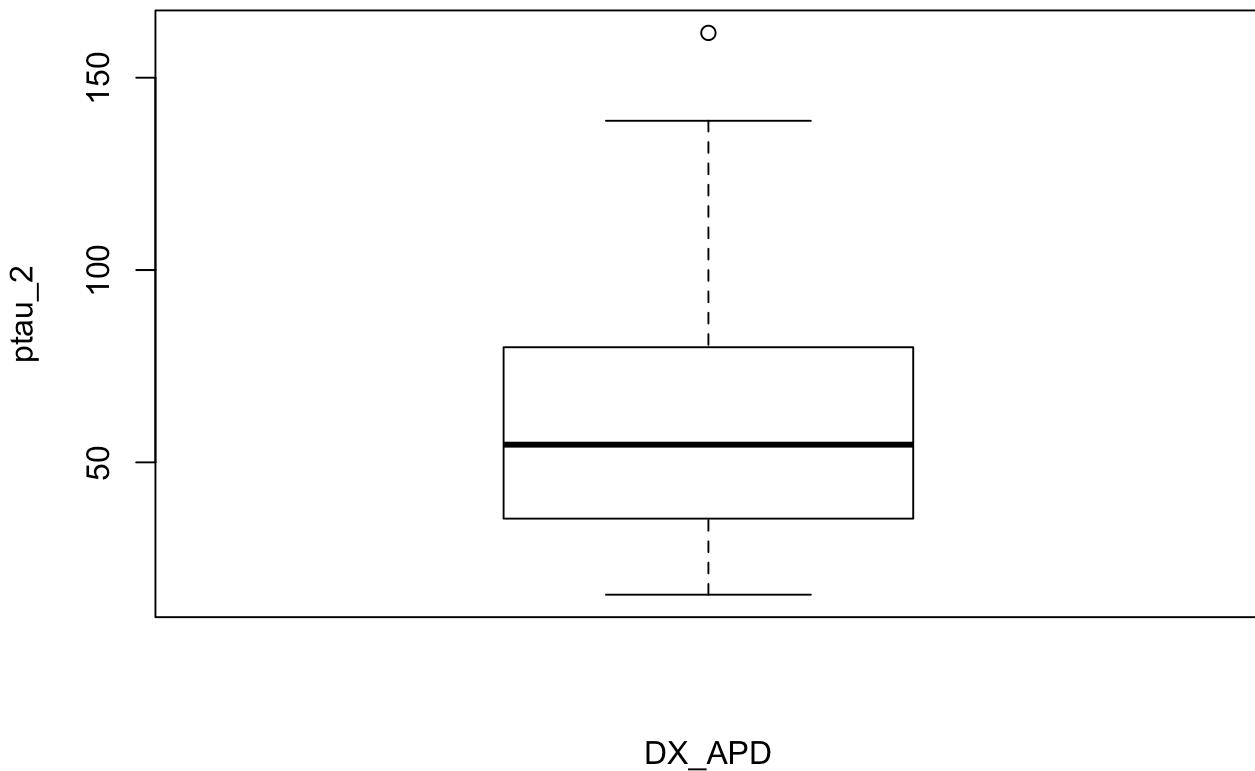
```
# First, display the data points of one group. boxplot() will identify outliers in given df but stripchart function can be funky hence convoluted script.
# Code below plots (boxplot() + assigns outliers to diagnosis-specific vector)
# Outliers are removed before calculating the values that are presented in table.
boxplot(ptau_2 ~ DX_APD, data= CBSdf, col = "white")$out #identify outliers in each diagnosis. First, look at CBS: there is one so attribute its value to vector.
```

```
## [1] 161.65
```

```
stripchart(ptau_2 ~ DX_APD, data = CBSdf, method = "jitter", pch = 19, col = 2:4, vertical = TRUE, add = TRUE)
```



```
CBSvec.outliers <- boxplot(ptau_2 ~ DX_APD, data= CBSdf, col = "white")$out #list of outlier (Tukey method 1.5 IQR in each RTQUIC_DX combination)
```

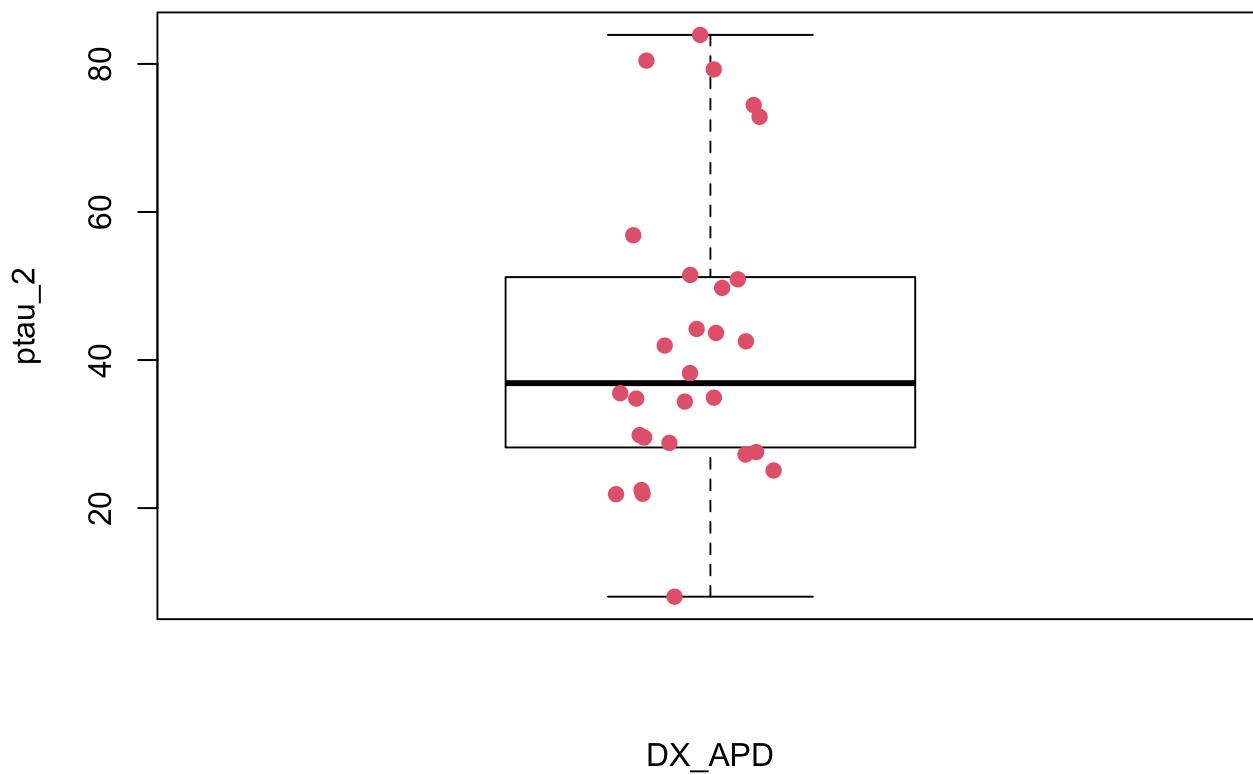


DX_APD

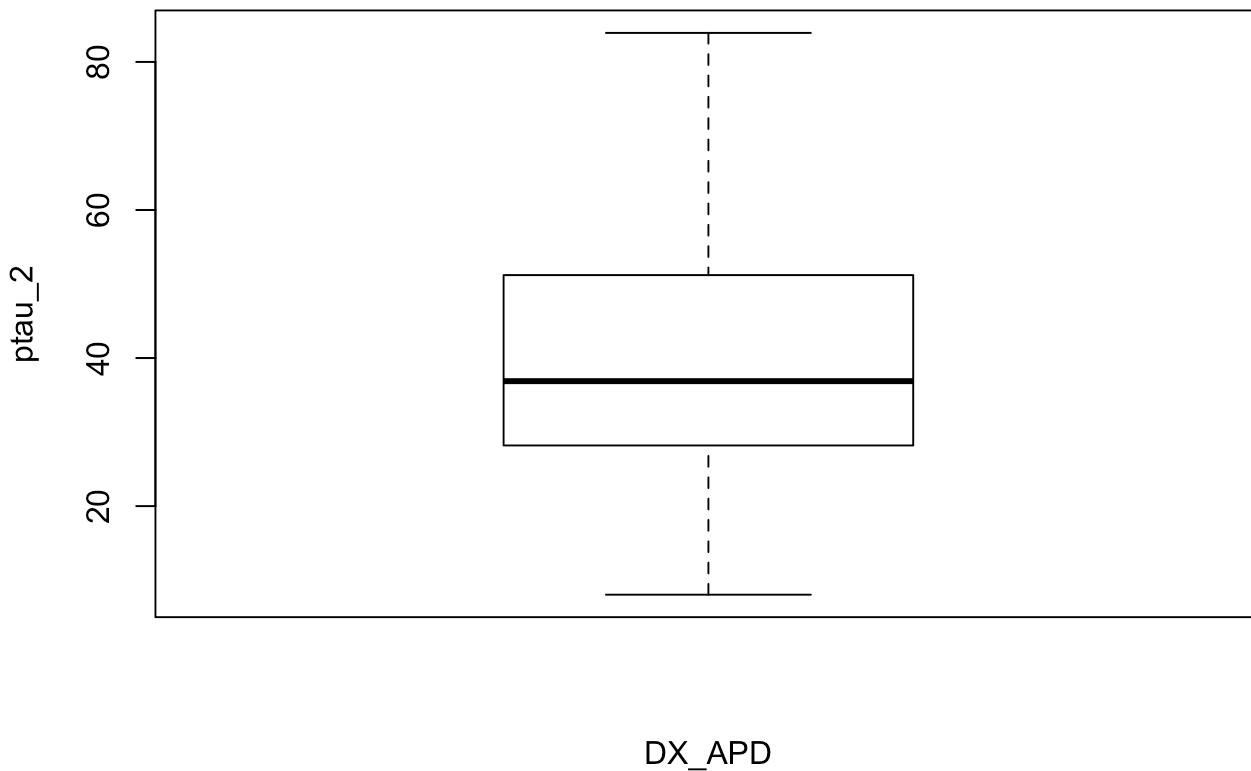
```
boxplot(ptau_2 ~ DX_APD, data= PSPdf, col = "white")$out #Now identify outliers in PSP:  
since there is none, no need to attribute to a vector.
```

```
## numeric(0)
```

```
stripchart(ptau_2 ~ DX_APD, data = PSPdf, method = "jitter", pch = 19, col = 2:4, ve  
rtical = TRUE, add = TRUE)
```



```
PSPvec.outliers <- boxplot(ptau_2 ~ DX_APD, data= PSPdf, col = "white")$out
```



DX_APD

```
dfptau<- df[!(df$DX_APD=="CBS" & (df$ptau_2 %in% CBSvec.outliers)), ] %>% remove_empty("rows") %>% data.frame() #Removes all subjects who are CBS and either have no NFL value or one over the threshold
dfptau<- dfptau[!(dfptau$DX_APD=="PSP" & (dfptau$ptau_2 %in% PSPvec.outliers)), ] %>% remove_empty("rows") %>% data.frame() #Removes all subjects who are CBS and either have no NFL value or one over the threshold

if (nrow(dfptau) == nrow(df)) {cat("No outliers were removed for ptau181 comparisons\n")} else {cat("The following outliers were removed from the CBS: ", CBSvec.outliers, "from the PSP: ", PSPvec.outliers, ".\n")}
```

```
## The following outliers were removed from the CBS: 161.65 from the PSP: .
```

PTAU181 STATISTICS: DISTRIBUTION

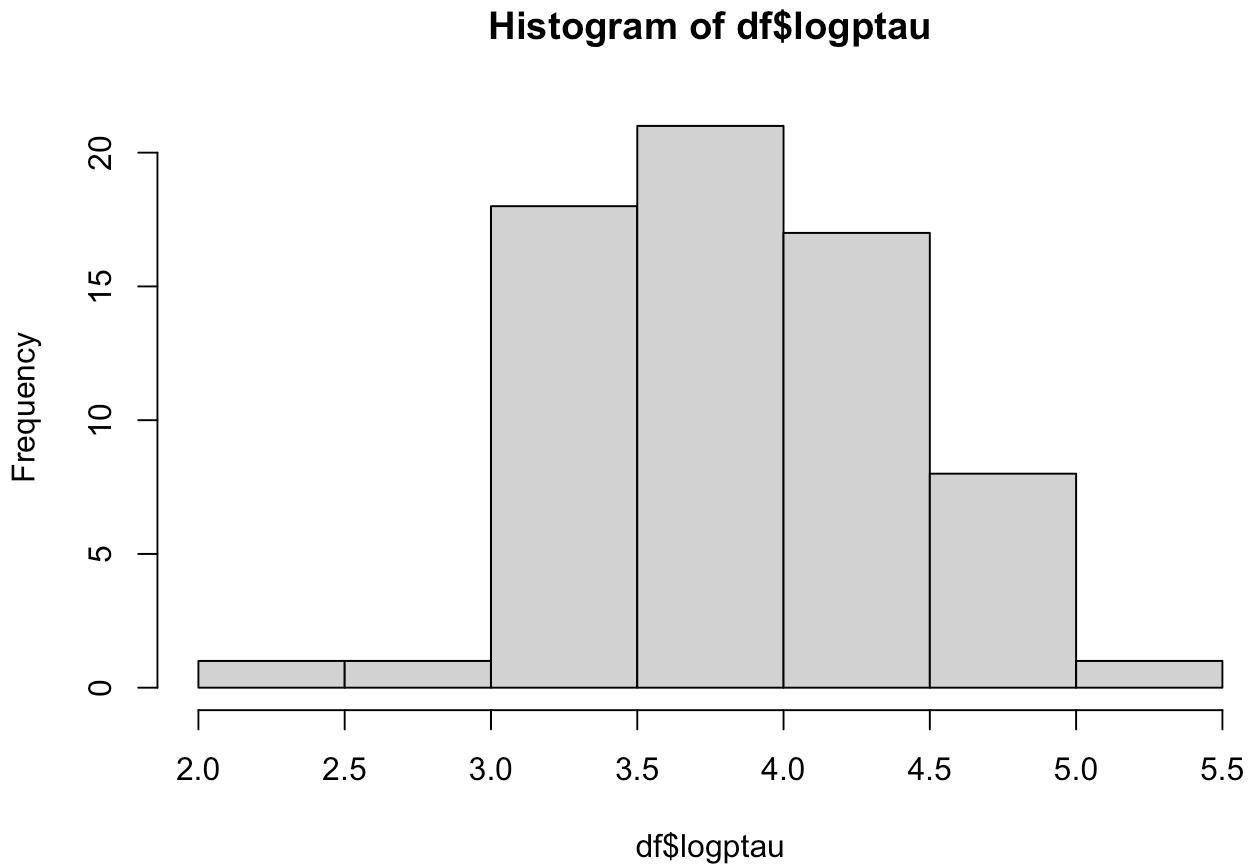
```
shapiro.test(dfptau[(dfptau$DX_APD == "CBS"), ]$logptau) #normal
```

```
##
## Shapiro-Wilk normality test
##
## data: dfptau[(dfptau$DX_APD == "CBS"), ]$logptau
## W = 0.98383, p-value = 0.8464
```

```
shapiro.test(PSPdf$logptau) #normal
```

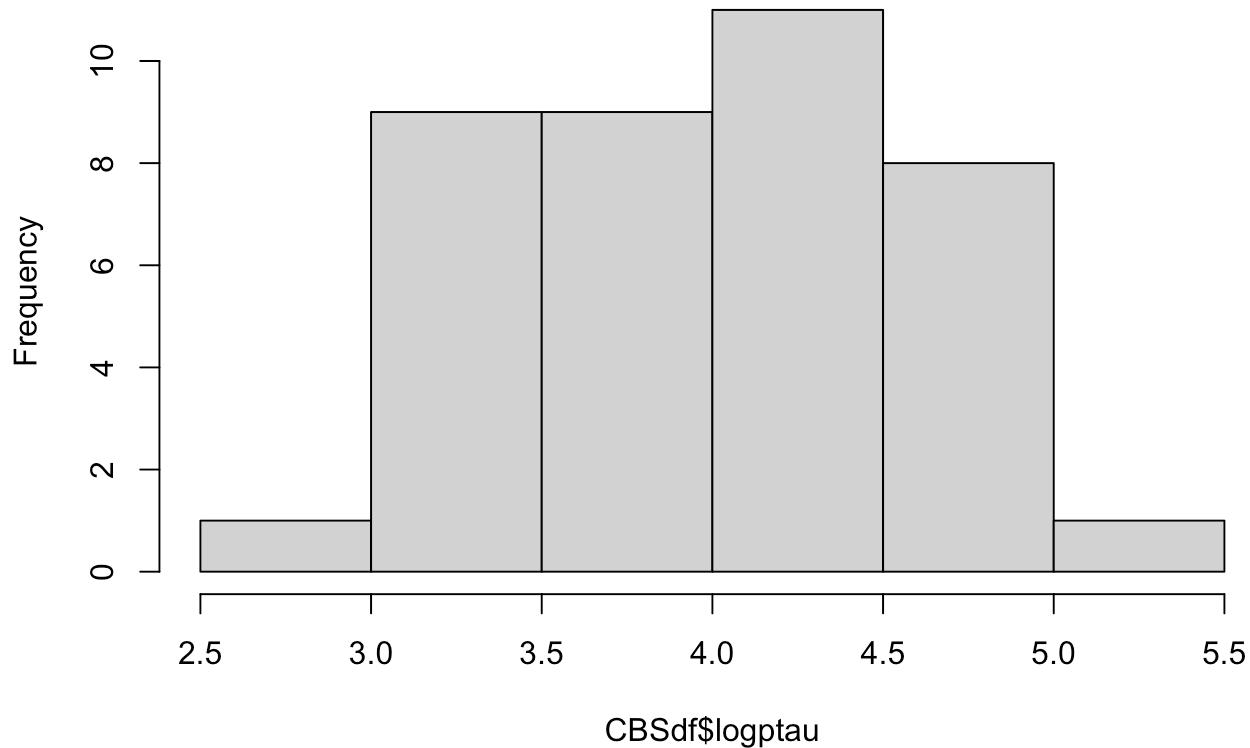
```
##  
## Shapiro-Wilk normality test  
##  
## data: PSPdf$logptau  
## W = 0.93649, p-value = 0.09
```

```
hist(df$logptau)
```



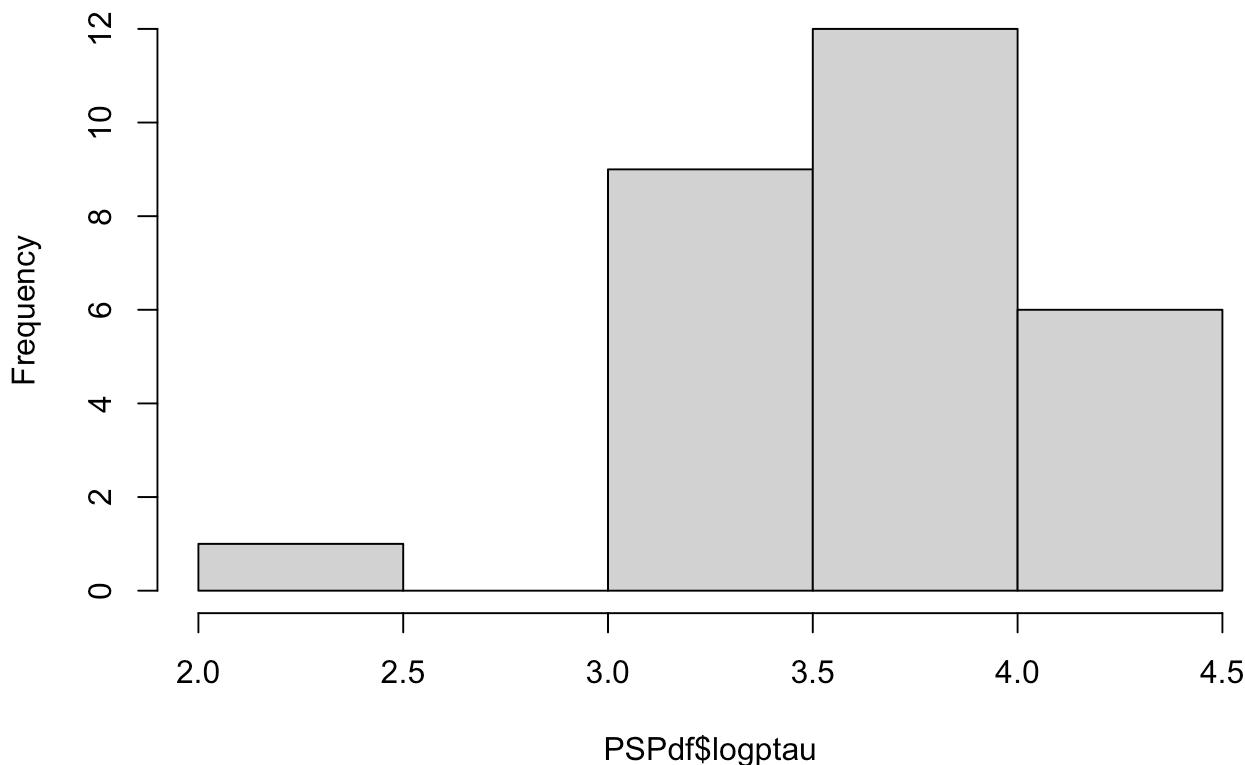
```
hist(CBSdf$logptau)
```

Histogram of CBSdf\$logptau



```
hist(PSPdf$logptau)
```

Histogram of PSPdf\$logptau



```
leveneTest(logptau ~ DX_APD, data = df) #homoscedasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.5859 0.4468
##       65
```

PTAU181 STATISTICS: ANCOVA

```
t.test(dfptau$logptau ~ dfptau$DX_APD, var.equal=TRUE)
```

```
## 
## Two Sample t-test
##
## data: dfptau$logptau by dfptau$DX_APD
## t = 2.4251, df = 64, p-value = 0.01813
## alternative hypothesis: true difference in means between group CBS and group PSP is not equal to 0
## 95 percent confidence interval:
## 0.05505646 0.56973159
## sample estimates:
## mean in group CBS mean in group PSP
## 3.950560 3.638165
```

```
aov <- aov(logptau ~ Age + DX_APD, dfptau)
Anova(aov, type="II") #no interaction in the model
```

```
## Anova Table (Type II tests)
##
## Response: logptau
##           Sum Sq Df F value    Pr(>F)
## Age        0.0300  1  0.1106  0.74054
## DX_APD     1.5289  1  5.6360  0.02066 *
## Residuals 17.0903 63
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
check_normality(aov)
```

```
## OK: residuals appear as normally distributed (p = 0.294).
```

```
etaSquared(aov)
```

```
##           eta.sq eta.sq.part
## Age      0.001605352 0.001752876
## DX_APD  0.081787677 0.082114395
```

4.1.7. BIOMARKERS: T-TAU

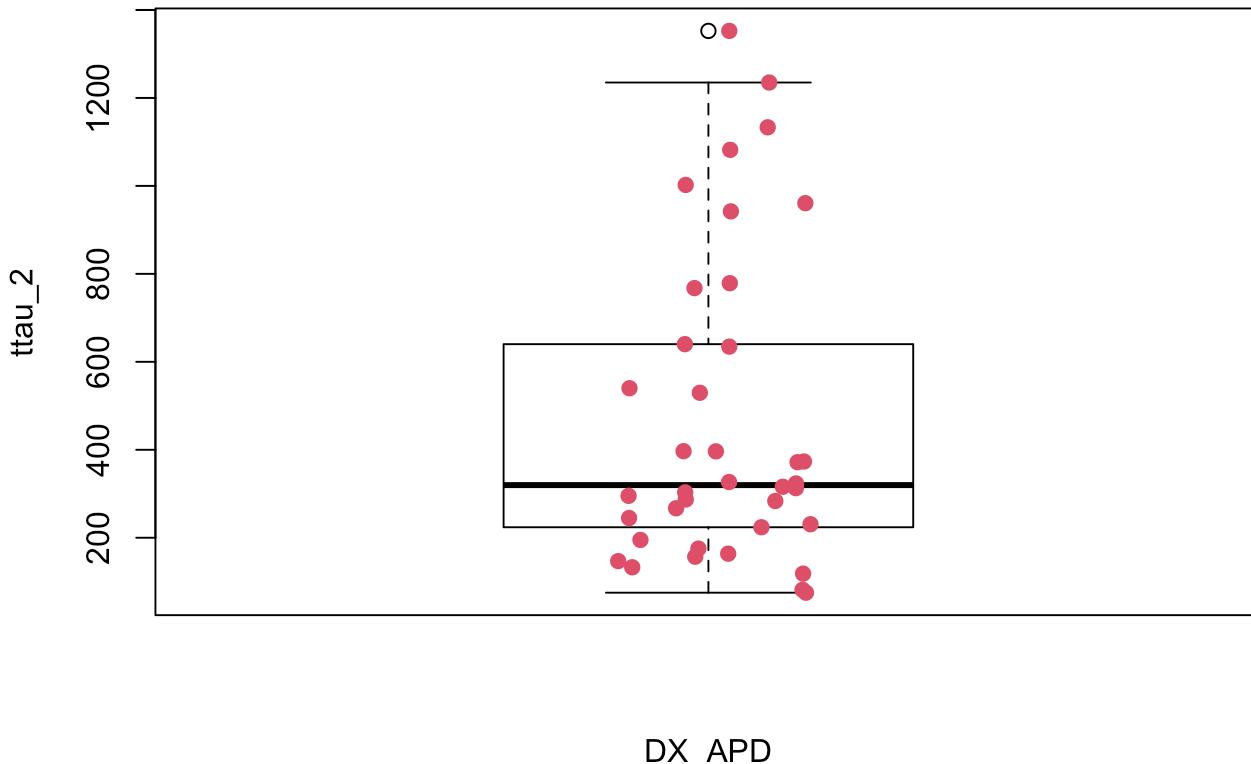
Comparisons shown in: Table 1

T-TAU STATISTICS: OUTLIERS

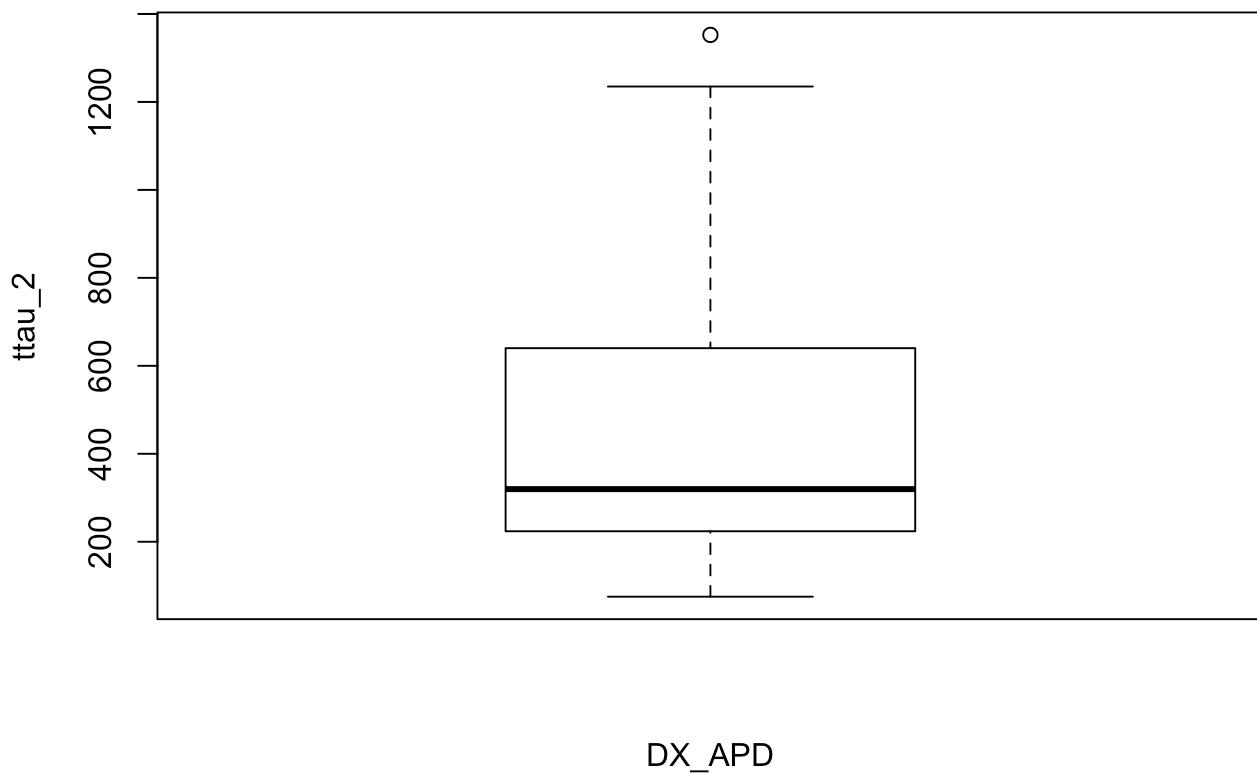
```
# First, display the data points of one group. boxplot() will identify outliers in given df but stripchart function can be funky hence convoluted script.
# Code below plots (boxplot() + assigns outliers to diagnosis-specific vector)
# Outliers are removed before calculating the values that are presented in table.
boxplot(ttau_2 ~ DX_APD, data= CBSdf, col = "white")$out #identify outliers in each diagnosis. First, look at CBS: there is one so attribute its value to vector.
```

```
## [1] 1352.4
```

```
stripchart(ttau_2 ~ DX_APD, data = CBSdf, method = "jitter", pch = 19, col = 2:4, vertical = TRUE, add = TRUE)
```



```
CBSvec.outliers <- boxplot(ttau_2 ~ DX_APD, data= CBSdf, col = "white")$out #list of outlier (Tukey method 1.5 IQR in each RTQUIC_DX combination)
```

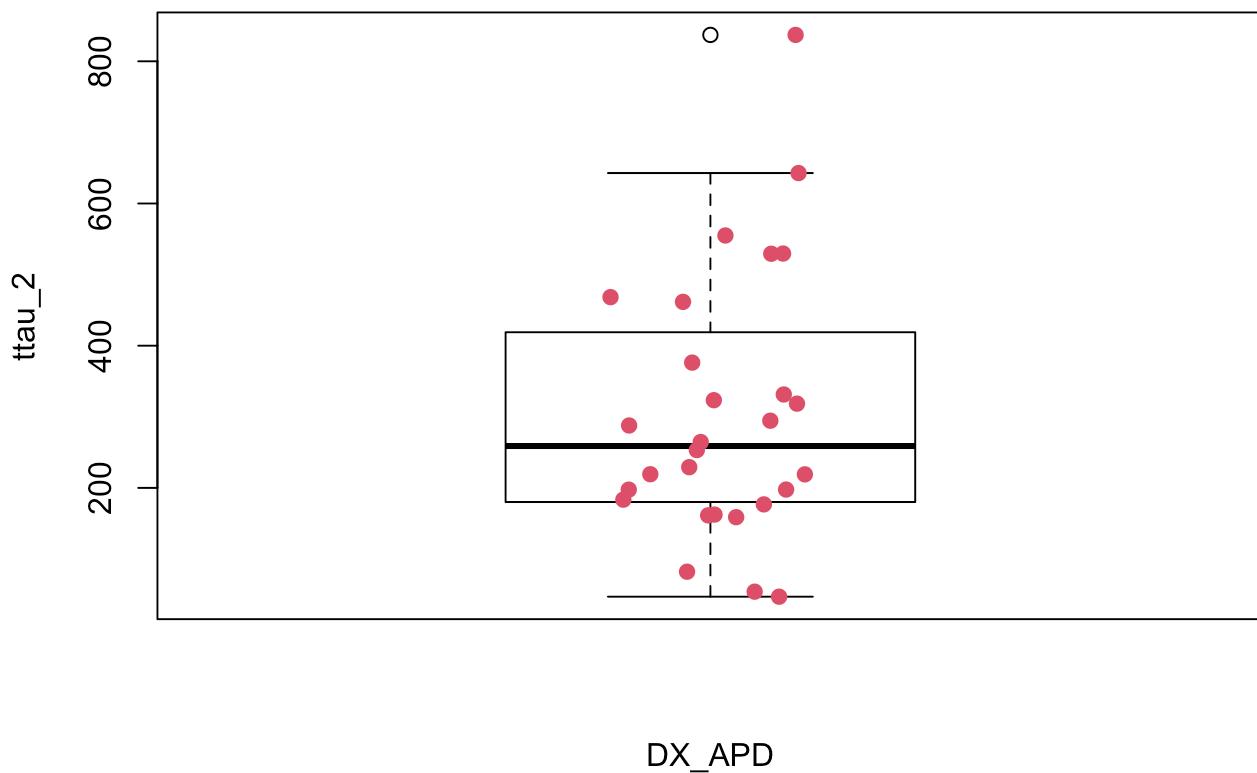


DX_APD

```
boxplot(ttau_2 ~ DX_APD, data= PSPdf, col = "white")$out #Now identify outliers in PSP:  
since there is none, no need to attribute to a vector.
```

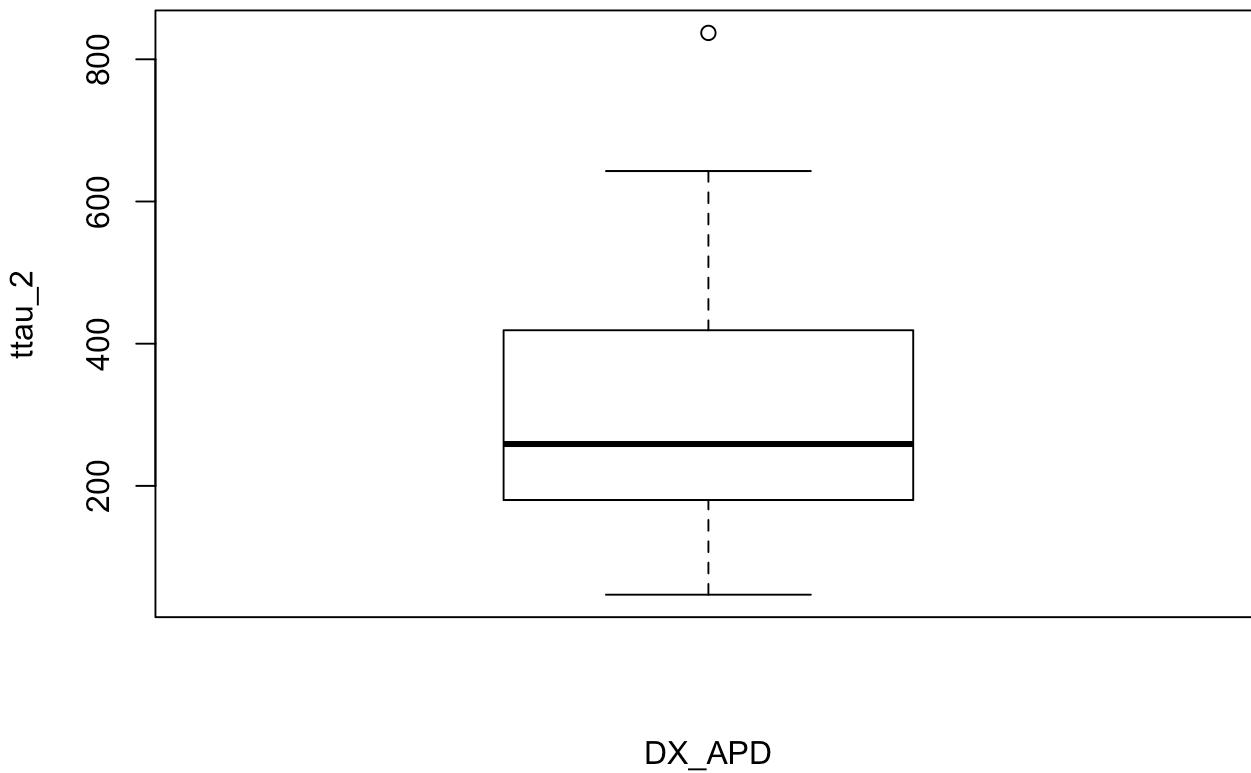
```
## [1] 837.1
```

```
stripchart(ttau_2 ~ DX_APD, data = PSPdf, method = "jitter", pch = 19, col = 2:4, ve  
rtical = TRUE, add = TRUE)
```



DX_APD

```
PSPvec.outliers <- boxplot(ttau_2 ~ DX_APD, data= PSPdf, col = "white")$out #list of outlier (Tukey method 1.5 IQR in each RTQUIC_DX combination)
```



DX_APD

```
dfttau<- df[!(df$DX_APD=="CBS" & (df$ttau_2 %in% CBSvec.outliers)), ] %>% remove_empty("rows") %>% data.frame() #Removes all subjects who are CBS and either have no NFL value or one over the threshold
dfttau<- dfttau[!(dfttau$DX_APD=="PSP" & (dfttau$ttau_2 %in% PSPvec.outliers)), ] %>% remove_empty("rows") %>% data.frame() #Removes all subjects who are CBS and either have no NFL value or one over the threshold

if (nrow(dfttau) == nrow(df)) {cat("No outliers were removed for t-tau comparisons \n")}
else {cat("The following outliers were removed from the CBS: ", CBSvec.outliers, " from the PSP: ", PSPvec.outliers, " .\n")}
```

```
## The following outliers were removed from the CBS: 1352.4 from the PSP: 837.1 .
```

T-TAU STATISTICS: DISTRIBUTION

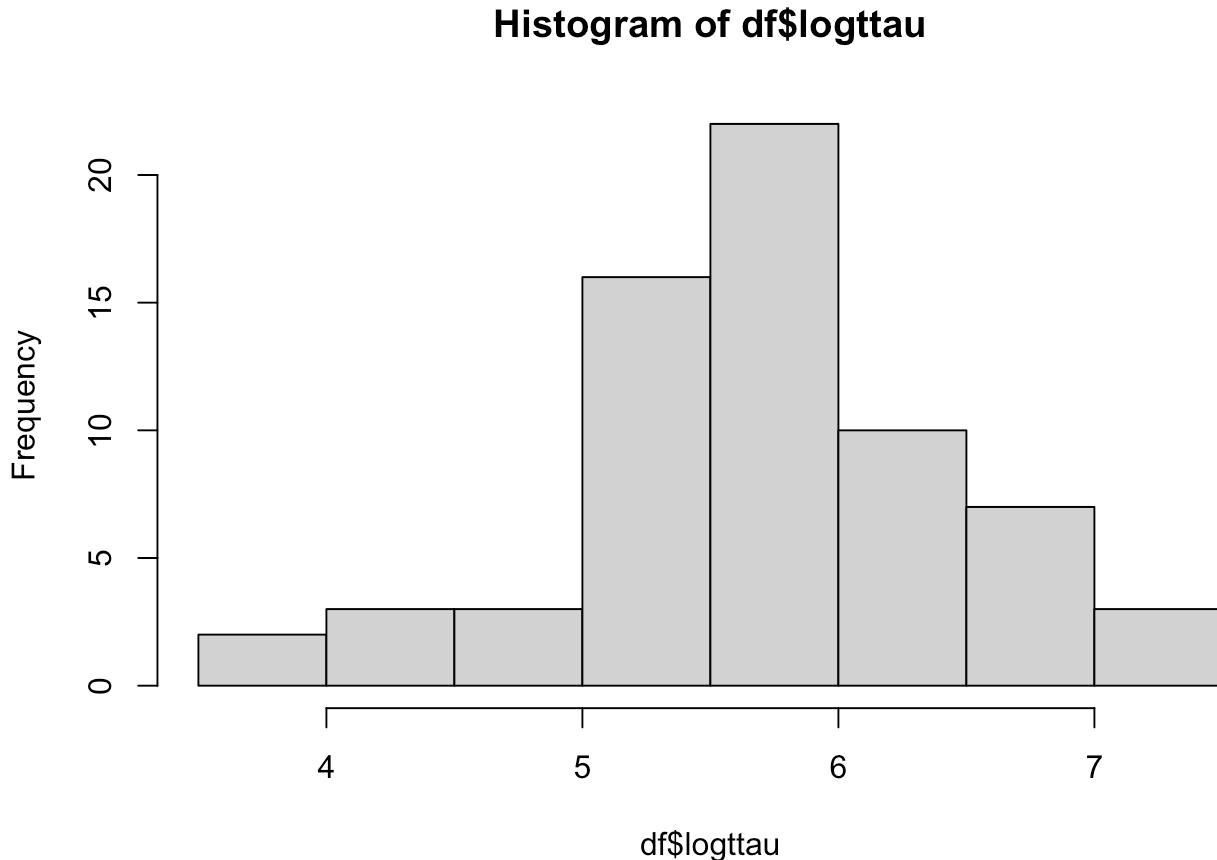
```
shapiro.test(dfttau[(dfttau$DX_APD == "CBS"), ]$logttau) #normal
```

```
##
## Shapiro-Wilk normality test
##
## data: dfttau[(dfttau$DX_APD == "CBS"), ]$logttau
## W = 0.96996, p-value = 0.4073
```

```
shapiro.test(dfttau[(dfttau$DX_APD == "PSP"), ]$logttau) #borderline
```

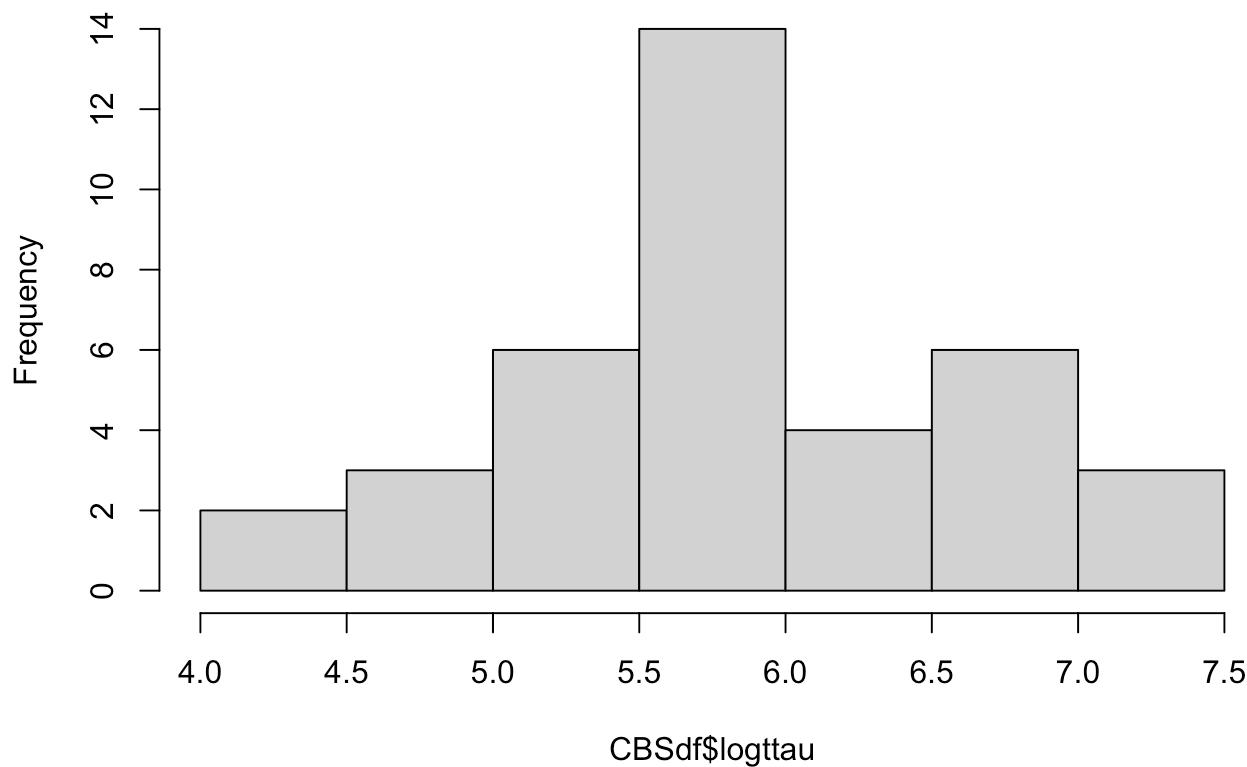
```
##  
## Shapiro-Wilk normality test  
##  
## data: dfttau[(dfttau$DX_APD == "PSP"), ]$logttau  
## W = 0.92868, p-value = 0.06416
```

```
hist(df$logttau)
```



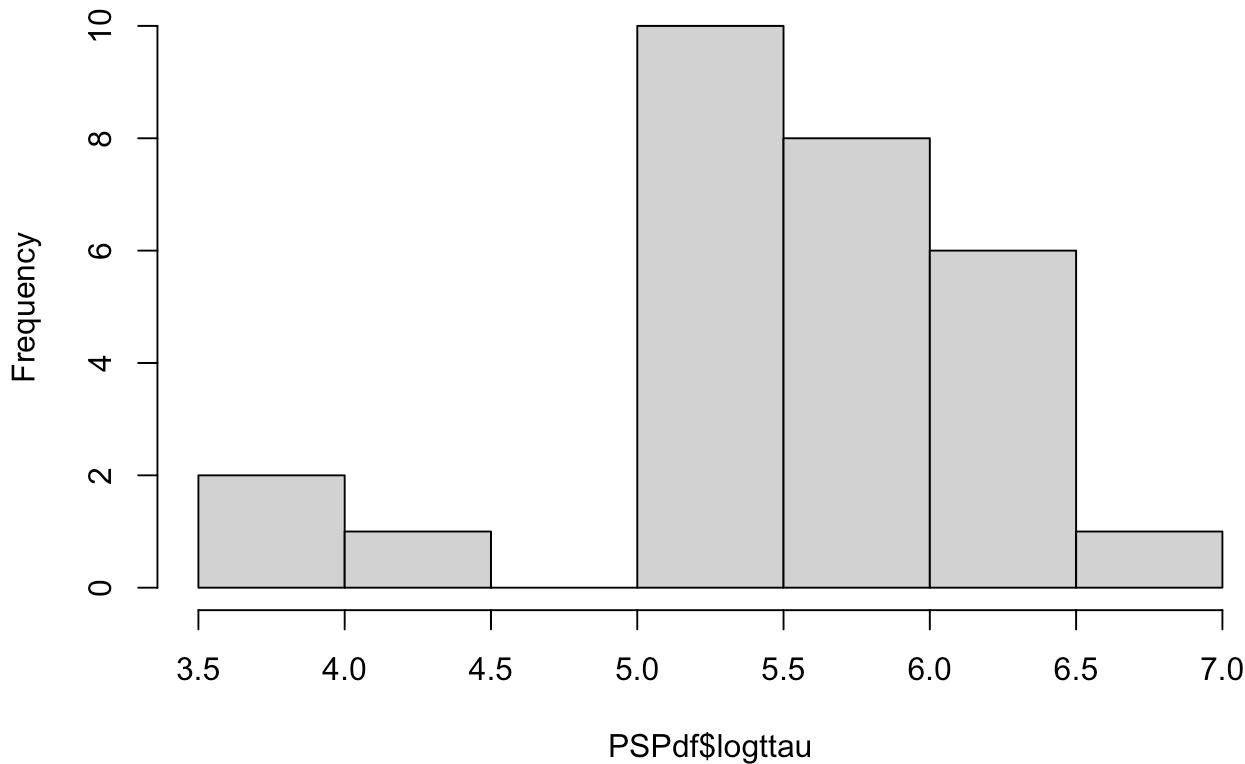
```
hist(CBSdf$logttau)
```

Histogram of CBSdf\$logtau



```
hist(PSPdf$logtau)
```

Histogram of PSPdf\$logtau



```
leveneTest(logtau ~ DX_APD, data = df) #homoscedasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.6627 0.4186
##       64
```

TAU STATISTICS: SUMMARY

```
# Choose mean because ran ANCOVA.
dfttau %>% summarize(count=n(), format(round(mean(ttau_2, na.rm=T),2),2), sd=sd(ttau_2,
na.rm=T)) #Rounds up the sd for some reason
```

```
##   count format(round(mean(ttau_2, na.rm = T), 2), 2)      sd
## 1    65                           377.64 277.9726
```

```
dfttau %>% group_by(DX_APD) %>% summarize(count=n(), format(round(mean(ttau_2, na.rm=T),
2),2), sd=sd(ttau_2, na.rm=T)) #Rounds up the sd for some reason
```

```
## # A tibble: 2 × 4
##   DX_APD count `format(round(mean(ttau_2, na.rm = T), 2), 2)`      sd
##   <chr>  <int> <chr>                                         <dbl>
## 1 CBS      38  444.45                                         326.
## 2 PSP      27  286.1                                         158.
```

```
sd(dfttau[(dfttau$DX_APD == "CBS"), ]$ttau_2, na.rm=T)
```

```
## [1] 326.1839
```

```
sd(dfttau[(dfttau$DX_APD == "PSP"), ]$ttau_2, na.rm=T)
```

```
## [1] 157.6613
```

T-TAU STATISTICS: ANCOVA

```
t.test(df$logtau ~ df$DX_APD, var.equal=TRUE)
```

```
##
## Two Sample t-test
##
## data: df$logtau by df$DX_APD
## t = 1.9107, df = 64, p-value = 0.06052
## alternative hypothesis: true difference in means between group CBS and group PSP is not equal to 0
## 95 percent confidence interval:
## -0.0158643 0.7128103
## sample estimates:
## mean in group CBS mean in group PSP
## 5.874890 5.526417
```

```
aov <- aov(logtau ~ Age + DX_APD, df)
Anova(aov, type="II")
```

```
## Anova Table (Type II tests)
##
## Response: logtau
##             Sum Sq Df F value    Pr(>F)
## Age          0.501  1  0.9327  0.33785
## DX_APD       2.438  1  4.5416  0.03699 *
## Residuals 33.817 63
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
check_normality(aov)
```

```
## OK: residuals appear as normally distributed (p = 0.400).
```

```
etaSquared(aov)
```

```
##          eta.sq eta.sq.part
## Age      0.01380163  0.01458896
## DX_APD  0.06720334  0.06724137
```

4.1.8. BIOMARKERS: ATI

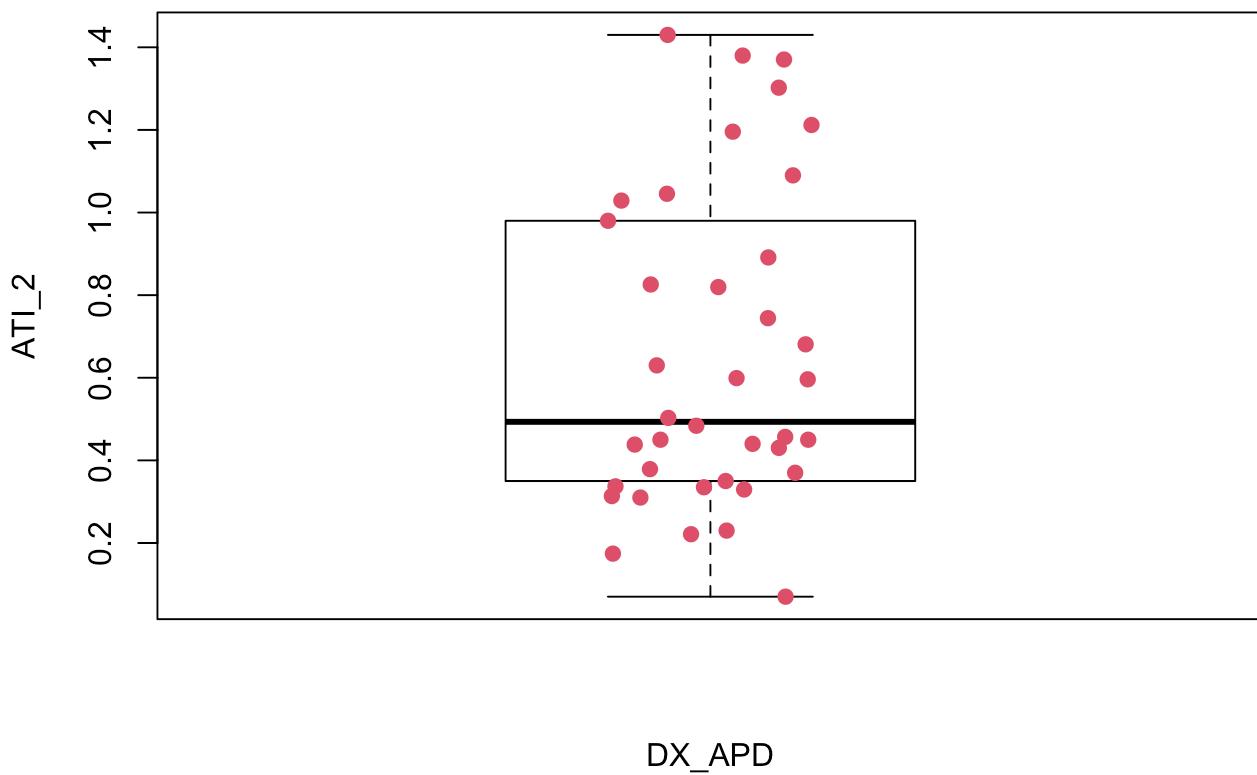
Comparisons shown in: Table 1

ATI STATISTICS: OUTLIERS

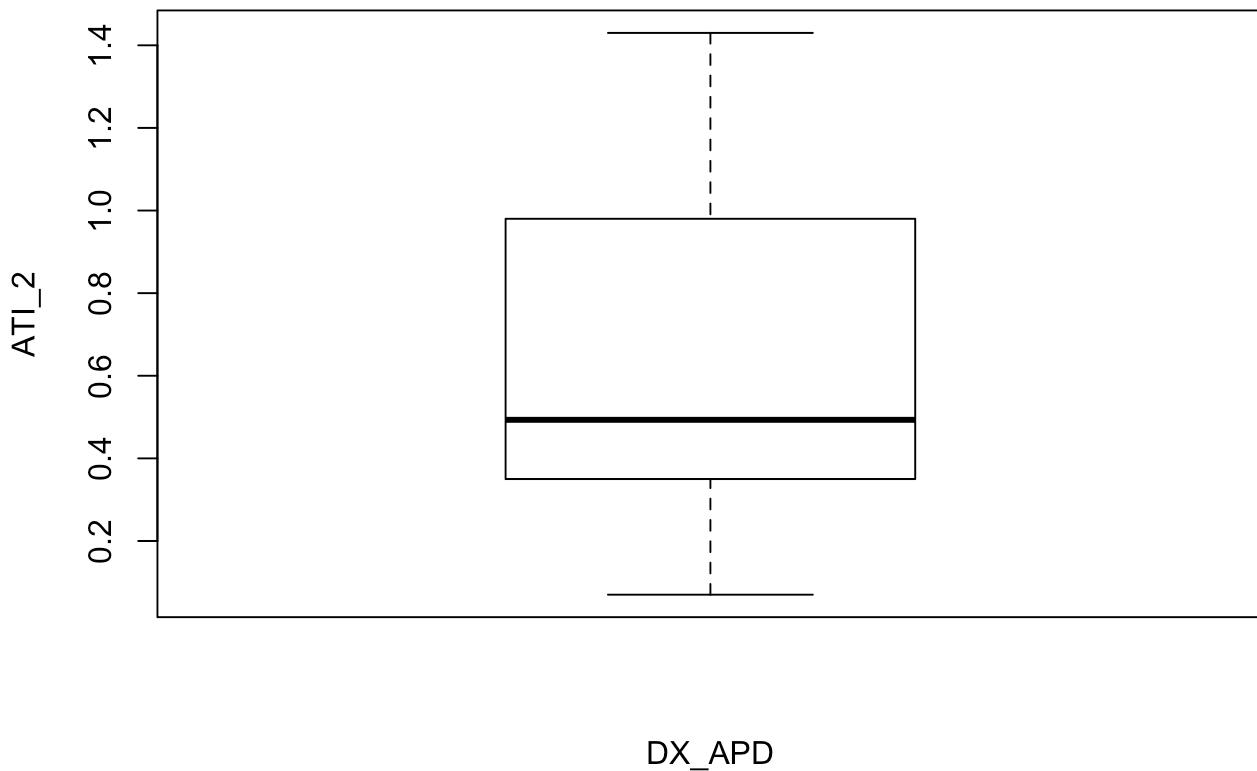
```
# First, display the data points of one group. boxplot() will identify outliers in given df but stripchart function can be funky hence convoluted script.
# Code below plots (boxplot() + assigns outliers to diagnosis-specific vector)
# Outliers are removed before calculating the values that are presented in table.
boxplot(ATI_2 ~ DX_APD, data= CBSdf, col = "white")$out #identify outliers in each diagnosis. First, look at CBS: there is one so attribute its value to vector.
```

```
## numeric(0)
```

```
stripchart(ATI_2 ~ DX_APD, data = CBSdf, method = "jitter", pch = 19, col = 2:4, vertical = TRUE, add = TRUE)
```



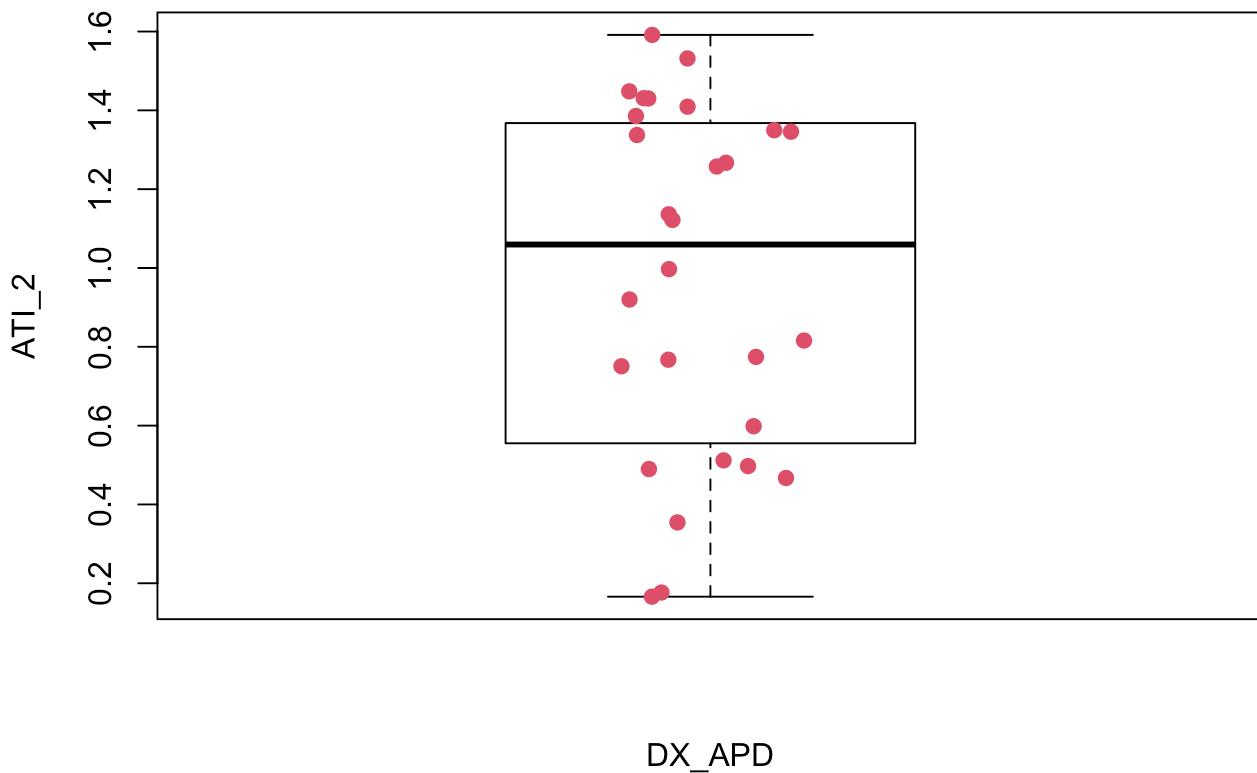
```
CBSvec.outliers <- boxplot(ATI_2 ~ DX_APD, data= CBSdf, col = "white")$out #list of outlier (Tukey method 1.5 IQR in each RTQUIC_DX combination)
```



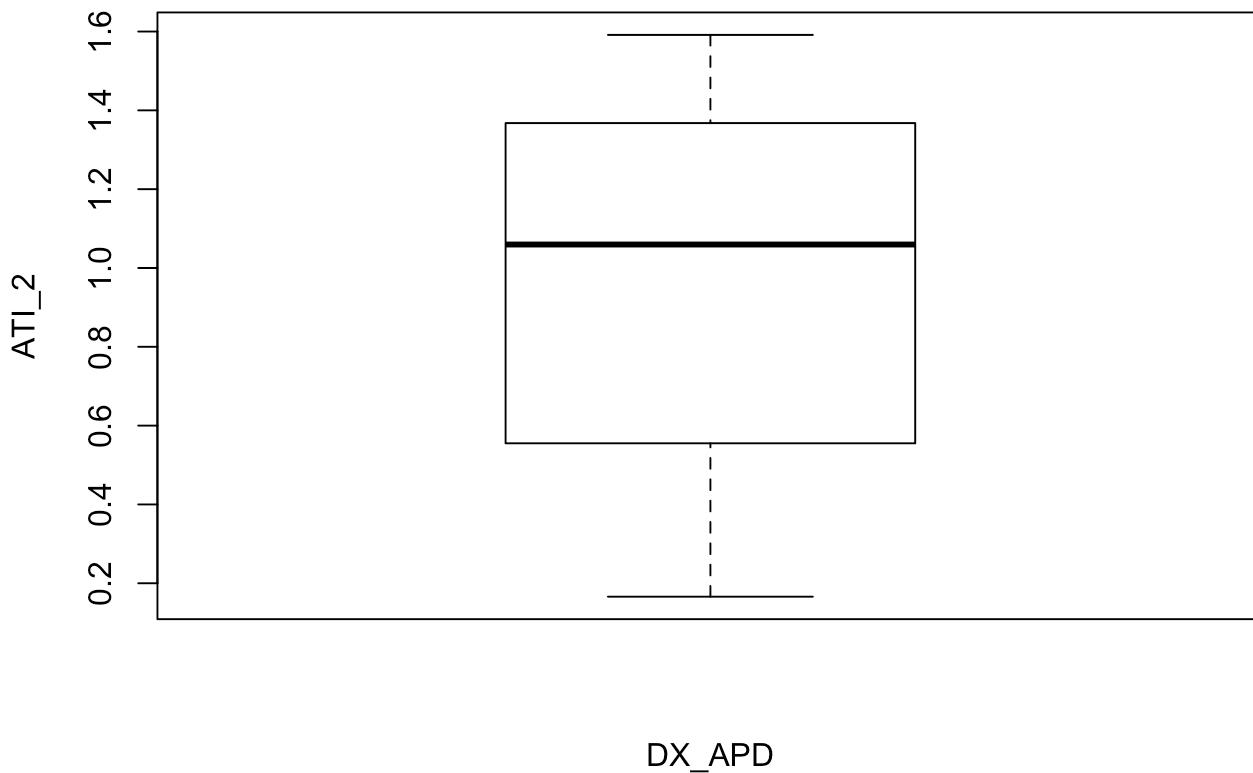
```
boxplot(ATI_2 ~ DX_APD, data= PSPdf, col = "white")$out #Now identify outliers in PSP: since there is none, no need to attribute to a vector.
```

```
## numeric(0)
```

```
stripchart(ATI_2 ~ DX_APD, data = PSPdf, method = "jitter", pch = 19, col = 2:4, vertical = TRUE, add = TRUE)
```



```
PSPvec.outliers <- boxplot(ATI_2 ~ DX_APD, data= PSPdf, col = "white")$out #list of outlier (Tukey method 1.5 IQR in each RTQUIC_DX combination)
```



DX_APD

```
dfati<- df[!(df$DX_APD=="CBS" & (df$ATI_2 %in% CBSvec.outliers)), ] %>% remove_empty("rows") %>% data.frame() #Removes all subjects who are CBS and either have no NFL value or one over the threshold
dfati<- dfati[!(dfati$DX_APD=="PSP" & (dfati$ATI_2 %in% PSPvec.outliers)), ] %>% remove_empty("rows") %>% data.frame() #Removes all subjects who are CBS and either have no NFL value or one over the threshold

if (nrow(dfati) == nrow(df)) {cat("No outliers were removed for ATI comparisons \n")} else {cat("The following outliers were removed from the CBS: ", CBSvec.outliers, " from the PSP: ", PSPvec.outliers, " .\n")}
```

```
## No outliers were removed for ATI comparisons
```

ATI STATISTICS: DISTRIBUTION

```
shapiro.test(CBSdf$ATI_2) #non-normal
```

```
##
## Shapiro-Wilk normality test
##
## data: CBSdf$ATI_2
## W = 0.91357, p-value = 0.006299
```

```
shapiro.test(PSPdf$ATI_2) #non-normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: PSPdf$ATI_2  
## W = 0.9203, p-value = 0.03529
```

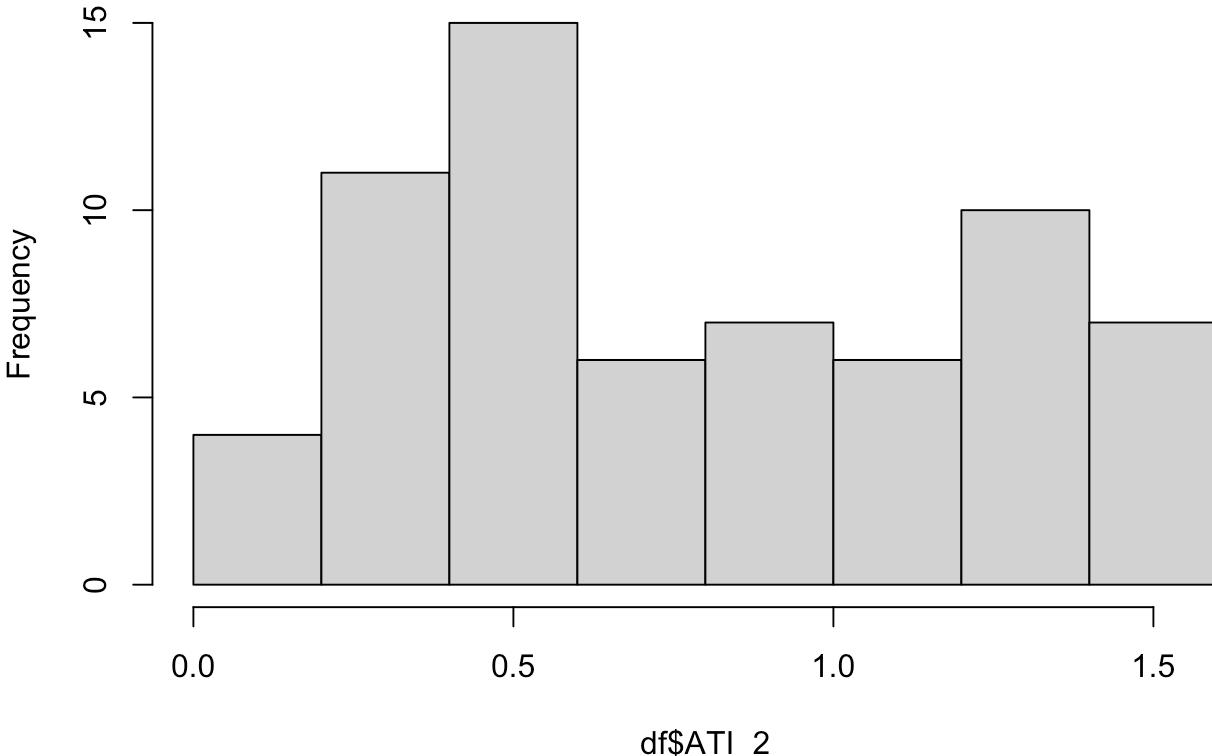
```
leveneTest(ATI_2 ~ DX_APD, data = df) #homoscedasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to  
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)  
## group 1 1.4151 0.2386  
## 64
```

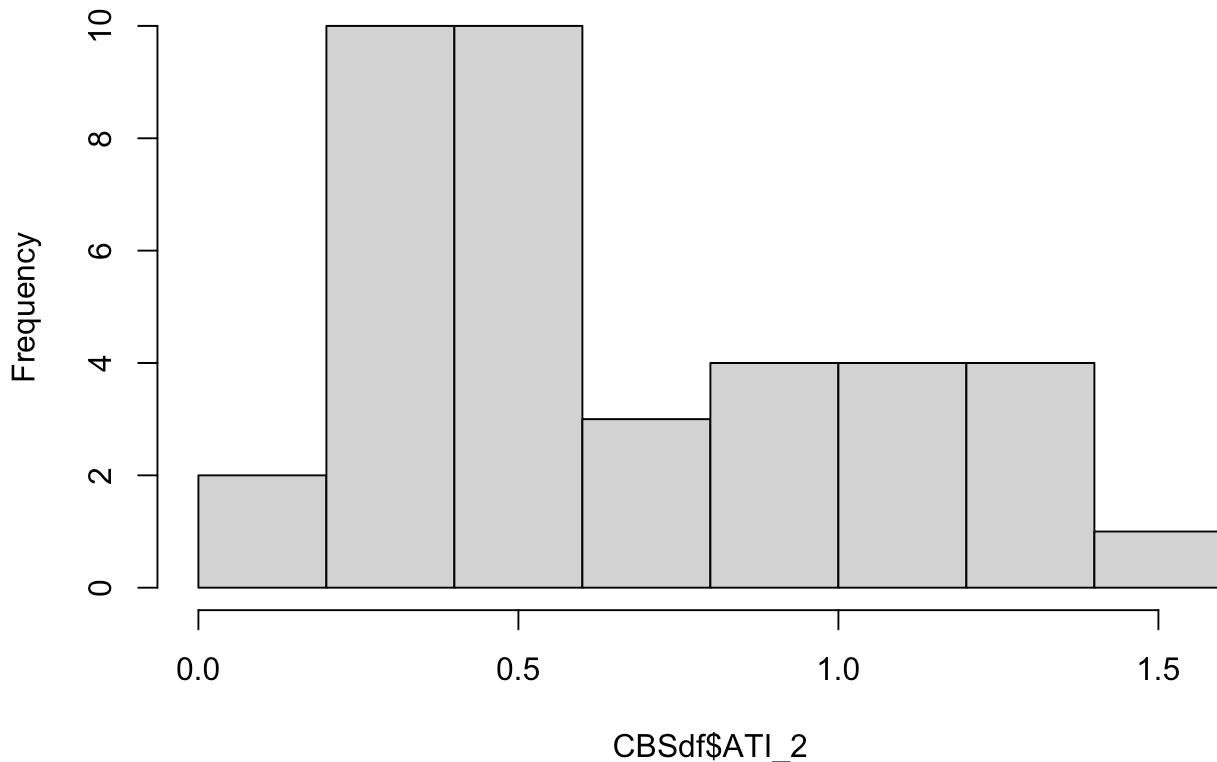
```
hist(df$ATI_2)
```

Histogram of df\$ATI_2



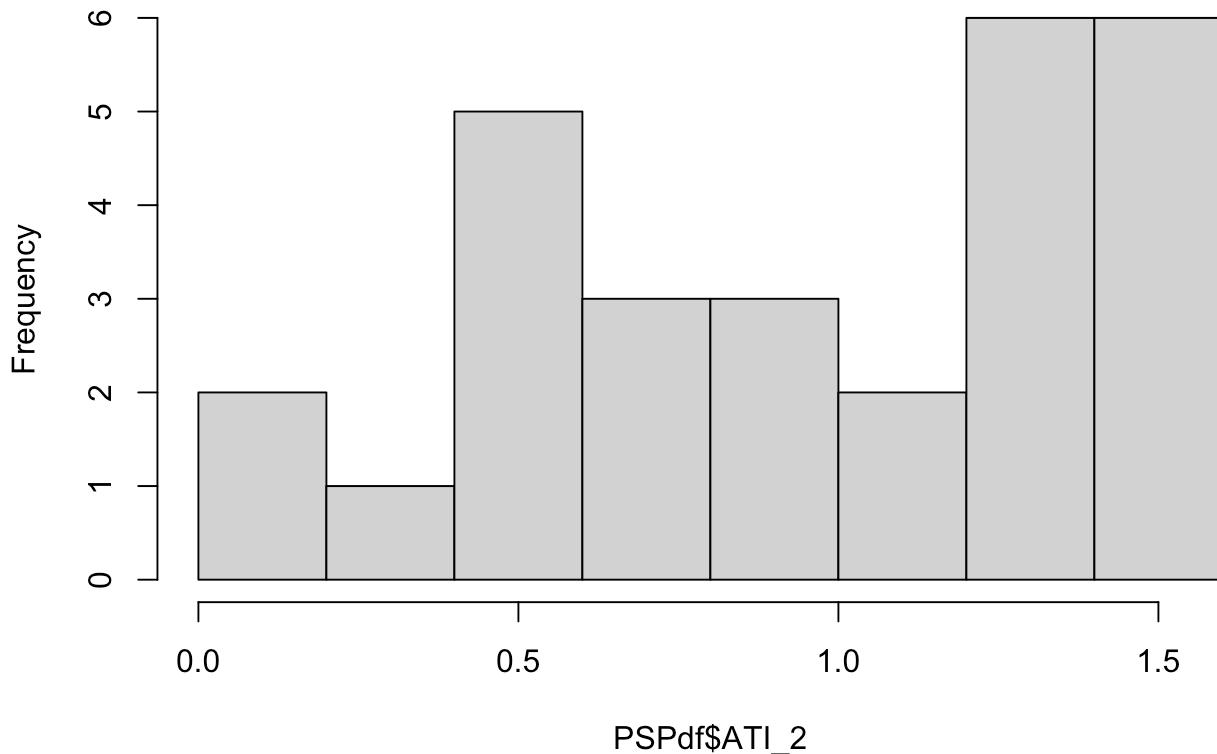
```
hist(CBSdf$ATI_2)
```

Histogram of CBSdf\$ATI_2



```
hist(PSPdf$ATI_2)
```

Histogram of PSPdf\$ATI_2



ATI STATISTICS: SUMMARY

```
# Choose mean because ran ANCOVA.
df %>% summarize(count=n(), format(round(mean(ATI_2, na.rm=T),2),2), sd=sd(ATI_2, na.rm=T)) #Rounds up the sd for some reason
```

```
##   count format(round(mean(ATI_2, na.rm = T), 2), 2)      sd
## 1     67                                         0.79  0.43534
```

```
df %>% group_by(DX_APD) %>% summarize(count=n(), format(round(mean(ATI_2, na.rm=T),2),2), sd=sd(ATI_2, na.rm=T)) #Rounds up the sd for some reason
```

```
## # A tibble: 2 × 4
##   DX_APD count `format(round(mean(ATI_2, na.rm = T), 2), 2)`      sd
##   <chr>  <int> <chr>                                         <dbl>
## 1 CBS      39  0.66                                         0.383
## 2 PSP      28  0.98                                         0.440
```

```
sd(CBSdf$ATI_2, na.rm=T)
```

```
## [1] 0.3831545
```

```
sd(PSPdf$ATI_2, na.rm=T)
```

```
## [1] 0.4399536
```

ATI STATISTICS: ANCOVA

```
wilcox.test(df$ATI_2 ~ df$DX_APD, paired=F)
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: df$ATI_2 by df$DX_APD
## W = 296, p-value = 0.002247
## alternative hypothesis: true location shift is not equal to 0
```

```
aov <- aov(ATI_2 ~ Age + DX_APD, df)
Anova(aov, type="II")
```

```
## Anova Table (Type II tests)
##
## Response: ATI_2
##             Sum Sq Df F value    Pr(>F)
## Age          0.4005  1 2.4600 0.12179
## DX_APD       0.9323  1 5.7262 0.01971 *
## Residuals 10.2574 63
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
check_normality(aov)
```

```
## OK: residuals appear as normally distributed (p = 0.426).
```

```
etaSquared(aov)
```

```
##              eta.sq eta.sq.part
## Age      0.03251295  0.03757962
## DX_APD  0.07568228  0.08331893
```

4.1.9. BIOMARKERS: NFL

Comparisons shown in: Table 1

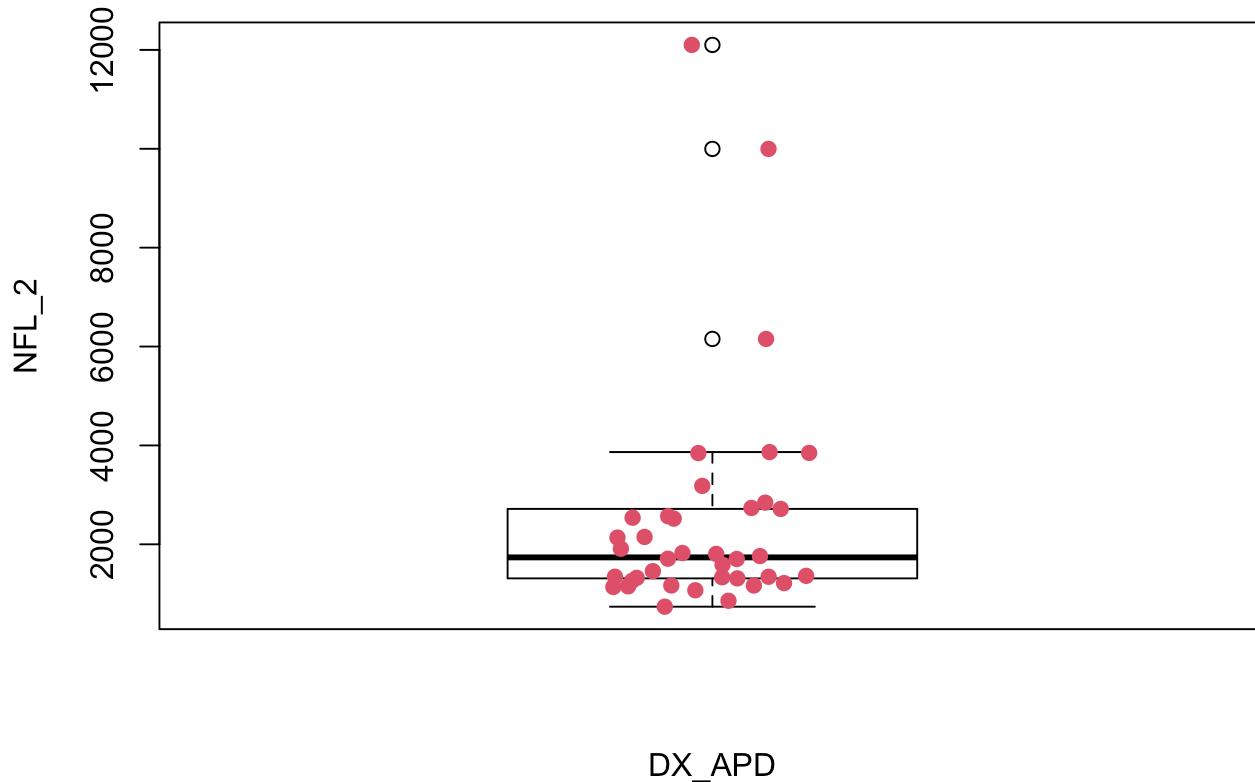
NFL STATISTICS: OUTLIERS

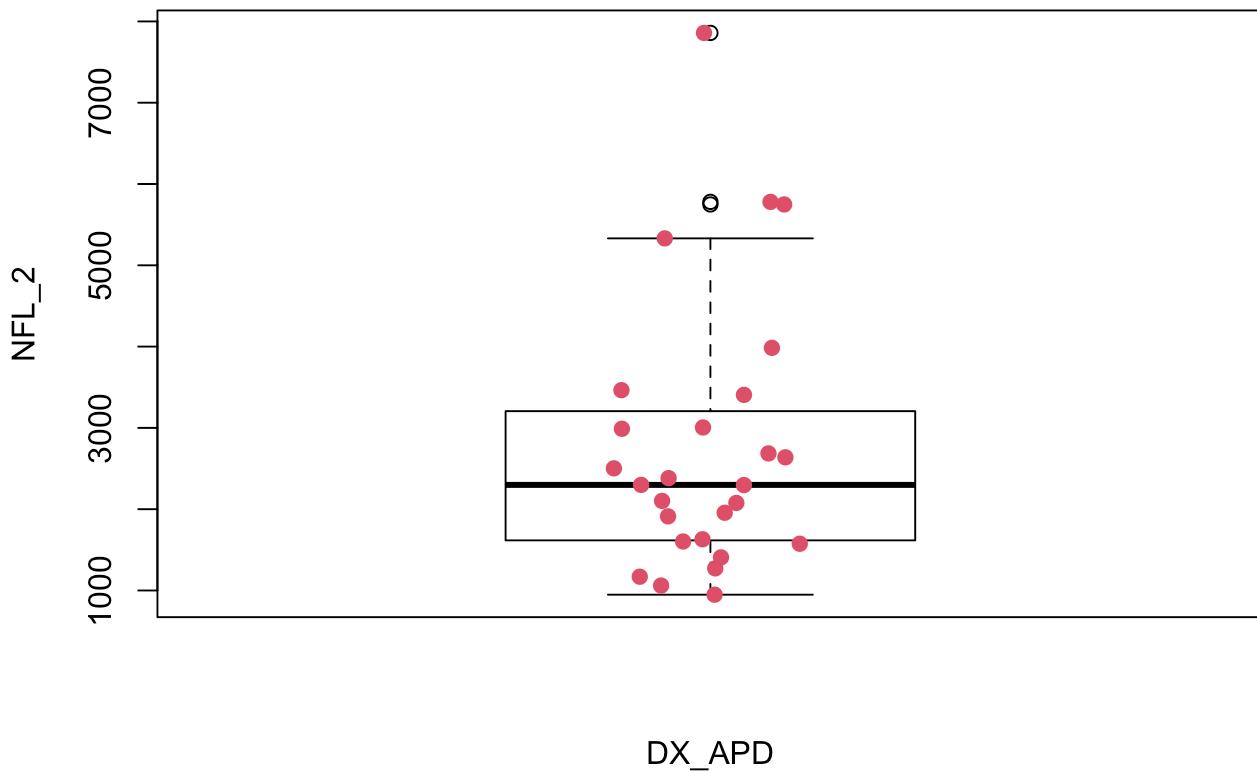
```
# NFL values are typically right skewed especially in FTLD-related diagnoses. Therefore,
different approach for outlier identification was chosen.

# Remove outliers over full dataset, but at a tolerant threshold (Q3+3*IQR instead of 1.
5 IQR. Reference for this is: https://www.nature.com/articles/s41598-020-66090-x
# For reference, outliers are added to the notes of Tables.
boxplot(NFL_2 ~ DX_APD, data= CBSdf, col = "white")$out #identify outliers in each diagnosis. First, look at CBS: there is one so attribute its value to vector.
```

```
## [1] 12101.20 6153.82 9997.64
```

```
stripchart(NFL_2 ~ DX_APD, data = CBSdf, method = "jitter", pch = 19, col = 2:4, vertical = TRUE, add = TRUE)
```





DX_APD

```
thresholdCBS <- min(max(CBSdf$NFL_2,na.rm=T), as.numeric(quantile(CBSdf$NFL_2, 0.75, na.rm=T)) + (IQR(na.rm=T, (CBSdf$NFL_2)*3))) #reports the value Q3+ IQR*3 (3 is very tolerant threshold)
thresholdPSP <- min(max(PSPdf$NFL_2,na.rm=T), as.numeric(quantile(PSPdf$NFL_2, 0.75, na.rm=T)) + (IQR(na.rm=T, (PSPdf$NFL_2)*3))) #reports the value Q3+ IQR*3 (3 is very tolerant threshold)
cat("Outliers are values above ", thresholdCBS, " in CBS subset. \n")
```

```
## Outliers are values above 6775.568 in CBS subset.
```

```
cat("Outliers are values above ", thresholdPSP, " in PSP subset. \n")
```

```
## Outliers are values above 7858.61 in PSP subset.
```

```
dfnfl<- df[df$DX_APD=="PSP" | df$NFL_2 <= thresholdCBS, ] %>% remove_empty("rows") %>% data.frame() #Removes all subjects who are CBS and either have no NFL value or one over the threshold
dfnfl<- dfnfl[dfnfl$DX_APD=="CBS" | dfnfl$NFL_2 <= thresholdPSP, ] %>% remove_empty("rows") %>% data.frame() #Removes all subjects who are PSP and either have no NFL value or one over the threshold

removed <- setdiff(df, dfnfl)
cat("Following values were removed for the descriptive stats on NfL: ", removed$NFL_2,
"\n")
```

```
## Following values were removed for the descriptive stats on NfL: 12101.2 NA NA 9997.6
4
```

NFL STATISTICS: DISTRIBUTION

```
shapiro.test(dfnfl[dfnfl$DX_APD=="CBS", ]$logNFL) #normal
```

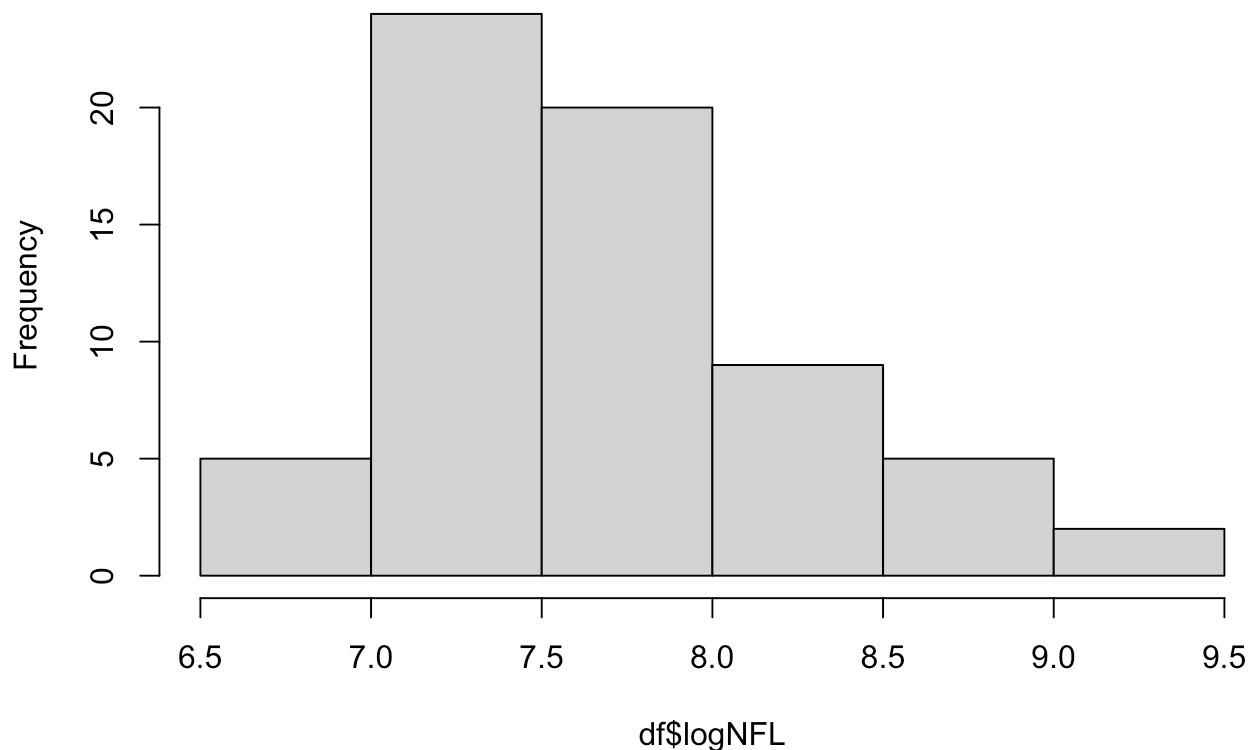
```
##
## Shapiro-Wilk normality test
##
## data: dfnfl[dfnfl$DX_APD == "CBS", ]$logNFL
## W = 0.96261, p-value = 0.2584
```

```
shapiro.test(dfnfl[dfnfl$DX_APD=="PSP", ]$logNFL) #normal
```

```
##
## Shapiro-Wilk normality test
##
## data: dfnfl[dfnfl$DX_APD == "PSP", ]$logNFL
## W = 0.97556, p-value = 0.7516
```

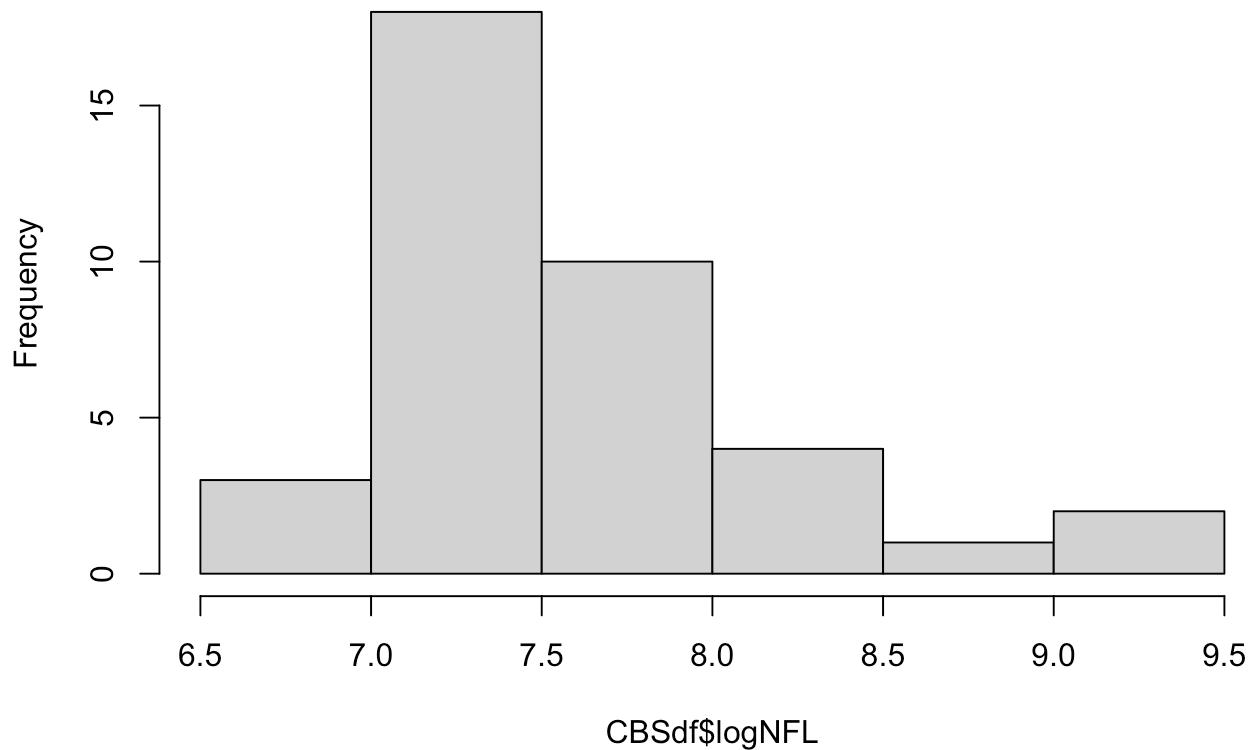
```
hist(df$logNFL)
```

Histogram of df\$logNFL



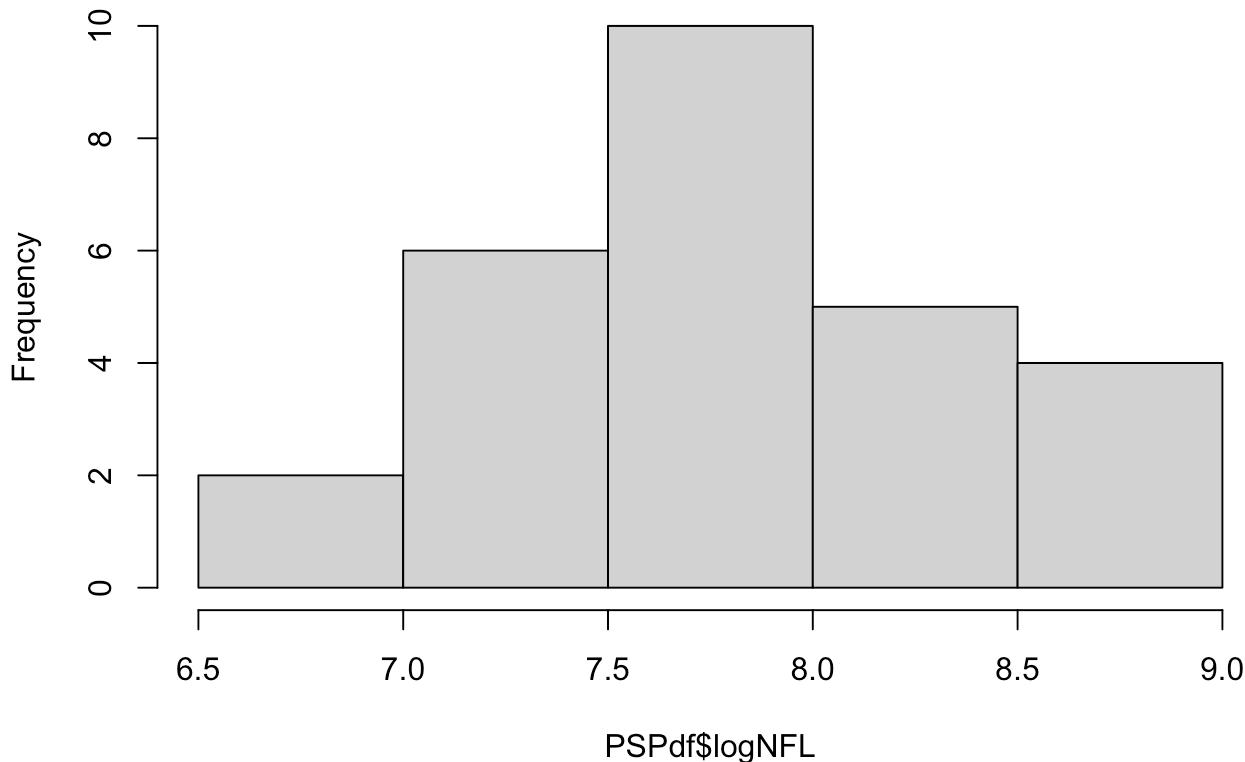
```
hist(CBSdf$logNFL)
```

Histogram of CBSdf\$logNFL



```
hist(PSPdf$logNFL)
```

Histogram of PSPdf\$logNFL



```
leveneTest(logNFL ~ DX_APD, data = dfnfl) #homoscedasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.2413  0.625
##       61
```

NFL STATISTICS: SUMMARY

```
dfnfl %>% summarize(count=n(), format(round(mean(NFL_2, na.rm=T),2),2), sd=sd(NFL_2, na.rm=T)) #Rounds up the sd for some reason
```

```
##   count format(round(mean(NFL_2, na.rm = T), 2), 2)      sd
## 1    63                           2344.74 1418.081
```

```
dfnfl %>% group_by(DX_APD) %>% summarize(count=n(), format(round(mean(NFL_2, na.rm=T), 2),2), sd=sd(NFL_2, na.rm=T)) #Rounds up the sd for some reason
```

```
## # A tibble: 2 × 4
##   DX_APD count `format(round(mean(NFL_2, na.rm = T), 2), 2)` sd
##   <chr>  <int> <chr>
## 1 CBS      36  2017.59 1106.
## 2 PSP      27  2780.95 1674.
```

```
sd(dfnfl[dfnfl$DX_APD=="CBS", ]$NFL_2)
```

```
## [1] 1106.203
```

```
sd(dfnfl[dfnfl$DX_APD=="PSP", ]$NFL_2)
```

```
## [1] 1674.006
```

NFL STATISTICS: ANCOVA

```
t.test(dfnfl$logNFL ~ dfnfl$DX_APD, var.equal=TRUE)
```

```
##
## Two Sample t-test
##
## data: dfnfl$logNFL by dfnfl$DX_APD
## t = -2.2729, df = 61, p-value = 0.02657
## alternative hypothesis: true difference in means between group CBS and group PSP is not equal to 0
## 95 percent confidence interval:
## -0.54668377 -0.03496251
## sample estimates:
## mean in group CBS mean in group PSP
## 7.492641 7.783464
```

```
aov <- aov(logNFL ~ Age + DX_APD, dfnfl)
Anova(aov, type="II")
```

```
## Anova Table (Type II tests)
##
## Response: logNFL
##             Sum Sq Df F value Pr(>F)
## Age          0.0394  1  0.1537 0.6965
## DX_APD       0.9835  1  3.8396 0.0547 .
## Residuals 15.3693 60
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
check_normality(aov)
```

```
## OK: residuals appear as normally distributed (p = 0.296).
```

```
etaSquared(aov)
```

```
##          eta.sq eta.sq.part
## Age      0.002354985 0.002554422
## DX_APD  0.058847046 0.060145167
```

4.2. COHORT CHARACTERISTICS: CATEGORICAL VARIABLES

For categorical variables, when an association between 2 variables was tested, the table of contingency was computed, and if the expected count was under 5 for any of the 4 cells, the Fisher test was chosen over the Chi-square.

4.2.1. SEX

Comparisons shown in: Table 1, eTable 1, eTable 2, Results > Cohort characteristics

SEX STATISTICS: SUMMARY

```
cat("Sex distribution in the dataset is: \n")
```

```
## Sex distribution in the dataset is:
```

```
df %>% group_by(DX_APD) %>% count(Sex)
```

```
## # A tibble: 4 × 3
## # Groups:   DX_APD [2]
##   DX_APD Sex     n
##   <chr>  <chr> <int>
## 1 CBS    F        19
## 2 CBS    M        20
## 3 PSP    F        13
## 4 PSP    M        15
```

SEX STATISTICS: PERCENTAGES

```
totalmatrix <- df %>% count(DX_APD)
sexmatrix <- df %>% group_by(DX_APD) %>% count(Sex)

cat("Proportion of females in total is: \n")
```

```
## Proportion of females in total is:
```

```
(as.numeric(sexmatrix$n[1]) + as.numeric(sexmatrix$n[3]))/(as.numeric(totalmatrix$n[1])
+ as.numeric(totalmatrix$n[2]))
```

```
## [1] 0.4776119
```

```
cat("Proportion of females in CBS is:")
```

```
## Proportion of females in CBS is:
```

```
as.numeric(sexmatrix$n[1])/as.numeric(totalmatrix$n[1])
```

```
## [1] 0.4871795
```

```
cat("Proportion of females in PSP is:")
```

```
## Proportion of females in PSP is:
```

```
as.numeric(sexmatrix$n[3])/as.numeric(totalmatrix$n[2])
```

```
## [1] 0.4642857
```

SEX STATISTICS: CHI-SQUARE

```
table(df$Sex, df$DX_APD)
```

```
##  
##      CBS  PSP  
##      F   19   13  
##      M   20   15
```

```
chisq.test(table(df$Sex, df$DX_APD), correct=F)
```

```
##  
## Pearson's Chi-squared test  
##  
## data: table(df$Sex, df$DX_APD)  
## X-squared = 0.034238, df = 1, p-value = 0.8532
```

4.2.2. APOE

Comparisons shown in: Table 1, eTable 1, eTable 2, Results > Cohort characteristics

APOE STATISTICS: SUMMARY

```
df %>% group_by(DX_APD) %>% count(AP0Ee4)
```

```
## # A tibble: 6 × 3
## # Groups:   DX_APD [2]
##   DX_APD AP0Ee4      n
##   <chr>   <fct>    <int>
## 1 CBS     Negative    30
## 2 CBS     Positive     7
## 3 CBS     <NA>        2
## 4 PSP     Negative    17
## 5 PSP     Positive     7
## 6 PSP     <NA>        4
```

```
table(df$AP0Ee4, df$DX_APD)
```

```
##
##          CBS PSP
## Negative 30 17
## Positive 7  7
```

APOE STATISTICS: SUMMARY

```
chisq.test(table(df$AP0Ee4, df$DX_APD), correct=F)
```

```
##
## Pearson's Chi-squared test
##
## data: table(df$AP0Ee4, df$DX_APD)
## X-squared = 0.86452, df = 1, p-value = 0.3525
```

4.2.3. AD

Comparisons shown in: Table 1, eTable 1, eTable 2, Results > Cohort characteristics

AD STATISTICS: SUMMARY

```
df %>% group_by(DX_APD) %>% count(AD)
```

```
## # A tibble: 4 × 3
## # Groups:   DX_APD [2]
##   DX_APD AD      n
##   <chr>   <chr>  <int>
## 1 CBS     AD Negative  27
## 2 CBS     AD Positive  12
## 3 PSP     AD Negative  25
## 4 PSP     AD Positive   3
```

```
table(df$DX_APD, df$AD)
```

```
##  
##      AD Negative AD Positive  
##    CBS          27          12  
##    PSP          25           3
```

AD STATISTICS: CHI-SQUARE

```
chisq.test(table(df$AD, df$DX_APD), correct=F)
```

```
##  
## Pearson's Chi-squared test  
##  
## data: table(df$AD, df$DX_APD)  
## X-squared = 3.7726, df = 1, p-value = 0.0521
```

5. ASYN-SAA+ & COHORT CHARACTERISTICS

Comparisons shown in: eTable 1, Results > ASyn-SAA+ & Demographics

```
df %>% group_by(RTQUIC) %>% count(DX_APD)
```

```
## # A tibble: 4 × 3  
## # Groups:   RTQUIC [2]  
##   RTQUIC             DX_APD     n  
##   <fct>            <chr>    <int>  
## 1 aSyn-SAA negative CBS      25  
## 2 aSyn-SAA negative PSP      20  
## 3 aSyn-SAA positive CBS     14  
## 4 aSyn-SAA positive PSP      8
```

```
sum(df$DX_APD=="CBS" & df$RTQUIC=="aSyn-SAA positive" & df$Sex=="F")
```

```
## [1] 9
```

```
sum(df$DX_APD=="PSP" & df$RTQUIC=="aSyn-SAA positive" & df$Sex=="F")
```

```
## [1] 3
```

5.1. ASYN-SAA+ COHORT CHARACTERISTICS: CATEGORICAL VARIABLES

Comparisons shown in: eTable 1, Results > ASyn-SAA+ & Demographics For comparisons within CBS and within PSP, go lower, to 8.

SEX STATISTICS: DISTRIBUTION

```
table(df$Sex, df$RTQUIC)
```

```
##  
##      aSyn-SAA negative aSyn-SAA positive  
##      F                  20                  12  
##      M                  25                  10
```

SEX STATISTICS: CHI-SQUARE

```
chisq.test(table(df$Sex, df$RTQUIC), correct=F)
```

```
##  
## Pearson's Chi-squared test  
##  
## data: table(df$Sex, df$RTQUIC)  
## X-squared = 0.60426, df = 1, p-value = 0.437
```

APOE STATISTICS: DISTRIBUTION

```
table(df$AP0Ee4, df$RTQUIC)
```

```
##  
##      aSyn-SAA negative aSyn-SAA positive  
##      Negative          31                  16  
##      Positive           11                   3
```

APOE STATISTICS: FISHER'S TEST

```
fisher.test(table(df$AP0Ee4, df$RTQUIC)) # Expected count is <5 for one cell
```

```

## Fisher's Exact Test for Count Data
##
## data: table(df$AP0Ee4, df$RTQUIC)
## p-value = 0.5164
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.08370541 2.43483909
## sample estimates:
## odds ratio
## 0.5335897

```

PARK ONSET STATISTICS: DISTRIBUTION

```
table(df$Parkinsonian_onset, df$RTQUIC)
```

	aSyn-SAA negative	aSyn-SAA positive
No	23	12
Yes	22	10

PARK ONSET STATISTICS: FISHER'S TEST

```
chisq.test(table(df$Parkinsonian_onset, df$RTQUIC), correct=F)
```

## Pearson's Chi-squared test	
##	
## data: table(df\$Parkinsonian_onset, df\$RTQUIC)	
## X-squared = 0.069852, df = 1, p-value = 0.7916	

5.2. ASYN-SAA+ COHORT CHARACTERISTICS: NUMERICAL VARIABLES

BONFERRONI CALCULATION FOR AGE/ONSET AGE/PARKINSONISM AGE COMPARISONS ATTRIBUTABLE TO RTQUIC Looking at three similar tests: Difference in mean age at LP, onset, and Parkinsonism onset, in RT-QUIC+ vs RT-QUIC-

```
0.05/3
```

```
## [1] 0.01666667
```

5.2.1. AGE

Comparisons shown in: eTable 1, Results > ASyn-SAA+ & Demographics

ASYN-SAA*AGE STATISTICS: DISTRIBUTION

```
shapiro.test(RTposdf$Age) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: RTposdf$Age  
## W = 0.96192, p-value = 0.5291
```

```
shapiro.test(RTnegdf$Age) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: RTnegdf$Age  
## W = 0.98166, p-value = 0.6873
```

```
var.test(Age ~ RTQUIC, data = df) #homoscedasticity
```

```
##  
## F test to compare two variances  
##  
## data: Age by RTQUIC  
## F = 1.5408, num df = 44, denom df = 21, p-value = 0.286  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.6912078 3.1070414  
## sample estimates:  
## ratio of variances  
## 1.540823
```

ASYN-SAA*AGE STATISTICS: SUMMARY

```
df %>% group_by(RTQUIC) %>% summarize(count=n(), mean=mean(Age, na.rm=T), sd=sd(Age, na.rm=T))
```

```
## # A tibble: 2 × 4  
##   RTQUIC           count   mean     sd  
##   <fct>         <int> <dbl>  <dbl>  
## 1 aSyn-SAA negative     45  67.9  9.75  
## 2 aSyn-SAA positive      22  70.2  7.85
```

ASYN-SAA*AGE STATISTICS: ANCOVA

```
t.test(df$Age ~ df$RTQUIC, var.equal=TRUE) #What is reported in eTable 1
```

```
## 
## Two Sample t-test
##
## data: df$Age by df$RTQUIC
## t = -0.93886, df = 65, p-value = 0.3513
## alternative hypothesis: true difference in means between group aSyn-SAA negative and
## group aSyn-SAA positive is not equal to 0
## 95 percent confidence interval:
## -7.011610 2.527332
## sample estimates:
## mean in group aSyn-SAA negative mean in group aSyn-SAA positive
## 67.93632 70.17846
```

```
aov <- aov(Age ~ DX_APD*RTQUIC, df)
Anova(aov, type="III")
```

```
## Warning in printHypothesis(L, rhs, names(b)): one or more coefficients in the hypothesis include
## arithmetic operators in their names;
## the printed representation of the hypothesis will be omitted

## Warning in printHypothesis(L, rhs, names(b)): one or more coefficients in the hypothesis include
## arithmetic operators in their names;
## the printed representation of the hypothesis will be omitted
```

```
## Anova Table (Type III tests)
##
## Response: Age
##             Sum Sq Df  F value    Pr(>F)
## (Intercept) 104781  1 1397.4393 < 2.2e-16 ***
## DX_APD       575   1    7.6656  0.007383 **
## RTQUIC        97   1    1.2889  0.260547
## DX_APD:RTQUIC  6   1    0.0739  0.786559
## Residuals    4724  63
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
etaSquared(aov)
```

```
##                   eta.sq eta.sq.part
## DX_APD          0.1347944842 0.136760851
## RTQUIC          0.0206138434 0.023654887
## DX_APD:RTQUIC  0.0009987086 0.001172432
```

p=.007 < .017 so reported bonf-adjusted p-value: .007x3=.02

5.2.2. ONSET

Comparisons shown in: eTable 1, Results > ASyn-SAA+ & Demographics

ASYN-SAA*AGE STATISTICS: DISTRIBUTION

```
shapiro.test(RTposdf$Onset_age) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: RTposdf$Onset_age  
## W = 0.96189, p-value = 0.5285
```

```
shapiro.test(RTnegdf$Onset_age) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: RTnegdf$Onset_age  
## W = 0.96942, p-value = 0.2758
```

```
var.test(Onset_age ~ RTQUIC, data = df) #homoscedasticity
```

```
##  
## F test to compare two variances  
##  
## data: Onset_age by RTQUIC  
## F = 1.5861, num df = 44, denom df = 21, p-value = 0.2547  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.7115405 3.1984383  
## sample estimates:  
## ratio of variances  
## 1.586148
```

ASYN-SAA*AGE STATISTICS: SUMMARY

```
df %>% group_by(RTQUIC) %>% summarize(count=n(), mean=mean(Onset_age, na.rm=T), sd=sd(Onset_age, na.rm=T))
```

```
## # A tibble: 2 × 4  
##   RTQUIC           count   mean     sd  
##   <fct>         <int> <dbl> <dbl>  
## 1 aSyn-SAA negative     45  62.2  9.07  
## 2 aSyn-SAA positive      22  65.9  7.20
```

ASYN-SAA*AGE STATISTICS: ANCOVA

```
t.test(df$Onset_age ~ df$RTQUIC, var.equal=TRUE) #What is reported in eTable 1
```

```
## 
## Two Sample t-test
##
## data: df$Onset_age by df$RTQUIC
## t = -1.645, df = 65, p-value = 0.1048
## alternative hypothesis: true difference in means between group aSyn-SAA negative and
## group aSyn-SAA positive is not equal to 0
## 95 percent confidence interval:
## -8.0623494 0.7795211
## sample estimates:
## mean in group aSyn-SAA negative mean in group aSyn-SAA positive
## 62.22222 65.86364
```

```
aov <- aov(Onset_age ~ DX_APD*RTQUIC, df)
Anova(aov, type="III")
```

```
## Warning in printHypothesis(L, rhs, names(b)): one or more coefficients in the hypothesis include
## arithmetic operators in their names;
## the printed representation of the hypothesis will be omitted

## Warning in printHypothesis(L, rhs, names(b)): one or more coefficients in the hypothesis include
## arithmetic operators in their names;
## the printed representation of the hypothesis will be omitted
```

```
## Anova Table (Type III tests)
##
## Response: Onset_age
##             Sum Sq Df  F value Pr(>F)
## (Intercept) 90360  1 1337.7303 <2e-16 ***
## DX_APD      249   1   3.6802 0.0596 .
## RTQUIC      107   1   1.5827 0.2130
## DX_APD:RTQUIC    9   1   0.1280 0.7217
## Residuals   4255  63
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
etaSquared(aov)
```

```
##                  eta.sq eta.sq.part
## DX_APD          0.090208763 0.094137139
## RTQUIC          0.049476840 0.053923494
## DX_APD:RTQUIC 0.001763424 0.002027334
```

p >.017 so no need to bonferroni correct as it is ns

5.2.3. PARK ONSET

Comparisons shown in: eTable 1, Results > ASyn-SAA+ & Demographics

ASYN-SAA*PARK ONSET STATISTICS: DISTRIBUTION

```
shapiro.test(RTposdf$Park_onset) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: RTposdf$Park_onset  
## W = 0.93718, p-value = 0.1731
```

```
shapiro.test(RTnegdf$Park_onset) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: RTnegdf$Park_onset  
## W = 0.96579, p-value = 0.2133
```

```
var.test(Park_onset ~ RTQUIC, data = df) #homoscedasticity
```

```
##  
## F test to compare two variances  
##  
## data: Park_onset by RTQUIC  
## F = 1.7798, num df = 43, denom df = 21, p-value = 0.1559  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.7969689 3.6030251  
## sample estimates:  
## ratio of variances  
## 1.779805
```

ASYN-SAA*PARK ONSET STATISTICS: SUMMARY

```
df %>% group_by(RTQUIC) %>% summarize(count=n(), mean=mean(Park_onset, na.rm=T), sd=sd(Park_onset, na.rm=T))
```

```
## # A tibble: 2 × 4  
##   RTQUIC           count   mean     sd  
##   <fct>         <int> <dbl>  <dbl>  
## 1 aSyn-SAA negative     45  63.6   8.97  
## 2 aSyn-SAA positive      22  67.3   6.72
```

ASYN-SAA*PARK ONSET STATISTICS: ANCOVA

```
t.test(df$Park_onset ~ df$RTQUIC, var.equal=TRUE)
```

```
## 
## Two Sample t-test
##
## data: df$Park_onset by df$RTQUIC
## t = -1.6787, df = 64, p-value = 0.09809
## alternative hypothesis: true difference in means between group aSyn-SAA negative and
## group aSyn-SAA positive is not equal to 0
## 95 percent confidence interval:
## -7.9638087 0.6910815
## sample estimates:
## mean in group aSyn-SAA negative mean in group aSyn-SAA positive
## 63.63636 67.27273
```

```
aov <- aov(Park_onset ~ DX_APD*RTQUIC, df)
Anova(aov, type="III")
```

```
## Warning in printHypothesis(L, rhs, names(b)): one or more coefficients in the hypothesis include
## arithmetic operators in their names;
## the printed representation of the hypothesis will be omitted

## Warning in printHypothesis(L, rhs, names(b)): one or more coefficients in the hypothesis include
## arithmetic operators in their names;
## the printed representation of the hypothesis will be omitted
```

```
## Anova Table (Type III tests)
##
## Response: Park_onset
##             Sum Sq Df  F value    Pr(>F)
## (Intercept) 91513  1 1382.1001 < 2e-16 ***
## DX_APD       188   1    2.8375 0.09711 .
## RTQUIC       129   1    1.9501 0.16755
## DX_APD:RTQUIC     1   1    0.0146 0.90407
## Residuals    4105  62
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
etaSquared(aov)
```

```
##                  eta.sq eta.sq.part
## DX_APD          0.0648794369 0.0677511026
## RTQUIC          0.0513827627 0.0544241009
## DX_APD:RTQUIC 0.0002108903 0.0002361736
```

p >.017 so no need to bonferroni correct as it is ns

6. AD+ & COHORT CHARACTERISTICS

Comparisons shown in: Results > ASyn-SAA+ & Demographics

```
df %>% group_by(DX_APD) %>% count(AD)
```

```
## # A tibble: 4 × 3
## # Groups:   DX_APD [2]
##   DX_APD AD      n
##   <chr>   <chr>   <int>
## 1 CBS     AD Negative    27
## 2 CBS     AD Positive     12
## 3 PSP     AD Negative    25
## 4 PSP     AD Positive      3
```

6.1. AD+ WHOLE COHORT CHARACTERISTICS: CATEGORICAL VARIABLES

Not in the manuscript

SEX STATISTICS: DISTRIBUTION

```
table(df$AD, df$Sex)
```

```
##
##          F   M
## AD Negative 22 30
## AD Positive 10  5
```

SEX STATISTICS: CHI SQUARE

```
chisq.test(table(df$AD, df$Sex), correct=F)
```

```
##
## Pearson's Chi-squared test
##
## data: table(df$AD, df$Sex)
## X-squared = 2.7687, df = 1, p-value = 0.09613
```

PPA STATISTICS: DISTRIBUTION

```
table(df$AD, df$anyPPA)
```

```
##  
##           No Yes  
##   AD Negative 44  8  
##   AD Positive  8  7
```

PPA STATISTICS: CHI SQUARE

```
fisher.test(table(df$AD, df$anyPPA)) #expected outcome < 5. Significant
```

```
##  
## Fisher's Exact Test for Count Data  
##  
## data: table(df$AD, df$anyPPA)  
## p-value = 0.02977  
## alternative hypothesis: true odds ratio is not equal to 1  
## 95 percent confidence interval:  
## 1.112209 20.203811  
## sample estimates:  
## odds ratio  
## 4.671131
```

PARK ONSET STATISTICS: DISTRIBUTION

```
table(df$AD, df$Parkinsonian_onset)
```

```
##  
##           No Yes  
##   AD Negative 26 26  
##   AD Positive  9  6
```

PARK ONSET STATISTICS: CHI SQUARE

```
chisq.test(table(df$AD, df$Parkinsonian_onset), correct=F)
```

```
##  
## Pearson's Chi-squared test  
##  
## data: table(df$AD, df$Parkinsonian_onset)  
## X-squared = 0.46661, df = 1, p-value = 0.4946
```

PARK ONSET STATISTICS: DISTRIBUTION

```
table(df$AD, df$AP0Ee4)
```

```
## Negative Positive
## AD Negative      36     11
## AD Positive      11      3
```

PARK ONSET STATISTICS: CHI SQUARE

```
fisher.test(table(df$AD, df$AP0Ee4)) #expected outcome < 5
```

```
## Fisher's Exact Test for Count Data
##
## data: table(df$AD, df$AP0Ee4)
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.1360715 4.3027210
## sample estimates:
## odds ratio
## 0.8941862
```

6.2. AD+ WHOLE COHORT CHARACTERISTICS: NUMERICAL VARIABLES

Not in the manuscript

AGE STATISTICS: DISTRIBUTION

```
shapiro.test(ADposdf$Age) #normal
```

```
## Shapiro-Wilk normality test
##
## data: ADposdf$Age
## W = 0.97209, p-value = 0.8877
```

```
shapiro.test(ADnegdf$Age) #normal
```

```
## Shapiro-Wilk normality test
##
## data: ADnegdf$Age
## W = 0.97163, p-value = 0.2474
```

```
leveneTest(Age ~ AD, data = df) #homoscedasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value Pr(>F)
## group    1  0.6562 0.4209
##          65
```

AGE STATISTICS: SUMMARY

```
df %>% group_by(AD) %>% summarize(count=n(), format(round(mean(Age, na.rm=T),2),2), sd=sd(Age, na.rm=T))
```

```
## # A tibble: 2 × 4
##   AD           count `format(round(mean(Age, na.rm = T), 2), 2)`     sd
##   <chr>        <int> <chr>                                         <dbl>
## 1 AD Negative    52  69.13                                         9.54
## 2 AD Positive     15  67.09                                         7.86
```

AGE STATISTICS: TTEST

```
t.test(df$Onset_age ~ df$AD, var.equal=TRUE)
```

```
##
## Two Sample t-test
##
## data: df$Onset_age by df$AD
## t = 0.44848, df = 65, p-value = 0.6553
## alternative hypothesis: true difference in means between group AD Negative and group
## AD Positive is not equal to 0
## 95 percent confidence interval:
## -3.935658 6.215146
## sample estimates:
## mean in group AD Negative mean in group AD Positive
##                      63.67308                      62.53333
```

ONSET STATISTICS: DISTRIBUTION

```
shapiro.test(ADposdf$Onset_age) #normal
```

```
##
## Shapiro-Wilk normality test
##
## data: ADposdf$Onset_age
## W = 0.95877, p-value = 0.6711
```

```
shapiro.test(ADnegdf$Onset_age) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: ADnegdf$Onset_age  
## W = 0.95838, p-value = 0.06657
```

```
leveneTest(Onset_age ~ AD, data = df) #homoscedasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to  
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)  
##      Df F value Pr(>F)  
## group  1  1.6381 0.2051  
##       65
```

ONSET STATISTICS: SUMMARY

```
df %>% group_by(AD) %>% summarize(count=n(), format(round(mean(Onset_age, na.rm=T),2),  
2), sd=sd(Onset_age, na.rm=T))
```

```
## # A tibble: 2 × 4  
##   AD           count `format(round(mean(Onset_age, na.rm = T), 2), 2)`     sd  
##   <chr>        <int> <chr>                                         <dbl>  
## 1 AD Negative    52  63.67                                         9.10  
## 2 AD Positive     15  62.53                                         6.88
```

ONSET STATISTICS: TTEST

```
t.test(df$Onset_age ~ df$AD, var.equal=TRUE)
```

```
##  
## Two Sample t-test  
##  
## data: df$Onset_age by df$AD  
## t = 0.44848, df = 65, p-value = 0.6553  
## alternative hypothesis: true difference in means between group AD Negative and group  
## AD Positive is not equal to 0  
## 95 percent confidence interval:  
## -3.935658  6.215146  
## sample estimates:  
## mean in group AD Negative mean in group AD Positive  
##                      63.67308                      62.53333
```

PARK ONSET STATISTICS: DISTRIBUTION

```
shapiro.test(ADposdf$Park_onset) #borderline
```

```
## 
## Shapiro-Wilk normality test
## 
## data: ADposdf$Park_onset
## W = 0.95173, p-value = 0.5879
```

```
shapiro.test(ADnegdf$Park_onset) #borderline
```

```
## 
## Shapiro-Wilk normality test
## 
## data: ADnegdf$Park_onset
## W = 0.95174, p-value = 0.03451
```

```
leveneTest(Park_onset ~ AD, data = df) #homoscedasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group  1  1.1953 0.2784
##       64
```

PARK ONSET STATISTICS: SUMMARY

```
df %>% group_by(AD) %>% summarize(count=n(), format(round(mean(Park_onset, na.rm=T),2),
2), sd=sd(Park_onset, na.rm=T))
```

```
## # A tibble: 2 × 4
##   AD           count `format(round(mean(Park_onset, na.rm = T), 2), 2)`     sd
##   <chr>        <int> <chr>                                         <dbl>
## 1 AD Negative    52  65.02                                         8.87
## 2 AD Positive     15  64.21                                         6.65
```

PARK ONSET STATISTICS: WILCOX

```
wilcox.test(df$Park_onset ~ df$AD, paired=F)
```

```
## 
## Wilcoxon rank sum test with continuity correction
## 
## data: df$Park_onset by df$AD
## W = 401, p-value = 0.5666
## alternative hypothesis: true location shift is not equal to 0
```

DURATION STATISTICS: DISTRIBUTION

```
shapiro.test(ADposdf$LP2_Disease_Duration) #not normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: ADposdf$LP2_Disease_Duration  
## W = 0.8728, p-value = 0.0371
```

```
shapiro.test(ADnegdf$LP2_Disease_Duration) #not normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: ADnegdf$LP2_Disease_Duration  
## W = 0.75445, p-value = 5.928e-08
```

```
leveneTest(LP2_Disease_Duration ~ AD, data = df) #homoscedasticity but borderline
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to  
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)  
##       Df F value Pr(>F)  
## group  1  0.0223 0.8818  
##       65
```

DURATION STATISTICS: SUMMARY

```
df %>% group_by(AD) %>% summarize(count=n(), format(round(median(LP2_Disease_Duration, na.rm=T),2),2), IQR=IQR(LP2_Disease_Duration, na.rm=T), min=min(LP2_Disease_Duration, na.rm=T), max=max(LP2_Disease_Duration, na.rm=T))
```

```
## # A tibble: 2 × 6  
##   AD      count format(round(median(LP2_Disease_Duration...¹    IQR    min    max  
##   <chr>     <int> <chr>                                <dbl> <dbl> <dbl>  
## 1 AD Negative     52 4.35                               2.63  1.65  17.3  
## 2 AD Positive      15 3.12                             3.03  1.03  10.7  
## # i abbreviated name:  
## #   ¹`format(round(median(LP2_Disease_Duration, na.rm = T), 2), 2)`
```

DURATION STATISTICS: WILCOX

```
wilcox.test(df$LP2_Disease_Duration ~ df$AD, paired=F)
```

```

## 
## Wilcoxon rank sum test with continuity correction
## 
## data: df$LP2_Disease_Duration by df$AD
## W = 469.5, p-value = 0.2347
## alternative hypothesis: true location shift is not equal to 0

```

6.3. AD+ CBS COHORT CHARACTERISTICS: CATEGORICAL VARIABLES

Comparisons shown in: eTable 2

SEX STATISTICS: DISTRIBUTION

```
table(CBSdf$AD, CBSdf$Sex)
```

```

## 
##          F   M
## AD Negative 11 16
## AD Positive  8  4

```

SEX STATISTICS: CHI SQUARE

```
chisq.test(table(CBSdf$AD, CBSdf$Sex), correct=F)
```

```

## 
## Pearson's Chi-squared test
## 
## data: table(CBSdf$AD, CBSdf$Sex)
## X-squared = 2.2351, df = 1, p-value = 0.1349

```

APOE STATISTICS: DISTRIBUTION

```
table(CBSdf$AP0Ee4, CBSdf$AD)
```

```

## 
##          AD Negative AD Positive
## Negative            21             9
## Positive            5              2

```

APOE STATISTICS: FISHER'S TEST

```
fisher.test(table(CBSdf$AP0Ee4, CBSdf$AD)) # Expected count is <5 for one cell
```

```

## 
## Fisher's Exact Test for Count Data
## 
## data: table(CBSdf$AP0Ee4, CBSdf$AD)
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.07563492 7.17281592
## sample estimates:
## odds ratio
## 0.935053

```

LANG ONSET STATISTICS: DISTRIBUTION

```
table(CBSdf$Language_onset, CBSdf$AD)
```

```

## 
##      AD Negative AD Positive
## No        23          5
## Yes       4           7

```

LANG ONSET STATISTICS: FISHER'S TEST

```
fisher.test(table(CBSdf$Language_onset, CBSdf$AD)) # Expected count is <5 for one cell
```

```

## 
## Fisher's Exact Test for Count Data
## 
## data: table(CBSdf$Language_onset, CBSdf$AD)
## p-value = 0.01698
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 1.332041 51.349652
## sample estimates:
## odds ratio
## 7.513935

```

PPA STATISTICS: DISTRIBUTION

```
table(CBSdf$anyPPA, CBSdf$AD)
```

```

## 
##      AD Negative AD Positive
## No        19          5
## Yes       8           7

```

PPA STATISTICS: FISHER'S TEST

```
fisher.test(table(CBSdf$anyPPA, CBSdf$AD)) # Expected count is <5 for one cell
```

```
## 
## Fisher's Exact Test for Count Data
##
## data: table(CBSdf$anyPPA, CBSdf$AD)
## p-value = 0.1532
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.6534606 17.3822576
## sample estimates:
## odds ratio
## 3.215136
```

6.4. AD+ CBS COHORT CHARACTERISTICS: NUMERICAL VARIABLES

6.4.1. AGE, ONSET, PARK ONSET

Comparisons shown in: eTable 2

BONFERRONI CALCULATION FOR AGE/ONSET AGE/PARKINSONISM AGE COMPARISONS ATTRIBUTABLE TO RTQUIC Looking at three similar tests: Difference in mean age at LP, onset, and Parkinsonism onset, in RT-QUIC+ vs RT-QUIC-

```
0.05/3
```

```
## [1] 0.01666667
```

AGE STATISTICS: DISTRIBUTION

```
shapiro.test(CBSdf[CBSdf$AD=="AD Positive", ]$Age) #normal
```

```
## 
## Shapiro-Wilk normality test
##
## data: CBSdf[CBSdf$AD == "AD Positive", ]$Age
## W = 0.98409, p-value = 0.9951
```

```
shapiro.test(CBSdf[CBSdf$AD=="AD Negative", ]$Age) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: CBSdf[CBSdf$AD == "AD Negative", ]$Age  
## W = 0.9459, p-value = 0.1703
```

`leveneTest(Age ~ AD, CBSdf)` #Homoscedasticity. Specify saturated model, ie includes the interaction term even if aovmodel does not

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to  
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)  
##          Df F value Pr(>F)  
## group    1  2.5781 0.1169  
##          37
```

AGE STATISTICS: SUMMARY

```
CBSdf %>% summarize(count=n(), mean=format(round(mean(Age, na.rm=T),3),3), sd=format(rou  
nd(sd(Age, na.rm=T),3),3))
```

```
##   count   mean     sd  
## 1    39 65.918 8.584
```

```
CBSdf %>% group_by(AD) %>% summarize(count=n(), mean=format(round(mean(Age, na.rm=T),3),  
3), sd=format(round(sd(Age, na.rm=T),3),3))
```

```
## # A tibble: 2 × 4  
##   AD           count   mean     sd  
##   <chr>        <int> <chr>   <chr>  
## 1 AD Negative    27 66.456 9.328  
## 2 AD Positive     12 64.707 6.825
```

AGE STATISTICS: TTEST

```
t.test(CBSdf$Age ~ CBSdf$AD, var.equal=TRUE)
```

```
## 
## Two Sample t-test
##
## data: CBSdf$Age by CBSdf$AD
## t = 0.58228, df = 37, p-value = 0.5639
## alternative hypothesis: true difference in means between group AD Negative and group
## AD Positive is not equal to 0
## 95 percent confidence interval:
## -4.338025 7.836807
## sample estimates:
## mean in group AD Negative mean in group AD Positive
## 66.45601 64.70662
```

ONSET STATISTICS: DISTRIBUTION

```
shapiro.test(CBSdf[CBSdf$AD=="AD Positive", ]$Onset_age) #normal
```

```
## 
## Shapiro-Wilk normality test
##
## data: CBSdf[CBSdf$AD == "AD Positive", ]$Onset_age
## W = 0.95193, p-value = 0.6654
```

```
shapiro.test(CBSdf[CBSdf$AD=="AD Negative", ]$Onset_age) #normal
```

```
## 
## Shapiro-Wilk normality test
##
## data: CBSdf[CBSdf$AD == "AD Negative", ]$Onset_age
## W = 0.94304, p-value = 0.1448
```

leveneTest(Onset_age ~ AD, CBSdf) #Homoscedasticity. Specify saturated model, ie includes the interaction term even if aovmodel does not

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value Pr(>F)
## group    1   1.516  0.226
##          37
```

ONSET STATISTICS: SUMMARY

```
CBSdf %>% summarize(count=n(), mean=format(round(mean(Onset_age, na.rm=T),3),3), sd=format(round(sd(Onset_age, na.rm=T),3),3))
```

```
##   count   mean    sd
## 1    39 61.359 8.327
```

```
CBSdf %>% group_by(AD) %>% summarize(count=n(), mean=format(round(mean(Onset_age, na.rm=T),3),3), sd=format(round(sd(Onset_age, na.rm=T),3),3))
```

```
## # A tibble: 2 × 4
##   AD           count   mean    sd
##   <chr>        <int> <chr> <chr>
## 1 AD Negative    27 61.63 9.17
## 2 AD Positive     12 60.75 6.341
```

ONSET STATISTICS: TTEST

```
t.test(CBSdf$Onset_age ~ CBSdf$AD, var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data: CBSdf$Onset_age by CBSdf$AD
## t = 0.3008, df = 37, p-value = 0.7652
## alternative hypothesis: true difference in means between group AD Negative and group
## AD Positive is not equal to 0
## 95 percent confidence interval:
## -5.045494 6.804754
## sample estimates:
## mean in group AD Negative mean in group AD Positive
##                      61.62963                  60.75000
```

PARK ONSET STATISTICS: DISTRIBUTION

```
shapiro.test(CBSdf[CBSdf$AD=="AD Positive", ]$Park_onset) #normal
```

```
##
## Shapiro-Wilk normality test
##
## data: CBSdf[CBSdf$AD == "AD Positive", ]$Park_onset
## W = 0.94659, p-value = 0.6008
```

```
shapiro.test(CBSdf[CBSdf$AD=="AD Negative", ]$Park_onset) #normal
```

```
##
## Shapiro-Wilk normality test
##
## data: CBSdf[CBSdf$AD == "AD Negative", ]$Park_onset
## W = 0.93793, p-value = 0.1083
```

```
leveneTest(Park_onset ~ AD, CBSdf) #Homoscedasticity. Specify saturated model, ie includes the interaction term even if aovmodel does not
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value Pr(>F)
## group    1  1.8368 0.1838
##          36
```

PARK ONSET STATISTICS: SUMMARY

```
CBSdf %>% summarize(count=n(), mean=format(round(mean(Park_onset, na.rm=T),3),3), sd=format(round(sd(Park_onset, na.rm=T),3),3))
```

```
##   count   mean     sd
## 1    39 63.158 8.228
```

```
CBSdf %>% group_by(AD) %>% summarize(count=n(), mean=format(round(mean(Park_onset, na.rm=T),3),3), sd=format(round(sd(Park_onset, na.rm=T),3),3))
```

```
## # A tibble: 2 × 4
##   AD           count   mean     sd
##   <chr>        <int> <chr>   <chr>
## 1 AD Negative    27 63.37  8.945
## 2 AD Positive     12 62.636 6.485
```

PARK ONSET STATISTICS: TTEST

```
t.test(CBSdf$Park_onset ~ CBSdf$AD, var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data: CBSdf$Park_onset by CBSdf$AD
## t = 0.2462, df = 36, p-value = 0.8069
## alternative hypothesis: true difference in means between group AD Negative and group
## AD Positive is not equal to 0
## 95 percent confidence interval:
## -5.312346  6.780360
## sample estimates:
## mean in group AD Negative mean in group AD Positive
##                      63.37037                  62.63636
```

6.4.2. NFL

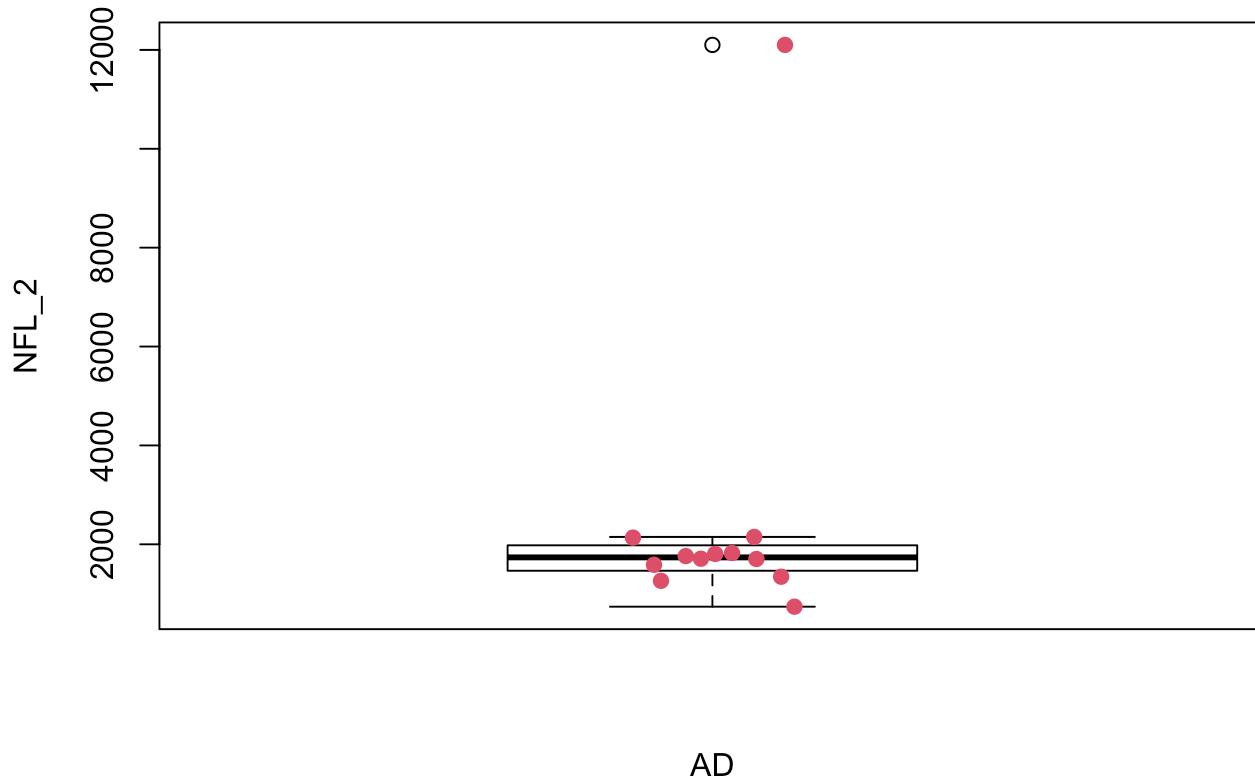
Comparisons shown in: eTable 2

NFL STATISTICS: DISTRIBUTION

```
boxplot(NFL_2 ~ AD, data= CBSdf[CBSdf$AD=="AD Positive", ], col = "white")$out #identify outliers in each diagnosis. First, look at CBS: there is one so attribute its value to vector.
```

```
## [1] 12101.2
```

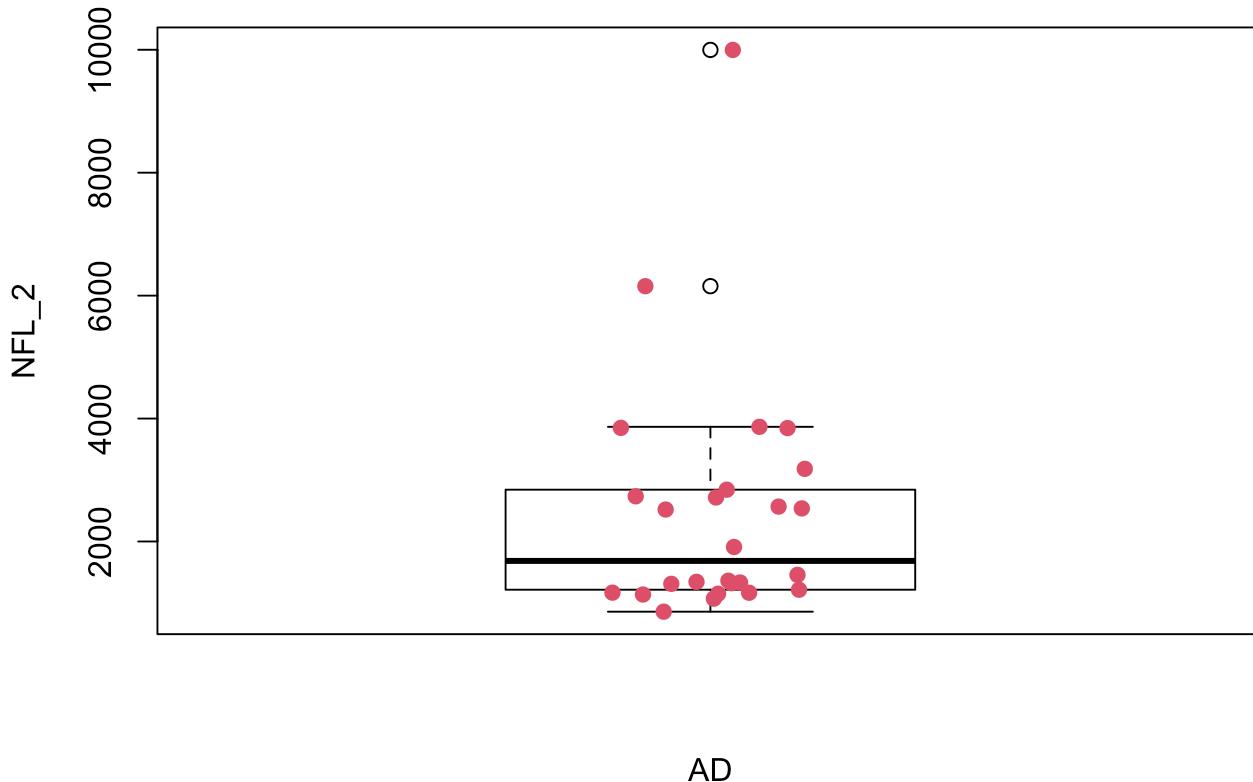
```
stripchart(NFL_2 ~ AD, data = CBSdf[CBSdf$AD=="AD Positive", ], method = "jitter", pch = 19, col = 2:4, vertical = TRUE, add = TRUE)
```



```
boxplot(NFL_2 ~ AD, data= CBSdf[CBSdf$AD=="AD Negative", ], col = "white")$out #Now identify outliers in PSP: since there is none, no need to attribute to a vector.
```

```
## [1] 6153.82 9997.64
```

```
stripchart(NFL_2 ~ AD, data = CBSdf[CBSdf$AD=="AD Negative", ], method = "jitter", pch = 19, col = 2:4, vertical = TRUE, add = TRUE)
```



```
# Remove outliers over full dataset, but at a tolerant threshold (Q3+3*IQR instead of 1.5 IQR. Reference for this is: https://www.nature.com/articles/s41598-020-66090-x
# This is because in FTLD population, the data can be very right-skewed.
# Although technically median would still be ok to show in the Table even with outliers,
# but because it may be of interest to share these extreme values, I prefer having them in the Notes.
thresholdAD <- min(max(CBSdf[CBSdf$AD=="AD Positive", ]$NFL_2, na.rm=T), as.numeric(quantile(CBSdf[CBSdf$AD=="AD Positive", ]$NFL_2, 0.75, na.rm=T)) + (IQR(na.rm=T, (CBSdf[CBSdf$AD=="AD Positive", ]$NFL_2)*3))) #reports the value Q3+ IQR*3 (3 is very tolerant threshold)
thresholdnonAD <- min(max(CBSdf[CBSdf$AD=="AD Negative", ]$NFL_2, na.rm=T), as.numeric(quantile(CBSdf[CBSdf$AD=="AD Negative", ]$NFL_2, 0.75, na.rm=T)) + (IQR(na.rm=T, (CBSdf[CBSdf$AD=="AD Negative", ]$NFL_2)*3))) #reports the value Q3+ IQR*3 (3 is very tolerant threshold)
cat("Outliers are values above ", thresholdAD, " in CBS-AD+ subset. \n")
```

```
## Outliers are values above 3030.122 in CBS-AD+ subset.
```

```
cat("Outliers are values above ", thresholdnonAD, " in CBS-AD- subset. \n")
```

```
## Outliers are values above 7545.9 in CBS-AD- subset.
```

```
CBSdfnfl<- CBSdf[CBSdf$AD=="AD Negative" | CBSdf$NFL_2 <= thresholdAD, ] %>% remove_empty("rows") %>% data.frame() #Removes all subjects who are CBS and either have no NFL value or one over the threshold
CBSdfnfl<- CBSdfnfl[CBSdfnfl$AD=="AD Positive" | CBSdfnfl$NFL_2 <= thresholdnonAD, ] %>% remove_empty("rows") %>% data.frame() #Removes all subjects who are PSP and either have no NFL value or one over the threshold

removed <- setdiff(CBSdf, CBSdfnfl)
cat("Following values were removed for the descriptive stats on NfL: ", removed$NFL_2,
"\n")
```

```
## Following values were removed for the descriptive stats on NfL: 12101.2 NA 9997.64
```

```
shapiro.test(CBSdfnfl[CBSdfnfl$AD=="AD Positive", ]$logNFL) #non-normal
```

```
##
## Shapiro-Wilk normality test
##
## data: CBSdfnfl[CBSdfnfl$AD == "AD Positive", ]$logNFL
## W = 0.82416, p-value = 0.01956
```

```
shapiro.test(CBSdfnfl[CBSdfnfl$AD=="AD Negative", ]$logNFL) #non-normal
```

```
##
## Shapiro-Wilk normality test
##
## data: CBSdfnfl[CBSdfnfl$AD == "AD Negative", ]$logNFL
## W = 0.91206, p-value = 0.03389
```

```
leveneTest(logNFL ~ AD, data = CBSdfnfl) #heterodasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group  1 4.8885 0.03386 *
##      34
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

NFL STATISTICS: SUMMARY

```
CBSdfnfl %>% summarize(count=n(), format(round(median(NFL_2, na.rm=T),2),2), IQR=IQR(NFL_2, na.rm=T), min=min(NFL_2, na.rm=T), max=max(NFL_2, na.rm=T))
```

```
##   count format(round(median(NFL_2, na.rm = T), 2), 2)      IQR min     max
## 1    36                                         1705.76 1247.092 738 6153.82
```

```
CBSdfnfl %>% group_by(AD) %>% summarize(count=n(), format(round(median(NFL_2, na.rm=T), 2),2), IQR=IQR(NFL_2, na.rm=T), min=min(NFL_2, na.rm=T), max=max(NFL_2, na.rm=T))
```

```
## # A tibble: 2 × 6
##   AD           count format(round(median(NFL_2, na.rm = T), 2)...¹ IQR   min   max
##   <chr>        <int> <chr>                               <dbl> <dbl> <dbl>
## 1 AD Negative    25  1456.97                           1520.  858.  6154.
## 2 AD Positive     11  1708.68                           350.   738.  2149.
## # i abbreviated name: `¹`format(round(median(NFL_2, na.rm = T), 2), 2)`
```

```
IQR(CBSdfnfl[CBSdfnfl$AD=="AD Positive", ]$NFL_2, na.rm=T)
```

```
## [1] 349.545
```

```
IQR(CBSdfnfl[CBSdfnfl$AD=="AD Negative", ]$NFL_2, na.rm=T)
```

```
## [1] 1520.12
```

NFL STATISTICS: WEIGHTED LEAST-SQUARE REGRESSION

```
t.test(CBSdfnfl$logNFL ~ CBSdfnfl$AD, var.equal=FALSE) #Welch t-test, to deal with the heterodasticity
```

```
##
## Welch Two Sample t-test
##
## data: CBSdfnfl$logNFL by CBSdfnfl$AD
## t = 1.3167, df = 31.416, p-value = 0.1975
## alternative hypothesis: true difference in means between group AD Negative and group
## AD Positive is not equal to 0
## 95 percent confidence interval:
## -0.1003594  0.4665510
## sample estimates:
## mean in group AD Negative mean in group AD Positive
##                      7.548587                      7.365491
```

```
gls1 <- gls(logNFL ~ Age*AD, CBSdfnfl, weights=varPower()) #to deal with the heterodasticity
gls2 <- gls(logNFL ~ Age + AD, CBSdfnfl, weights=varPower())
AIC(gls1, gls2) #gls2 better than gls1
```

```
## Warning in AIC.default(gls1, gls2): models are not all fitted to the same
## number of observations
```

```
##      df      AIC
## gls1  6 74.75601
## gls2  5 66.77470
```

```
summary(gls2)
```

```
## Generalized least squares fit by REML
## Model: logNFL ~ Age + AD
## Data: CBSdfnfl
##      AIC      BIC    logLik
## 66.7747 74.25724 -28.38735
##
## Variance function:
## Structure: Power of variance covariate
## Formula: ~fitted(.)
## Parameter estimates:
## power
## 10.5513
##
## Coefficients:
##              Value Std.Error t-value p-value
## (Intercept) 7.032552 0.5817078 12.089492 0.0000
## Age         0.007731 0.0089026  0.868421 0.3914
## ADAD Positive -0.158978 0.1538188 -1.033543 0.3089
##
## Correlation:
##          (Intr) Age
## Age     -0.985
## ADAD Positive -0.176 0.067
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.9632510 -0.7229886  0.2280254  0.4943355  2.7695795
##
## Residual standard error: 2.74902e-10
## Degrees of freedom: 36 total; 33 residual
```

6.4.3. MOCA Z-SCORES

Comparisons shown in: eTable 2

Of note, while the MoCA scores are z-scored to control for the effect of age, that does not mean there is no effect of age when analyzing them. MoCA z-scores are worse in younger subjects, since the latter are compared to a population with no cognitive deficit. So even small changes in MoCA score in young subjects translates to very low z-scores.

Here it doesn't matter as we are just presenting the values in the table to summarize the dataset.

MOCA Z-SCORE STATISTICS: DISTRIBUTION

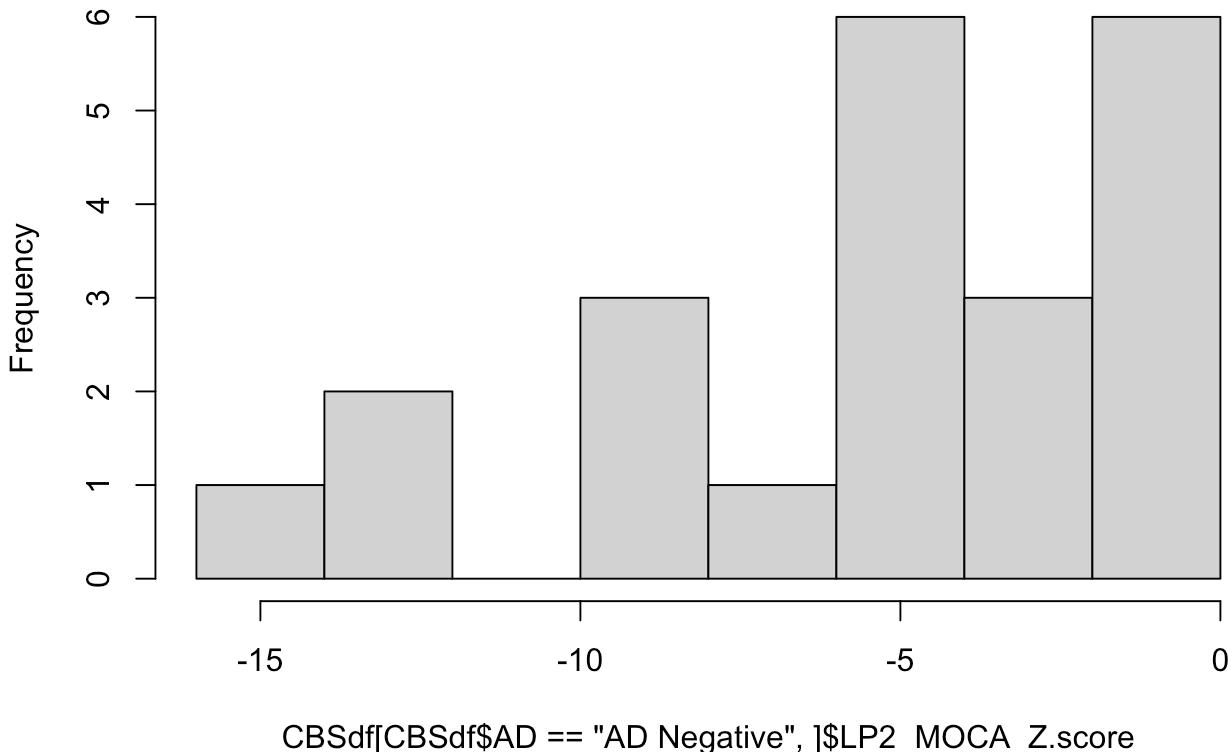
```
shapiro.test(CBSdf[CBSdf$AD=="AD Positive", ]$LP2_MOCA_Z.score) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: CBSdf[CBSdf$AD == "AD Positive", ]$LP2_MOCA_Z.score  
## W = 0.97774, p-value = 0.973
```

```
shapiro.test(CBSdf[CBSdf$AD=="AD Negative", ]$LP2_MOCA_Z.score) #borderline
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: CBSdf[CBSdf$AD == "AD Negative", ]$LP2_MOCA_Z.score  
## W = 0.92192, p-value = 0.08331
```

```
hist(CBSdf[CBSdf$AD=="AD Negative", ]$LP2_MOCA_Z.score) #not normal
```

Histogram of CBSdf[CBSdf\$AD == "AD Negative",]\$LP2_MOCA_Z.score

```
leveneTest(LP2_MOCA_Z.score ~ AD, CBSdf) #Homoscedasticity. Specify saturated model, ie  
includes the interaction term even if aovmodel does not
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value Pr(>F)
## group    1  0.0232   0.88
##          32
```

MOCA Z-SCORE STATISTICS: SUMMARY

```
CBSdf %>% summarize(count=n(), format(round(median(LP2_MOCA_Z.score, na.rm=T),2),2), IQR=IQR(LP2_MOCA_Z.score, na.rm=T), min=min(LP2_MOCA_Z.score, na.rm=T), max=max(LP2_MOCA_Z.score, na.rm=T))
```

```
##   count format(round(median(LP2_MOCA_Z.score, na.rm = T), 2), 2)      IQR      min
## 1     39                           -6.32  7.9875 -18.35
##   max
## 1 -0.16
```

```
CBSdf %>% group_by(AD) %>% summarize(count=n(), format(round(median(LP2_MOCA_Z.score, na.rm=T),2),2), IQR=IQR(LP2_MOCA_Z.score, na.rm=T), min=min(LP2_MOCA_Z.score, na.rm=T), max=max(LP2_MOCA_Z.score, na.rm=T))
```

```
## # A tibble: 2 × 6
##   AD           count format(round(median(LP2_MOCA_Z.score, na.rm = T), 2), 2)^  IQR      min      max
##   <chr>        <int> <chr>                                         <dbl> <dbl> <dbl>
## 1 AD Negative    27 -5.03                                         6.33 -15.7 -0.16
## 2 AD Positive     12 -10.73                                        5.11 -18.4 -1.84
## # i abbreviated name:
## # ^`format(round(median(LP2_MOCA_Z.score, na.rm = T), 2), 2)`
```

MOCA Z-SCORE STATISTICS: WILCOX

```
wilcox.test(CBSdf$LP2_MOCA_Z.score ~ CBSdf$AD)
```

```
##
## Wilcoxon rank sum exact test
##
## data: CBSdf$LP2_MOCA_Z.score by CBSdf$AD
## W = 200, p-value = 0.0134
## alternative hypothesis: true location shift is not equal to 0
```

7. ASYN-SAA+ & AD

7.1. ASYN-SAA+ & AD: CATEGORICAL VARIABLES

Comparisons shown in: eTable 1, Results > ASyn-SAA+ & Demographics

```
sum(df$AD=="AD Positive" & df$RTQUIC=="aSyn-SAA positive" & df$Sex=="M")
```

```
## [1] 2
```

```
sum(df$AD=="AD Positive" & df$RTQUIC=="aSyn-SAA positive" & df$Sex=="F")
```

```
## [1] 5
```

```
sum(df$AD=="AD Negative" & df$RTQUIC=="aSyn-SAA positive" & df$Sex=="M")
```

```
## [1] 8
```

```
sum(df$AD=="AD Negative" & df$RTQUIC=="aSyn-SAA positive" & df$Sex=="F")
```

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

7.1.1. FIGURE 1A: ONSET * AD QUADRANTS

```
## Below are the values for the creation of Fig 1.A:
```

```
## Lower left quadrant: AD+ and young-onset
```

```
## CBS subjects who are RTQUIC+ and young-onset and AD+: 5
```

```
## PSP subjects who are RTQUIC+ and young-onset and AD+: 0
```

```
## CBS subjects who are RTQUIC- and young-onset and AD+: 4
```

```
## PSP subjects who are RTQUIC+ and young-onset and AD+: 0
```

```
## Upper left quadrant: AD- and young-onset
```

```
## CBS subjects who are RTQUIC+ and young-onset and AD-: 2
```

```
## PSP subjects who are RTQUIC+ and young-onset and AD-: 1
```

```
## CBS subjects who are RTQUIC- and young-onset and AD-: 14
```

```
## PSP subjects who are RTQUIC+ and young-onset and AD-: 9
```

```
## Lower right quadrant:
```

```
## CBS subjects who are RTQUIC+ and late-onset and AD+: 2
```

```
## PSP subjects who are RTQUIC+ and late-onset and AD+: 0
```

```
## CBS subjects who are RTQUIC- and late-onset and AD+: 1
```

```
## PSP subjects who are RTQUIC- and late-onset and AD+: 3
```

```
## Upper right quadrant:
```

```
## CBS subjects who are RTQUIC+ and late-onset and AD-: 5
```

```
## PSP subjects who are RTQUIC+ and late-onset and AD-: 7
```

```
## CBS subjects who are RTQUIC- and late-onset and AD-: 6
```

```
## PSP subjects who are RTQUIC- and late-onset and AD-: 8
```

7.1.2. FREQUENCY DATA

Comparisons shown in: Fig 1A, Results > ASyn-SAA+ & AD+

BONFERRONI CALCULATION FOR FISHER TESTS USED FOR ASYN/AD RELATIONSHIP IN OVERALL DATASET AND YO DATASET AND LO DATASET Looking at three similar tests: Difference in mean age at LP, onset, and Parkinsonism onset, in RT-QUIC+ vs RT-QUIC-

```
0.05/3
```

```
## [1] 0.01666667
```

WHOLE COHORT STATISTICS: DISTRIBUTION

```
table(df$RTQUIC, df$AD)
```

```
##  
## AD Negative AD Positive  
## aSyn-SAA negative 37 8  
## aSyn-SAA positive 15 7
```

```
sum(YODdf$AD=="AD Positive" & YODdf$RTQUIC=="aSyn-SAA positive")
```

```
## [1] 5
```

```
sum(YODdf$RTQUIC=="aSyn-SAA positive")
```

```
## [1] 8
```

```
sum(LODdf$AD=="AD Positive" & LODdf$RTQUIC=="aSyn-SAA positive")
```

```
## [1] 2
```

```
sum(LODdf$RTQUIC=="aSyn-SAA positive")
```

```
## [1] 14
```

WHOLE COHORT STATISTICS: FISHER

```
fisher.test(table(df$AD, df$RTQUIC)) # Expected count is <5 for one cell. Gives OR
```

```
##  
## Fisher's Exact Test for Count Data  
##  
## data: table(df$AD, df$RTQUIC)  
## p-value = 0.223  
## alternative hypothesis: true odds ratio is not equal to 1  
## 95 percent confidence interval:  
## 0.5516084 8.1666232  
## sample estimates:  
## odds ratio  
## 2.131801
```

```
cramerV(table(df$AD, df$RTQUIC))
```

```
## Cramer V  
## 0.1582
```

```
phi(table(df$AD, df$RTQUIC), digits=6)
```

```
##      phi
## 0.158179
```

NOTES ON CRAMER AND PHI: df = $(r - 1) * (c - 1)$ = $1*1 = 1$. At df=1, medium Cramer's V=.3 CRAMER: COHEN 1988 REFERS TO AS CRAMER'S PHI. See p. 223

<https://www.utstat.toronto.edu/~brunner/oldclass/378f16/readings/CohenPower.pdf>

(<https://www.utstat.toronto.edu/~brunner/oldclass/378f16/readings/CohenPower.pdf>) PHI: IN 2X2 CASE, SAME AS W IN COHEN 1988 7.2. See p.223

<https://www.utstat.toronto.edu/~brunner/oldclass/378f16/readings/CohenPower.pdf>

(<https://www.utstat.toronto.edu/~brunner/oldclass/378f16/readings/CohenPower.pdf>)

YOUNG-ONSET STATISTICS+: DISTRIBUTION

```
table(Y0Ddf$AD, Y0Ddf$RTQUIC)
```

	aSyn-SAA negative	aSyn-SAA positive
AD Negative	23	3
AD Positive	4	5

YOUNG-ONSET STATISTICS+: FISHER

```
fisher.test(table(Y0Ddf$AD, Y0Ddf$RTQUIC)) # Expected count is <5 for one cell
```

```
##
## Fisher's Exact Test for Count Data
##
## data: table(Y0Ddf$AD, Y0Ddf$RTQUIC)
## p-value = 0.01512
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 1.187507 82.550067
## sample estimates:
## odds ratio
## 8.754574
```

```
cramerV(table(Y0Ddf$AD, Y0Ddf$RTQUIC))
```

```
## Cramer V
## 0.4581
```

```
phi(table(Y0Ddf$AD, Y0Ddf$RTQUIC), digits=6)
```

```
##      phi
## 0.458144
```

p=.015 < .017 so reported bonf-adjusted p-value: .015x3=.045

LATE-ONSET STATISTICS: DISTRIBUTION

```
table(L0Ddf$AD, L0Ddf$RTQUIC)
```

```
##          aSyn-SAA negative aSyn-SAA positive
##    AD Negative              14                  12
##    AD Positive                4                   2
```

LATE-ONSET STATISTICS: FISHER

```
fisher.test(table(L0Ddf$AD, L0Ddf$RTQUIC)) # Expected count is <5 for one cell
```

```
## 
## Fisher's Exact Test for Count Data
##
## data: table(L0Ddf$AD, L0Ddf$RTQUIC)
## p-value = 0.6722
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.0459508 5.0206949
## sample estimates:
## odds ratio
## 0.5929948
```

```
cramerV(table(L0Ddf$AD, L0Ddf$RTQUIC))
```

```
## Cramer V
## 0.1009
```

```
phi(table(L0Ddf$AD, L0Ddf$RTQUIC), digits=6)
```

```
##      phi
## -0.100871
```

7.1.3. FREQUENCY DATA WITHIN DX

Comparisons shown in Supp Text - ASYN-SAA+ & AD+ within DX

CBS: TRY TO SEE IF IN CBS, AD AND RTQUIC ARE ASSOCIATED. THEN, SEE IF EFFECT OF AGE.

CBS ASSOCIATION OF AD+/RTQUIC+ OVERALL

```
table(CBSdf$RTQUIC, CBSdf$AD)
```

```
##  
##  
##          AD Negative AD Positive  
##    aSyn-SAA negative      20       5  
##    aSyn-SAA positive       7        7
```

```
fisher.test(table(CBSdf$AD, CBSdf$RTQUIC)) # Expected count is <5 for one cell. Gives OR
```

```
##  
## Fisher's Exact Test for Count Data  
##  
## data: table(CBSdf$AD, CBSdf$RTQUIC)  
## p-value = 0.07488  
## alternative hypothesis: true odds ratio is not equal to 1  
## 95 percent confidence interval:  
## 0.7664236 21.3857232  
## sample estimates:  
## odds ratio  
## 3.843273
```

```
cramerV(table(CBSdf$AD, CBSdf$RTQUIC))
```

```
## Cramer V  
## 0.3118
```

```
phi(table(CBSdf$AD, CBSdf$RTQUIC), digits=6)
```

```
## phi  
## 0.311805
```

There is almost a significant relationship between AD+ and RTQUIC+ in CBS (p<.10)

CBS ASSOCIATION OF AD+/RTQUIC+ IN YOUNG-ONSET ONLY (25/39 subjects)

```
nrow(CBSdf[CBSdf$"Early_onset"=="Young-onset", ])
```

```
## [1] 25
```

```
table(CBSdf[CBSdf$"Early_onset"=="Young-onset", ]$RTQUIC, CBSdf[CBSdf$"Early_onset"=="Young-onset", ]$AD)
```

```
##  
##          AD Negative AD Positive  
##    aSyn-SAA negative      14       4  
##    aSyn-SAA positive       2        5
```

```
fisher.test(table(CBSdf[CBSdf$"Early_onset"=="Young-onset", ]$RTQUIC, CBSdf[CBSdf$"Early_onset"=="Young-onset", ]$AD)) # Expected count is <5 for one cell. Gives OR
```

```
## 
## Fisher's Exact Test for Count Data
##
## data: table(CBSdf[CBSdf$Early_onset == "Young-onset", ]$RTQUIC, CBSdf[CBSdf$Early_onset == "Young-onset", ]$AD)
## p-value = 0.05812
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.8865897 113.3329588
## sample estimates:
## odds ratio
## 7.85303
```

There is almost a significant relationship between AD+ and RTQUIC+ in young-onset CBS (p<.10)

```
nrow(CBSdf[CBSdf$"Early_onset"=="Late-onset", ])
```

```
## [1] 14
```

```
table(CBSdf[CBSdf$"Early_onset"=="Late-onset", ]$RTQUIC, CBSdf[CBSdf$"Early_onset"=="Late-onset", ]$AD)
```

	AD Negative	AD Positive
aSyn-SAA negative	6	1
aSyn-SAA positive	5	2

```
fisher.test(table(CBSdf[CBSdf$"Early_onset"=="Late-onset", ]$RTQUIC, CBSdf[CBSdf$"Early_onset"=="Late-onset", ]$AD)) # Expected count is <5 for one cell. Gives OR
```

```
## 
## Fisher's Exact Test for Count Data
##
## data: table(CBSdf[CBSdf$Early_onset == "Late-onset", ]$RTQUIC, CBSdf[CBSdf$Early_onset == "Late-onset", ]$AD)
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.09086265 164.58062230
## sample estimates:
## odds ratio
## 2.2554
```

From above we can conclude that there is evidence of a potential relationship between AD+ and RTQUIC+ in CBS, but not necessarily affected by age at onset. Hard to make conclusions regarding late-onset CBS as the sample size is low (14, including 5 who are RTQUIC+) but there is no evidence supporting that AD+ and RTQUIC+ are associated in late-onset CBS.

CBS ASSOCIATION OF RTQUIC+ WITH AGE

```
shapiro.test(CBSdf[CBSdf$"RTQUIC"=="aSyn-SAA positive", ]$Onset_age) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: CBSdf[CBSdf$RTQUIC == "aSyn-SAA positive", ]$Onset_age  
## W = 0.93077, p-value = 0.3128
```

```
shapiro.test(CBSdf[CBSdf$"RTQUIC"=="aSyn-SAA negative", ]$Onset_age)#normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: CBSdf[CBSdf$RTQUIC == "aSyn-SAA negative", ]$Onset_age  
## W = 0.94595, p-value = 0.2028
```

```
var.test(Onset_age ~ RTQUIC, data = CBSdf) #homoscedasticity
```

```
##  
## F test to compare two variances  
##  
## data: Onset_age by RTQUIC  
## F = 1.8716, num df = 24, denom df = 13, p-value = 0.2403  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.6469066 4.6826837  
## sample estimates:  
## ratio of variances  
## 1.871625
```

CBS RTQUIC*ONSET STATISTICS: ANCOVA

```
t.test(CBSdf$Onset_age ~ CBSdf$RTQUIC, var.equal=TRUE) #What is reported in eTable 1
```

```
## 
## Two Sample t-test
##
## data: CBSdf$Onset_age by CBSdf$RTQUIC
## t = -1.2509, df = 37, p-value = 0.2188
## alternative hypothesis: true difference in means between group aSyn-SAA negative and
## group aSyn-SAA positive is not equal to 0
## 95 percent confidence interval:
## -9.042210 2.139353
## sample estimates:
## mean in group aSyn-SAA negative mean in group aSyn-SAA positive
## 60.12000 63.57143
```

```
aov <- aov(Onset_age ~ RTQUIC + AD, data=CBSdf)
summary(aov)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
## RTQUIC	1	106.9	106.91	1.545	0.222
## AD	1	36.7	36.74	0.531	0.471
## Residuals	36	2491.3	69.20		

```
etaSquared(aov)
```

	eta.sq	eta.sq.part
## RTQUIC	0.05207605	0.05220340
## AD	0.01394372	0.01453336

CBS RTQUIC*AGE STATISTICS: EXTRA

```
t.test(CBSdf$Age ~ CBSdf$RTQUIC, var.equal=TRUE) #What is reported in eTable 1
```

```
## 
## Two Sample t-test
##
## data: CBSdf$Age by CBSdf$RTQUIC
## t = -1.1501, df = 37, p-value = 0.2575
## alternative hypothesis: true difference in means between group aSyn-SAA negative and
## group aSyn-SAA positive is not equal to 0
## 95 percent confidence interval:
## -9.062894 2.499684
## sample estimates:
## mean in group aSyn-SAA negative mean in group aSyn-SAA positive
## 64.73973 68.02133
```

```
aov <- aov(Age ~ AD + RTQUIC, CBSdf)
summary(aov)
```

```
##          Df Sum Sq Mean Sq F value Pr(>F)
## AD         1   25.4   25.42   0.348  0.559
## RTQUIC     1 144.0  144.03   1.971  0.169
## Residuals  36 2630.5   73.07
```

```
etaSquared(aov)
```

```
##           eta.sq eta.sq.part
## AD      0.02600470  0.02693438
## RTQUIC 0.05144085  0.05191224
```

Conclusion for CBS: There is no evidence of relationship between age and RTQUIC status within the CBS diagnosis. There IS evidence of a relationship between AD+ and RTQUIC+. These results could be attributable to the low sample size (especially fewer subjects who have late-onset disease.) The same is seen when looking directly at Age instead of Onset age.

PSP: TRY TO SEE IF IN PSP, THERE IS AN EFFECT OF AGE.

PSP ASSOCIATION OF RTQUIC+ WITH AGE

```
table(PSPdf$RTQUIC, PSPdf$Early_onset)
```

```
##                                Late-onset Young-onset
## aSyn-SAA negative             11            9
## aSyn-SAA positive              7            1
```

```
fisher.test(table(PSPdf$RTQUIC, PSPdf$Early_onset)) # Expected count is <5 for one cell.
Gives OR
```

```
##
## Fisher's Exact Test for Count Data
##
## data: table(PSPdf$RTQUIC, PSPdf$Early_onset)
## p-value = 0.1937
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.003499204 1.901090103
## sample estimates:
## odds ratio
## 0.1844736
```

```
shapiro.test(PSPdf[PSPdf$"RTQUIC"=="aSyn-SAA positive", ]$Onset_age) #normal
```

```
## 
## Shapiro-Wilk normality test
##
## data: PSPdf[PSPdf$RTQUIC == "aSyn-SAA positive", ]$Onset_age
## W = 0.90133, p-value = 0.297
```

```
shapiro.test(PSPdf[PSPdf$"RTQUIC"=="aSyn-SAA negative", ]$Onset_age)#normal
```

```
## 
## Shapiro-Wilk normality test
##
## data: PSPdf[PSPdf$RTQUIC == "aSyn-SAA negative", ]$Onset_age
## W = 0.96336, p-value = 0.613
```

```
var.test(Onset_age ~ RTQUIC, data = PSPdf) #homoscedasticity
```

```
## 
## F test to compare two variances
##
## data: Onset_age by RTQUIC
## F = 1.6274, num df = 19, denom df = 7, p-value = 0.5252
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.3630251 4.9650044
## sample estimates:
## ratio of variances
## 1.627407
```

PSP RTQUIC*ONSET STATISTICS: ANCOVA

```
t.test(PSPdf$Onset_age ~ PSPdf$RTQUIC, var.equal=TRUE) #What is reported in eTable 1
```

```
## 
## Two Sample t-test
##
## data: PSPdf$Onset_age by PSPdf$RTQUIC
## t = -1.4737, df = 26, p-value = 0.1526
## alternative hypothesis: true difference in means between group aSyn-SAA negative and
## group aSyn-SAA positive is not equal to 0
## 95 percent confidence interval:
## -12.033993 1.983993
## sample estimates:
## mean in group aSyn-SAA negative mean in group aSyn-SAA positive
## 64.850 69.875
```

```
aov <- aov(Onset_age ~ RTQUIC, data=PSPdf)
summary(aov)
```

```
##          Df Sum Sq Mean Sq F value Pr(>F)
## RTQUIC      1 144.3 144.29  2.172  0.153
## Residuals   26 1727.4   66.44
```

```
etaSquared(aov)
```

```
##          eta.sq eta.sq.part
## RTQUIC 0.07708938 0.07708938
```

```
t.test(PSPdf$Age ~ PSPdf$RTQUIC, var.equal=TRUE) #What is reported in eTable 1
```

```
##
##  Two Sample t-test
##
## data: PSPdf$Age by PSPdf$RTQUIC
## t = -0.54813, df = 26, p-value = 0.5883
## alternative hypothesis: true difference in means between group aSyn-SAA negative and
## group aSyn-SAA positive is not equal to 0
## 95 percent confidence interval:
## -9.601622 5.558882
## sample estimates:
## mean in group aSyn-SAA negative mean in group aSyn-SAA positive
##                               71.93205                           73.95342
```

```
aov <- aov(Age ~ AD + RTQUIC, CBSdf)
etaSquared(aov)
```

```
##          eta.sq eta.sq.part
## AD      0.02600470 0.02693438
## RTQUIC 0.05144085 0.05191224
```

Conclusion for PSP: There is no evidence of relationship between age and RTQUIC status within the PSP diagnosis. However, due to the much smaller size of this group, it is hard to tell whether overall age-related/pathology-specific changes in PSP are responsible for alpha-syn misfolding (as PSP subjects overall are older). Of note, among the 8 RTQUIC+ PSP subjects, only 1 has an onset age <65. Given the aSyn-SAA- is balanced between young-onset and late-onset, this would suggest that we are simply underpowered.

7.2. ASYN-SAA+ & AD: NUMERICAL VARIABLES

7.2.1. ASYN-SAA+ & ABETA42

Comparisons shown in: Table 2, Results > ASyn-SAA+ & AD+

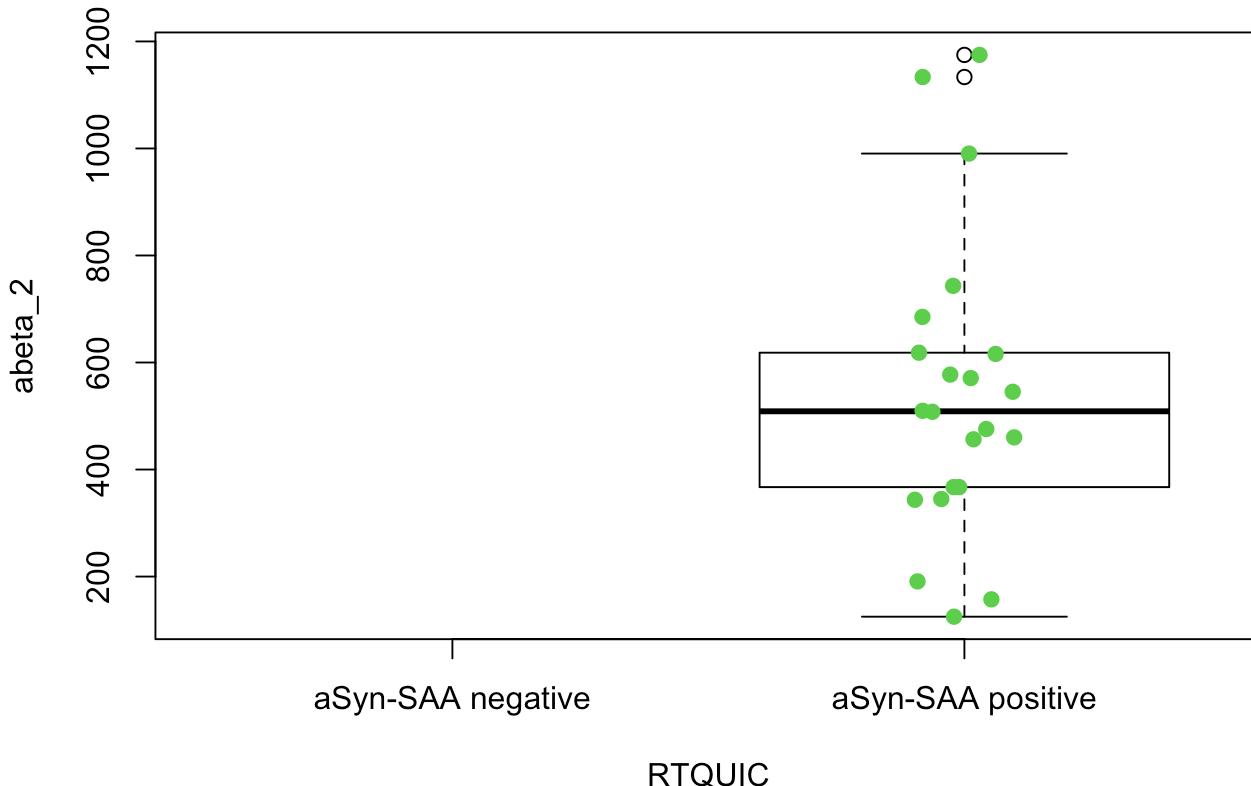
Due to relative low sample size and heterogeneity of the cohort (especially for RTQUIC+ or young-onset AD+ subjects) for model complexity, preferred approach is to run model on all points then run diagnostic analyses and present different model runs (leaving one outlier out)

ABETA STATS: OUTLIERS CHECK

```
#Compare RTQUIC+ vs RTQUIC-
boxplot(abeta_2 ~ RTQUIC, data= RTposdf, col = "white")$out
```

```
## [1] 1133.521 1174.820
```

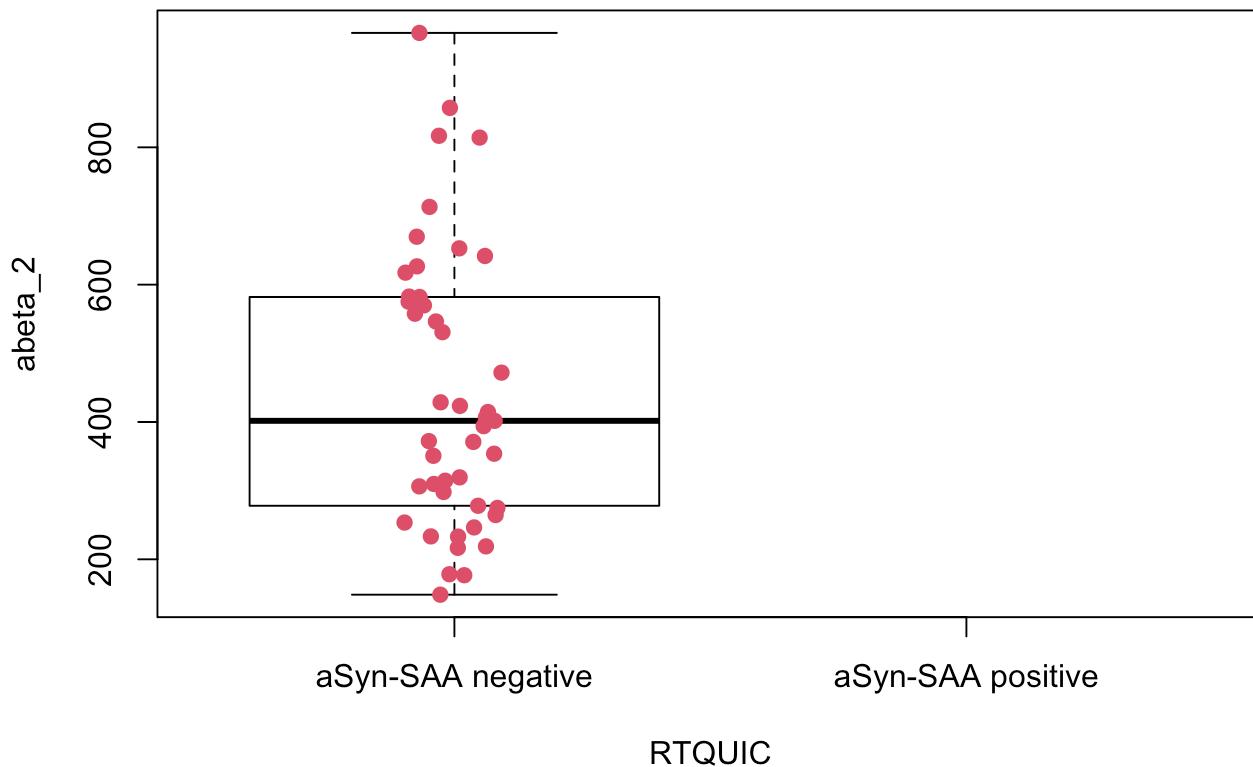
```
stripchart(abeta_2 ~ RTQUIC, data = RTposdf, method = "jitter", pch = 19, col = 2:4,
vertical = TRUE, add = TRUE)
```



```
boxplot(abeta_2 ~ RTQUIC, data= RTnegdf, col = "white")$out
```

```
## numeric(0)
```

```
stripchart(abeta_2 ~ RTQUIC, data = RTnegdf, method = "jitter", pch = 19, col = 2:4,
vertical = TRUE, add = TRUE)
```



ABETA STATS: DISTRIBUTION

```
shapiro.test(RTposdf$logabeta) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: RTposdf$logabeta  
## W = 0.9362, p-value = 0.1651
```

```
shapiro.test(RTnegdf$logabeta) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: RTnegdf$logabeta  
## W = 0.97854, p-value = 0.5616
```

```
shapiro.test(CBSdf$logabeta) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: CBSdf$logabeta  
## W = 0.98765, p-value = 0.9391
```

```
shapiro.test(PSPdf$logabeta) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: PSPdf$logabeta  
## W = 0.94239, p-value = 0.1272
```

```
leveneTest(logabeta ~ DX_APD*RTQUIC, df) #homoscedasticity
```

```
## Levene's Test for Homogeneity of Variance (center = median)  
##          Df F value Pr(>F)  
## group     3  0.4888 0.6913  
##          63
```

ABETA STATS: LINEAR REGRESSION MODEL SELECTION

```
test1 <- lm(logabeta ~ RTQUIC, df)  
test2 <- lm(logabeta ~ RTQUIC + DX_APD, df)  
test3 <- lm(logabeta ~ RTQUIC + Sex, df) #not adding any value to the model  
anova(test1, test2)
```

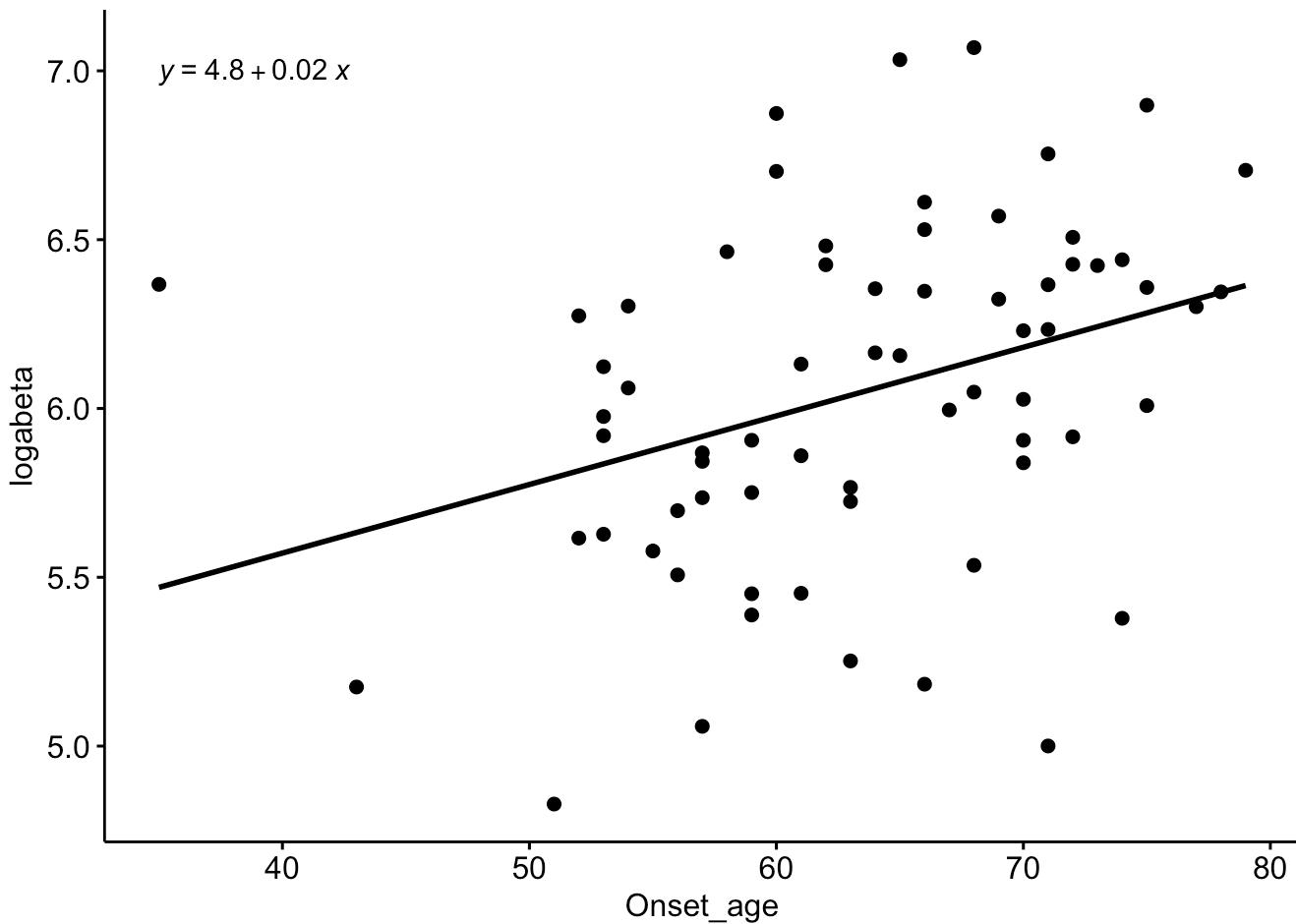
```
## Analysis of Variance Table  
##  
## Model 1: logabeta ~ RTQUIC  
## Model 2: logabeta ~ RTQUIC + DX_APD  
##          Res.Df   RSS Df Sum of Sq    F   Pr(>F)  
## 1       65 16.443  
## 2       64 15.658  1   0.78431 3.2057 0.07811 .  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(test1, test3)
```

```
## Analysis of Variance Table
##
## Model 1: logabeta ~ RTQUIC
## Model 2: logabeta ~ RTQUIC + Sex
##   Res.Df   RSS Df Sum of Sq    F Pr(>F)
## 1     65 16.443
## 2     64 16.432  1  0.01032 0.0402 0.8417
```

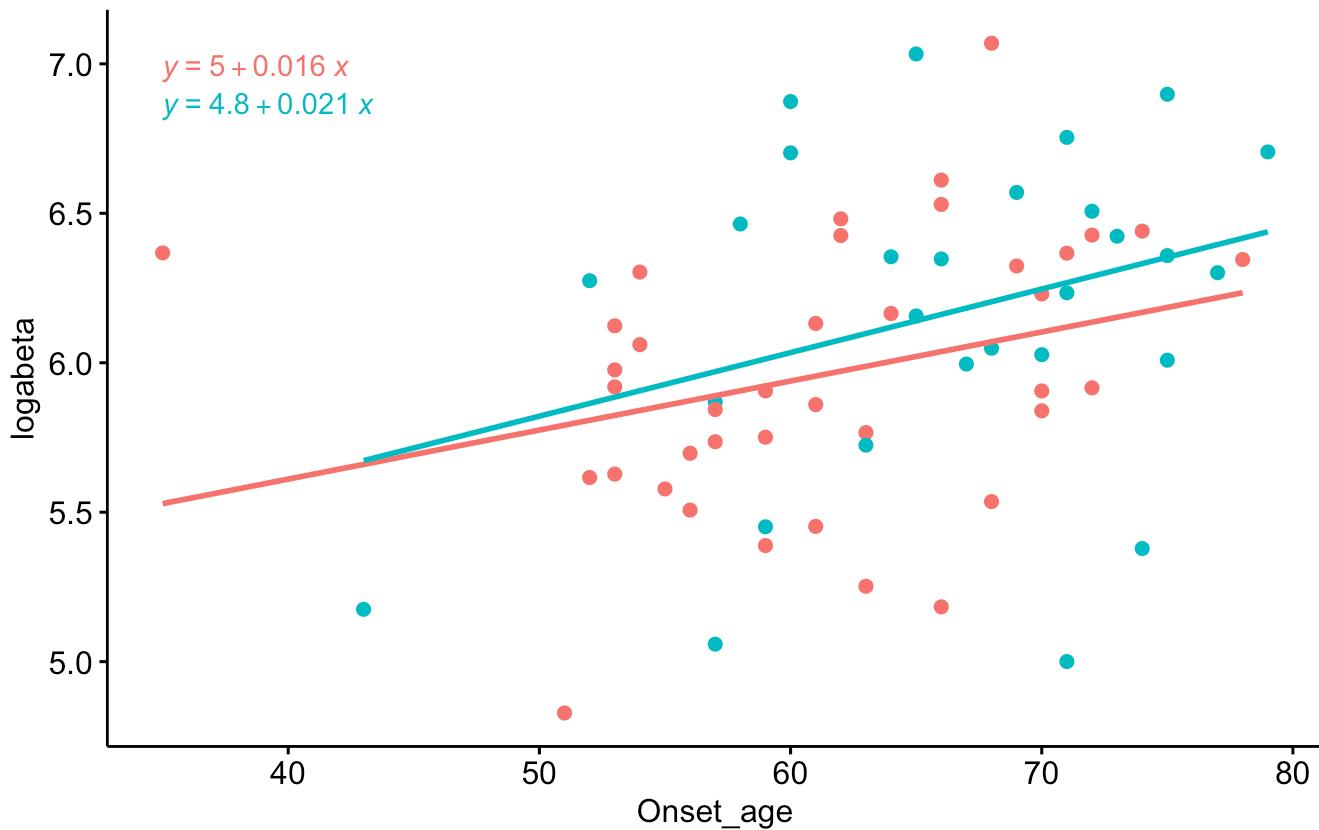
Inclusion of Onset_age as a covariate. Check independence of DV and covariate (not expected in observational data) + homogeneity of regression slopes.

```
ggscatter(df, x = "Onset_age", y = "logabeta", add = "reg.line")+
  stat_regrline_equation(aes())
```



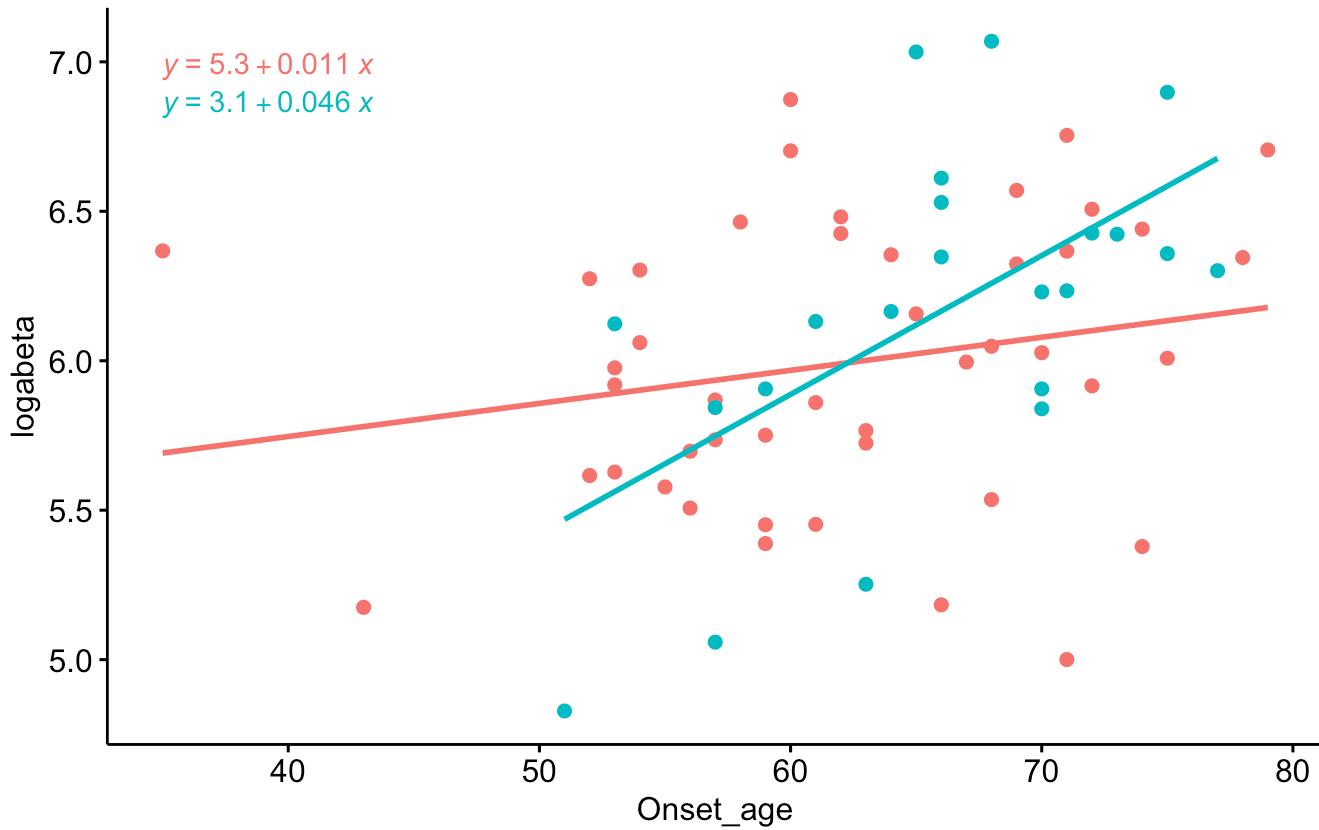
```
ggscatter(df, x = "Onset_age", y = "logabeta", color = "DX_APD", add = "reg.line")+
  stat_regrline_equation(aes(color = DX_APD))
```

DX_APD CBS PSP



```
ggscatter(df, x = "Onset_age", y = "logabeta", color = "RTQUIC", add = "reg.line")+
  stat_regrline_equation(aes(color = RTQUIC))
```

RTQUIC ● aSyn-SAA negative ● aSyn-SAA positive



```
cor.test(df$logabeta, df$Onset_age) #corr
```

```
## 
## Pearson's product-moment correlation
## 
## data: df$logabeta and df$Onset_age
## t = 2.9784, df = 65, p-value = 0.004069
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1159849 0.5416588
## sample estimates:
##        cor
## 0.3465388
```

```
cor.test(df$logabeta, df$Onset_age, method="spearman") #corr
```

```
## Warning in cor.test.default(df$logabeta, df$Onset_age, method = "spearman"):
## Cannot compute exact p-value with ties
```

```
## 
## Spearman's rank correlation rho
##
## data: df$logabeta and df$Onset_age
## S = 30561, p-value = 0.001097
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.3902022
```

```
summary(lm(df$logabeta ~ df$Onset_age)) #linear relationship
```

```
## 
## Call:
## lm(formula = df$logabeta ~ df$Onset_age)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.20118 -0.27489  0.03238  0.29163  0.95346
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.758811  0.436583 10.900 2.59e-16 ***
## df$Onset_age 0.020320  0.006822   2.978  0.00407 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4777 on 65 degrees of freedom
## Multiple R-squared:  0.1201, Adjusted R-squared:  0.1066
## F-statistic: 8.871 on 1 and 65 DF,  p-value: 0.004069
```

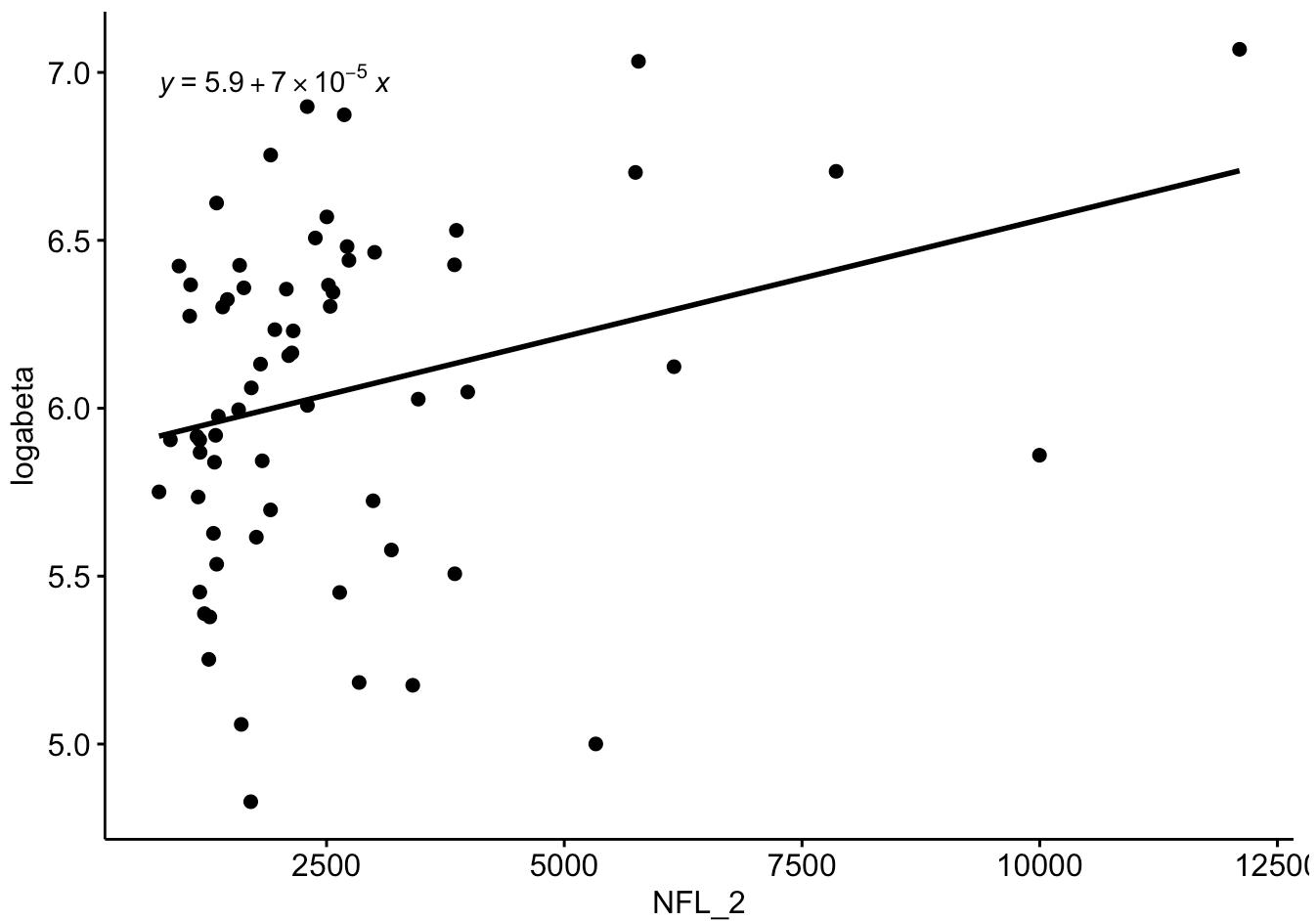
Inclusion of NFL as a covariate. Check independence of DV and covariate (not expected in observational data) + homogeneity of regression slopes.

```
ggscatter(df, x = "NFL_2", y = "logabeta", add = "reg.line")+
  stat_regress_line(aes()))
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_regress_line()`).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



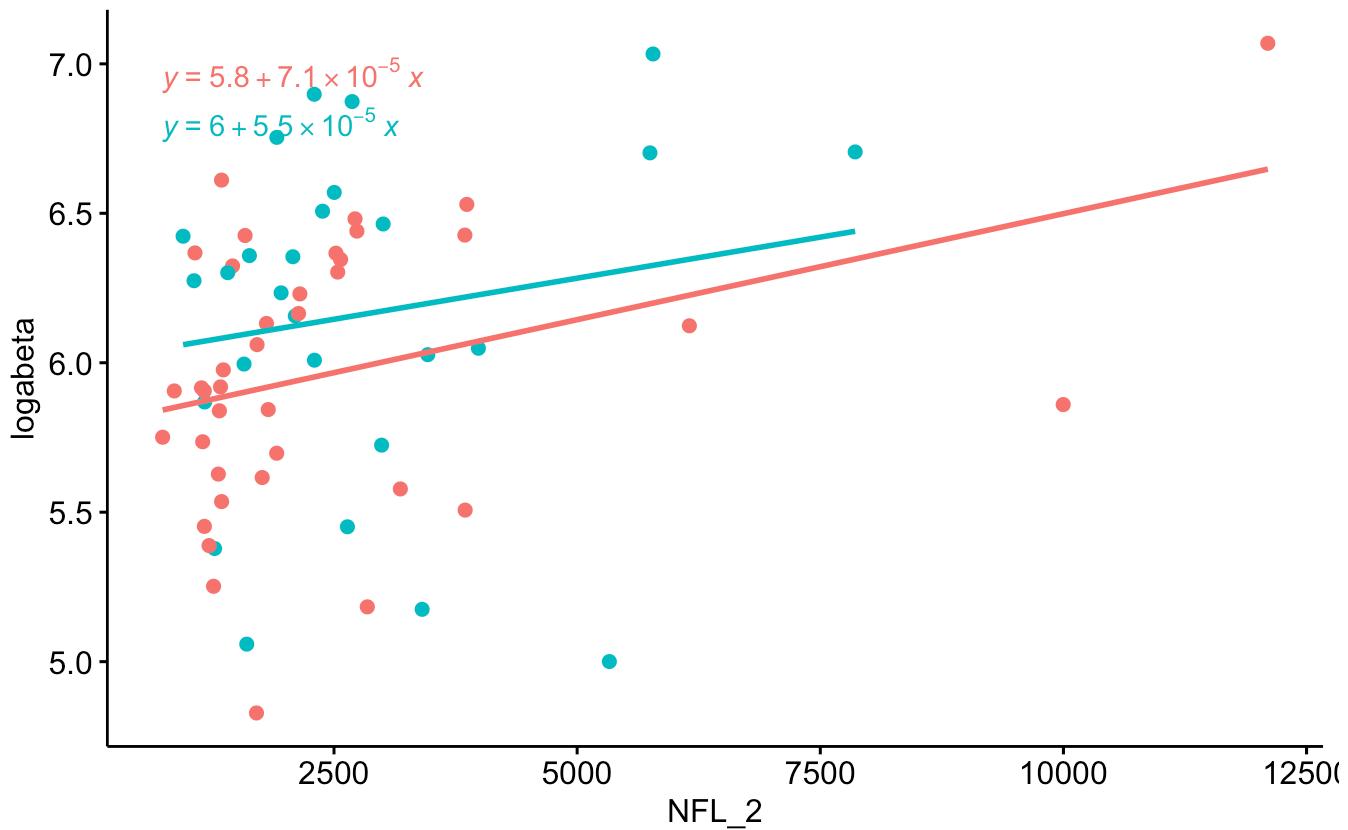
```
ggscatter(df, x = "NFL_2", y = "logabeta", color = "DX_APD", add = "reg.line")+
  stat_regline_equation(aes(color = DX_APD))
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_regline_equation()`).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

DX_APD CBS PSP



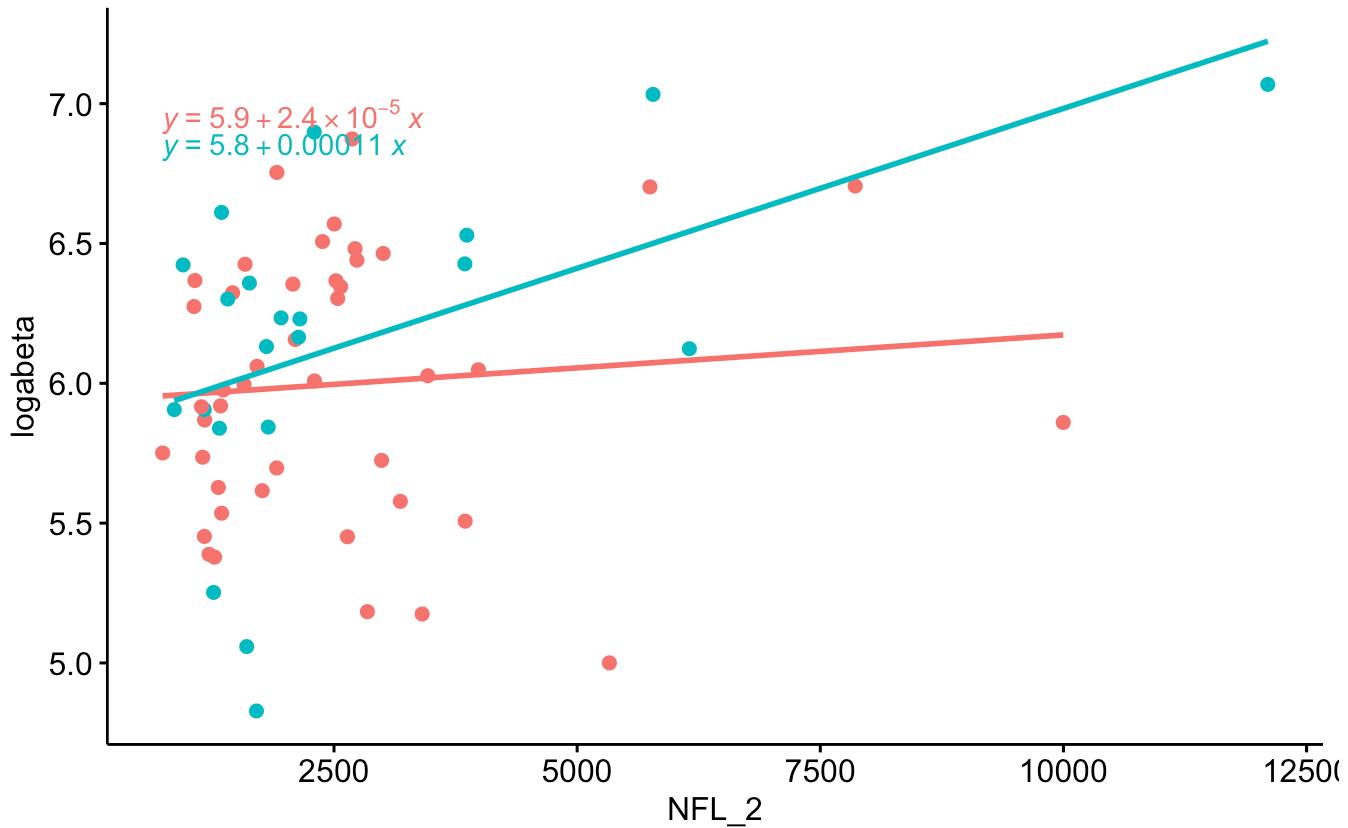
```
ggscatter(df, x = "NFL_2", y = "logabeta", color = "RTQUIC", add = "reg.line")+
  stat_regline_equation(aes(color = RTQUIC))
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_regline_equation()`).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

RTQUIC ● aSyn-SAA negative ● aSyn-SAA positive



```
cor.test(df$logabeta, df$NFL_2) #corr
```

```
## 
## Pearson's product-moment correlation
## 
## data: df$logabeta and df$NFL_2
## t = 2.3309, df = 63, p-value = 0.02297
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.04065892 0.49186131
## sample estimates:
##        cor
## 0.2817639
```

```
cor.test(df$logabeta, df$NFL_2, method="spearman") #corr
```

```
## 
## Spearman's rank correlation rho
##
## data: df$logabeta and df$NFL_2
## S = 32704, p-value = 0.02155
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.2853147
```

summary(lm(df\$logabeta ~ df\$NFL_2)) #linear relationship

```
## 
## Call:
## lm(formula = df$logabeta ~ df$NFL_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2359 -0.3291  0.0206  0.3616  0.8730
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 5.865e+00  9.915e-02 59.157   <2e-16 ***
## df$NFL_2    6.956e-05  2.984e-05  2.331    0.023 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4938 on 63 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.07939,    Adjusted R-squared:  0.06478
## F-statistic: 5.433 on 1 and 63 DF,  p-value: 0.02297
```

Inclusion of other variables as covariate:
summary(lm(df\$logabeta ~ df\$LP2_Disease_Duration)) #no linear relationship

```

## 
## Call:
## lm(formula = df$logabeta ~ df$LP2_Disease_Duration)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -1.15064 -0.33198  0.04481  0.34365  1.06798 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             5.94199   0.11434  51.969 <2e-16 ***
## df$LP2_Disease_Duration 0.02008   0.01833   1.095   0.277    
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.5046 on 65 degrees of freedom
## Multiple R-squared:  0.01813, Adjusted R-squared:  0.003022 
## F-statistic:  1.2 on 1 and 65 DF,  p-value: 0.2774

```

```
summary(lm(df$logabeta ~ df$ttau_2)) #no linear relationship
```

```

## 
## Call:
## lm(formula = df$logabeta ~ df$ttau_2)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -1.12024 -0.31782  0.04374  0.30155  1.09479 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  6.0839558  0.1038904  58.561 <2e-16 ***
## df$ttau_2   -0.0001096  0.0002076  -0.528   0.599    
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.5085 on 64 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.004338, Adjusted R-squared: -0.01122 
## F-statistic: 0.2788 on 1 and 64 DF,  p-value: 0.5993

```

```
summary(lm(df$logabeta ~ df$pptau_2)) #no linear relationship
```

```
##  
## Call:  
## lm(formula = df$logabeta ~ df$ptau_2)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -1.20862 -0.31758  0.03157  0.34593  1.02863  
##  
## Coefficients:  
##             Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 6.0566369  0.1276037 47.464 <2e-16 ***  
## df$ptau_2   -0.0001706  0.0020746 -0.082    0.935  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 0.5092 on 65 degrees of freedom  
## Multiple R-squared:  0.000104, Adjusted R-squared: -0.01528  
## F-statistic: 0.006759 on 1 and 65 DF, p-value: 0.9347
```

```
# Comparison of complex models  
# For biomarkers, only kept logged value on one side of the model equation  
stdmlr <- lm(logabeta ~ scale(Onset_age)*RTQUIC + DX_APD + scale(NFL_2), df)  
summary(stdmlr)
```

```

## 
## Call:
## lm(formula = logabeta ~ scale(Onset_age) * RTQUIC + DX_APD +
##     scale(NFL_2), data = df)
##
## Residuals:
##      Min      1Q  Median      3Q     Max 
## -1.28752 -0.32248 -0.01025  0.34356  0.84231 
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                5.97571   0.09025  66.216 <2e-16 ***
## scale(Onset_age)            0.06085   0.06775   0.898  0.3728    
## RTQUICaSyn-SAA positive    0.03079   0.12848   0.240  0.8115    
## DX_APDPSP                  0.07497   0.12250   0.612  0.5429    
## scale(NFL_2)                0.13973   0.05786   2.415  0.0189 ***
## scale(Onset_age):RTQUICaSyn-SAA positive  0.33499   0.13768   2.433  0.0180  
## 
## (Intercept)                 ***
## scale(Onset_age)             * 
## RTQUICaSyn-SAA positive    * 
## DX_APDPSP                   --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
##
## Residual standard error: 0.4554 on 59 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.2668, Adjusted R-squared:  0.2047 
## F-statistic: 4.294 on 5 and 59 DF,  p-value: 0.002131

```

```
AIC(stdmlr)
```

```
## [1] 89.90517
```

```
stdmlr2 <- lm(logabeta ~ scale(Onset_age)*RTQUIC + DX_APD, df)
summary(stdmlr2)
```

```

## 
## Call:
## lm(formula = logabeta ~ scale(Onset_age) * RTQUIC + DX_APD, data = df)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -1.13342 -0.28168 -0.05665  0.34966  0.85374 
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)               5.95824   0.09015  66.093 <2e-16 ***
## scale(Onset_age)          0.08251   0.06859   1.203  0.2336    
## RTQUICaSyn-SAA positive   0.05793   0.12826   0.452  0.6531    
## DX_APDPSP                 0.10296   0.12256   0.840  0.4041    
## scale(Onset_age):RTQUICaSyn-SAA positive  0.29171   0.13990   2.085  0.0412    
## 
## (Intercept) *** 
## scale(Onset_age) 
## RTQUICaSyn-SAA positive 
## DX_APDPSP 
## scale(Onset_age):RTQUICaSyn-SAA positive * 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.4666 on 62 degrees of freedom 
## Multiple R-squared:  0.1993, Adjusted R-squared:  0.1477 
## F-statistic: 3.858 on 4 and 62 DF,  p-value: 0.007301

```

```
AIC(stdmlr2)
```

```
## [1] 94.78542
```

```
stdmlr3 <- lm(logabeta ~ scale(Onset_age)*RTQUIC, df)
summary(stdmlr3)
```

```
##  
## Call:  
## lm(formula = logabeta ~ scale(Onset_age) * RTQUIC, data = df)  
##  
## Residuals:  
##      Min      1Q  Median      3Q     Max  
## -1.08935 -0.27009 -0.01783  0.35302  0.91339  
##  
## Coefficients:  
##                               Estimate Std. Error t value Pr(>|t|)  
## (Intercept)                 6.0058    0.0700 85.793   <2e-16  
## scale(Onset_age)            0.0954    0.0667  1.430   0.1576  
## RTQUICaSyn-SAA positive     0.0404    0.1263  0.320   0.7500  
## scale(Onset_age):RTQUICaSyn-SAA positive  0.3050    0.1387  2.199   0.0315  
##  
## (Intercept)                  ***  
## scale(Onset_age)  
## RTQUICaSyn-SAA positive  
## scale(Onset_age):RTQUICaSyn-SAA positive *  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 0.4655 on 63 degrees of freedom  
## Multiple R-squared:  0.1902, Adjusted R-squared:  0.1516  
## F-statistic: 4.932 on 3 and 63 DF,  p-value: 0.003858
```

```
AIC(stdmlr3)
```

```
## [1] 93.54383
```

```
stdmlr4 <- lm(logabeta ~ scale(Onset_age)+ RTQUIC, df)  
summary(stdmlr4)
```

```
##  
## Call:  
## lm(formula = logabeta ~ scale(Onset_age) + RTQUIC, data = df)  
##  
## Residuals:  
##      Min      1Q  Median      3Q     Max  
## -1.16120 -0.25300 -0.00915  0.32704  0.92406  
##  
## Coefficients:  
##                               Estimate Std. Error t value Pr(>|t|)  
## (Intercept)             6.01558   0.07192 83.637 < 2e-16 ***  
## scale(Onset_age)        0.16594   0.06020  2.756  0.00761 **  
## RTQUICaSyn-SAA positive 0.09714   0.12724  0.763  0.44798  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 0.4792 on 64 degrees of freedom  
## Multiple R-squared:  0.128, Adjusted R-squared:  0.1008  
## F-statistic: 4.699 on 2 and 64 DF,  p-value: 0.01247
```

```
AIC(stdmlr4)
```

```
## [1] 96.49943
```

```
stdmlr5 <- lm(logabeta ~ scale(Onset_age)*RTQUIC + scale(NFL_2), df)  
summary(stdmlr5)
```

```

## 
## Call:
## lm(formula = logabeta ~ scale(Onset_age) * RTQUIC + scale(NFL_2),
##      data = df)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -1.2585 -0.3092  0.0265  0.3361  0.8854 
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                6.01111   0.06890  87.243 <2e-16 ***
## scale(Onset_age)            0.06995   0.06576   1.064  0.2917    
## RTQUICaSyn-SAA positive     0.01484   0.12515   0.119  0.9060    
## scale(NFL_2)                0.14166   0.05747   2.465  0.0166    
## scale(Onset_age):RTQUICaSyn-SAA positive  0.34509   0.13598   2.538  0.0138  
## 
## (Intercept)                 ***
## scale(Onset_age)             * 
## RTQUICaSyn-SAA positive    
## scale(NFL_2)                  * 
## scale(Onset_age):RTQUICaSyn-SAA positive * 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.453 on 60 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.2622, Adjusted R-squared:  0.213 
## F-statistic: 5.329 on 4 and 60 DF,  p-value: 0.0009697

```

AIC(stdmlr5)

```
## [1] 88.3165
```

The model with the lowest AIC is the one including an interaction term of Onset age by aSyn-SAA status as NfL levels. Its AIC is almost identical to the one that incorporates Diagnosis. Diagnosis is an extremely important covariate that is relevant clinically. Therefore, we prefer to report the model with the lowest AIC that still includes diagnosis, model stdmlr. Of note, all models have similar AIC and the interaction term was significant in every model that it was included in. Removing NfL did not affect the results, which is important since NfL outliers were not removed prior to running the analysis. “)

ABETA STATS: MULTIVARIABLE LINEAR REGRESSION MODEL

```
stdmlr <- lm(logabeta ~ scale(Onset_age)*RTQUIC + DX_APD + scale(NFL_2), df)
summary(stdmlr)
```

```

## 
## Call:
## lm(formula = logabeta ~ scale(Onset_age) * RTQUIC + DX_APD +
##     scale(NFL_2), data = df)
##
## Residuals:
##      Min      1Q  Median      3Q     Max 
## -1.28752 -0.32248 -0.01025  0.34356  0.84231 
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                5.97571   0.09025  66.216 <2e-16 ***
## scale(Onset_age)           0.06085   0.06775   0.898  0.3728    
## RTQUICaSyn-SAA positive    0.03079   0.12848   0.240  0.8115    
## DX_APDPSP                  0.07497   0.12250   0.612  0.5429    
## scale(NFL_2)                0.13973   0.05786   2.415  0.0189 ***
## scale(Onset_age):RTQUICaSyn-SAA positive  0.33499   0.13768   2.433  0.0180  
## 
## (Intercept)                 ***
## scale(Onset_age)             * 
## RTQUICaSyn-SAA positive    
## DX_APDPSP                    *
## scale(NFL_2)                  *
## scale(Onset_age):RTQUICaSyn-SAA positive * 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.4554 on 59 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.2668, Adjusted R-squared:  0.2047 
## F-statistic: 4.294 on 5 and 59 DF,  p-value: 0.002131

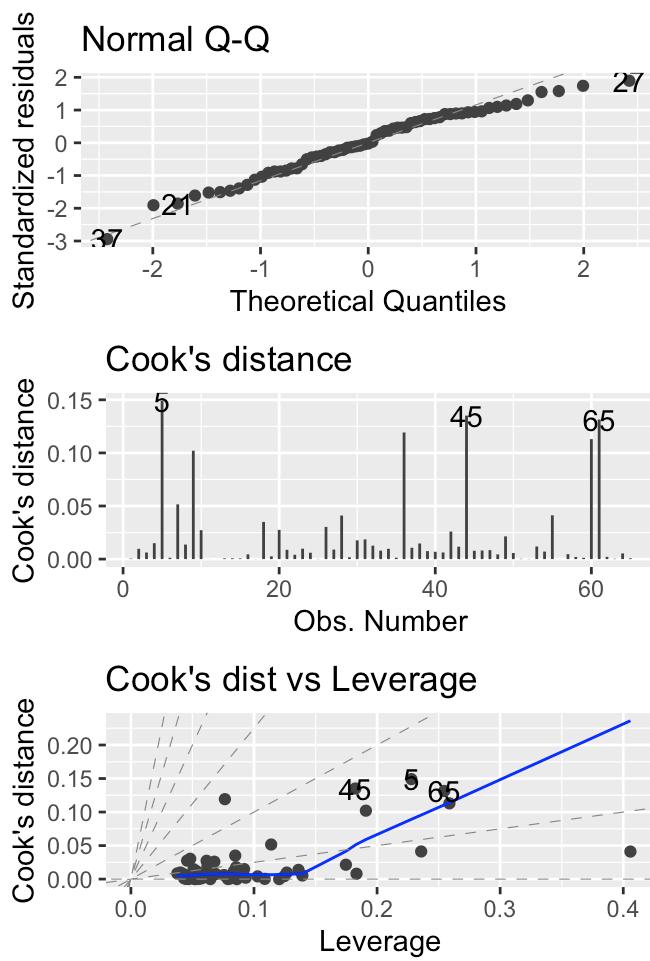
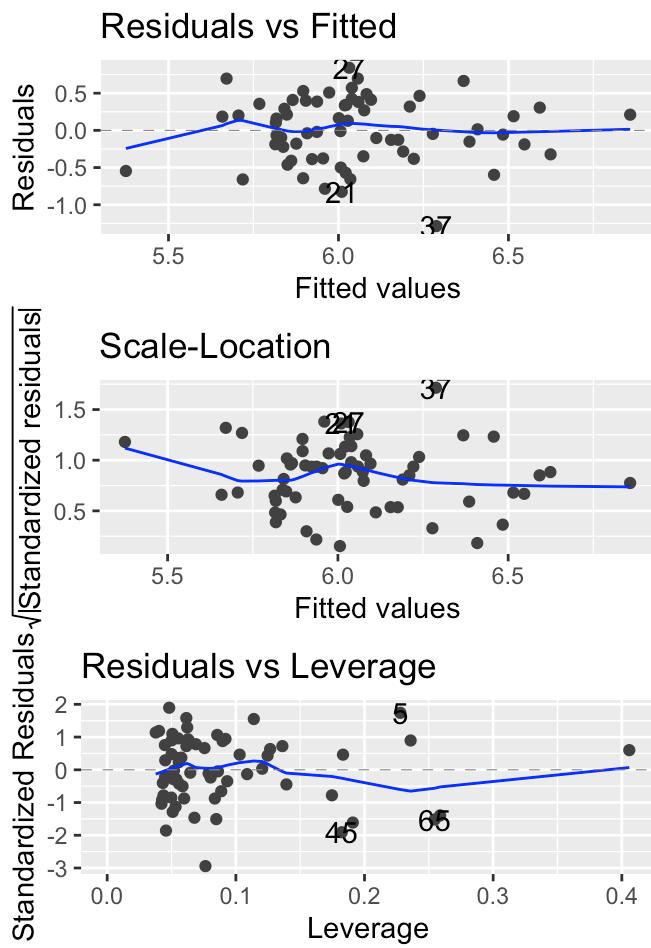
```

ABETA STATS: MULTIVARIABLE LINEAR REGRESSION MODEL DIAGNOSTICS

```
check_normality(stdmlr) #Ok normality of residuals
```

```
## OK: residuals appear as normally distributed (p = 0.267).
```

```
autoplot(stdmlr, which = 1:6) #All plots including scale location plot
```



```
# Visualize the values for each of the IDs that are indexed on above plots
df[5, c("DX_APD", "abeta_2", "logabeta", "Onset_age", "RTQUIC", "NFL_2")]
```

```
##   DX_APD abeta_2 logabeta Onset_age          RTQUIC   NFL_2
## 5    CBS  582.615  6.367527      35 aSyn-SAA negative 1070.23
```

```
df[21, c("DX_APD", "abeta_2", "logabeta", "Onset_age", "RTQUIC", "NFL_2")]
```

```
##   DX_APD abeta_2 logabeta Onset_age          RTQUIC   NFL_2
## 22   CBS   557.7  6.323821      69 aSyn-SAA negative 1456.97
```

```
df[27, c("DX_APD", "abeta_2", "logabeta", "Onset_age", "RTQUIC", "NFL_2")]
```

```
##   DX_APD abeta_2 logabeta Onset_age          RTQUIC   NFL_2
## 28   PSP  641.779  6.464244      58 aSyn-SAA negative 3005.4
```

```
df[37, c("DX_APD", "abeta_2", "logabeta", "Onset_age", "RTQUIC", "NFL_2")]
```

```
##   DX_APD abeta_2 logabeta Onset_age          RTQUIC   NFL_2
## 38   PSP  713.237  6.569814      69 aSyn-SAA negative 2502.6
```

```
df[45, c("DX_APD", "abeta_2", "logabeta", "Onset_age", "RTQUIC", "NFL_2")]
```

```
##      DX_APD abeta_2 logabeta Onset_age          RTQUIC NFL_2
## 46      CBS  319.35  5.766288       63 aSyn-SAA negative     NA
```

```
df[65, c("DX_APD", "abeta_2", "logabeta", "Onset_age", "RTQUIC", "NFL_2")]
```

```
##      DX_APD abeta_2 logabeta Onset_age          RTQUIC NFL_2
## 67      CBS  309.75  5.735766       57 aSyn-SAA negative 1149.77
```

```
# for-loop to test the model without each of these values (ie once without outlier 1, then outlier 2, etc)
vecIDs <- df[c(5,21,27,37,45,65), "ID"] #Create vector of each ID
for (i in vecIDs) {
  test <- subset(df, ID!=i) #for some analysis, need to exclude the potential false negative too
  teststdmlr <- lm(logabeta ~ scale(Onset_age)*RTQUIC + DX_APD + scale(NFL_2), test)
  print(summary(teststdmlr))
}
```

```

## 
## Call:
## lm(formula = logabeta ~ scale(Onset_age) * RTQUIC + DX_APD +
##     scale(NFL_2), data = test)
##
## Residuals:
##      Min      1Q   Median      3Q      Max 
## -1.33044 -0.31106 -0.00376  0.32119  0.87557 
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 5.96938   0.08974  66.519 <2e-16  
## scale(Onset_age)            0.10900   0.06813   1.600  0.1150  
## RTQUICaSyn-SAA positive    0.05744   0.12501   0.460  0.6476  
## DX_APDPSP                  0.07842   0.12035   0.652  0.5172  
## scale(NFL_2)                0.14222   0.05705   2.493  0.0155  
## scale(Onset_age):RTQUICaSyn-SAA positive 0.25439   0.12800   1.987  0.0516 
## 
## (Intercept)                 ***      
## scale(Onset_age)            *        
## RTQUICaSyn-SAA positive    .        
## DX_APDPSP                  ---    
## scale(NFL_2)                *        
## scale(Onset_age):RTQUICaSyn-SAA positive .
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
##
## Residual standard error: 0.4473 on 58 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.3001, Adjusted R-squared:  0.2398 
## F-statistic: 4.974 on 5 and 58 DF,  p-value: 0.0007393
## 
## 
## Call:
## lm(formula = logabeta ~ scale(Onset_age) * RTQUIC + DX_APD +
##     scale(NFL_2), data = test)
##
## Residuals:
##      Min      1Q   Median      3Q      Max 
## -1.28271 -0.33144  0.00364  0.34214  0.84028 
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 5.95855   0.09254  64.390 <2e-16  
## scale(Onset_age)            0.05162   0.06904   0.748  0.4577  
## RTQUICaSyn-SAA positive    0.04082   0.12993   0.314  0.7546  
## DX_APDPSP                  0.09100   0.12409   0.733  0.4663  
## scale(NFL_2)                0.14460   0.05848   2.473  0.0164  
## scale(Onset_age):RTQUICaSyn-SAA positive 0.34232   0.13872   2.468  0.0166 
## 
## (Intercept)                 ***      
## scale(Onset_age)            *        

```

```

## RTQUICaSyn-SAA positive
## DX_APDPSP
## scale(NFL_2) *
## scale(Onset_age):RTQUICaSyn-SAA positive *
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4563 on 58 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared: 0.273, Adjusted R-squared: 0.2103
## F-statistic: 4.356 on 5 and 58 DF, p-value: 0.001962
##
##
## Call:
## lm(formula = logabeta ~ scale(Onset_age) * RTQUIC + DX_APD +
##     scale(NFL_2), data = test)
##
## Residuals:
##      Min      1Q Median      3Q      Max
## -1.27180 -0.31960 -0.00466  0.34186  0.86515
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  5.97557   0.09046 66.060 <2e-16
## scale(Onset_age)              0.06845   0.06854  0.999  0.3221
## RTQUICaSyn-SAA positive       0.03958   0.12841  0.308  0.7590
## DX_APDPSP                     0.05546   0.12425  0.446  0.6570
## scale(NFL_2)                   0.13908   0.05837  2.383  0.0205
## scale(Onset_age):RTQUICaSyn-SAA positive  0.33400   0.13843  2.413  0.0190
##
## (Intercept) ***

## scale(Onset_age)
## RTQUICaSyn-SAA positive
## DX_APDPSP
## scale(NFL_2) *
## scale(Onset_age):RTQUICaSyn-SAA positive *
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4557 on 58 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared: 0.2706, Adjusted R-squared: 0.2077
## F-statistic: 4.303 on 5 and 58 DF, p-value: 0.002136
##
##
## Call:
## lm(formula = logabeta ~ scale(Onset_age) * RTQUIC + DX_APD +
##     scale(NFL_2), data = test)
##
## Residuals:
##      Min      1Q Median      3Q      Max
## -1.26254 -0.31886  0.00363  0.35433  0.86131

```

```

## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|) 
## (Intercept)                5.97036   0.09004 66.308 <2e-16
## scale(Onset_age)           0.05456   0.06819  0.800  0.4270
## RTQUICaSyn-SAA positive    0.03655   0.12882  0.284  0.7776
## DX_APDPSP                  0.05824   0.12322  0.473  0.6382
## scale(NFL_2)                0.14270   0.05824  2.450  0.0173
## scale(Onset_age):RTQUICaSyn-SAA positive 0.34748   0.13841  2.511  0.0149
## 
## (Intercept)                 ***
## scale(Onset_age)             *
## RTQUICaSyn-SAA positive     *
## DX_APDPSP                   *
## scale(NFL_2)                 *
## scale(Onset_age):RTQUICaSyn-SAA positive *
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4546 on 58 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.2696, Adjusted R-squared:  0.2067
## F-statistic: 4.283 on 5 and 58 DF,  p-value: 0.002205
## 
## 
## Call:
## lm(formula = logabeta ~ scale(Onset_age) * RTQUIC + DX_APD +
##      scale(NFL_2), data = test)
## 
## Residuals:
##    Min      1Q      Median      3Q      Max 
## -1.28752 -0.32248 -0.01025  0.34356  0.84231
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|) 
## (Intercept)                5.97575   0.09026 66.208 <2e-16
## scale(Onset_age)           0.06132   0.06827  0.898  0.3728
## RTQUICaSyn-SAA positive    0.03103   0.12846  0.242  0.8099
## DX_APDPSP                  0.07497   0.12250  0.612  0.5429
## scale(NFL_2)                0.13973   0.05786  2.415  0.0189
## scale(Onset_age):RTQUICaSyn-SAA positive 0.33755   0.13873  2.433  0.0180
## 
## (Intercept)                 ***
## scale(Onset_age)             *
## RTQUICaSyn-SAA positive     *
## DX_APDPSP                   *
## scale(NFL_2)                 *
## scale(Onset_age):RTQUICaSyn-SAA positive *
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4554 on 59 degrees of freedom

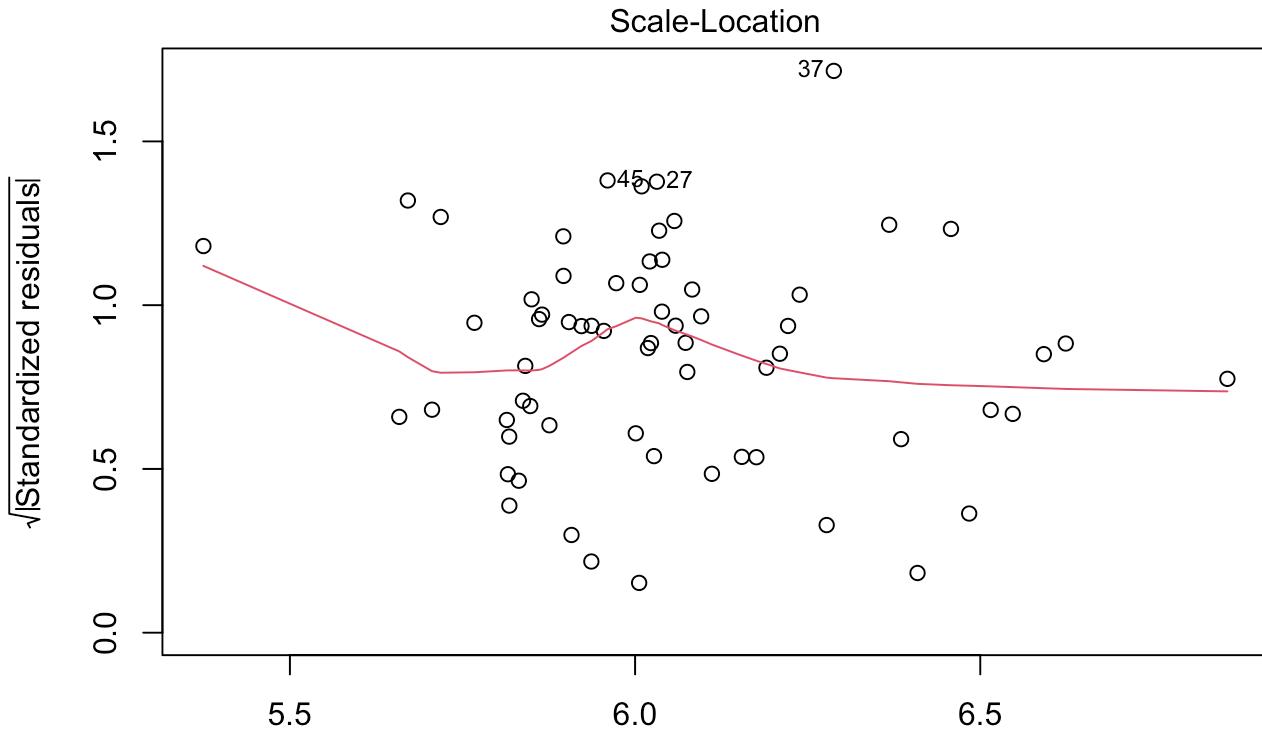
```

```

## (1 observation deleted due to missingness)
## Multiple R-squared:  0.2668, Adjusted R-squared:  0.2047
## F-statistic: 4.294 on 5 and 59 DF,  p-value: 0.002131
##
##
## Call:
## lm(formula = logabeta ~ scale(Onset_age) * RTQUIC + DX_APD +
##     scale(NFL_2), data = test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.28644 -0.32886  0.00183  0.34507  0.84133
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 5.98127   0.09252  64.647 <2e-16  
## scale(Onset_age)            0.06030   0.06863   0.879  0.3832  
## RTQUICaSyn-SAA positive    0.03199   0.12976   0.247  0.8061  
## DX_APDPSP                  0.07233   0.12412   0.583  0.5623  
## scale(NFL_2)                0.13942   0.05870   2.375  0.0209  
## scale(Onset_age):RTQUICaSyn-SAA positive 0.33745   0.13942   2.420  0.0187  
## 
## (Intercept)                   ***
## scale(Onset_age)                
## RTQUICaSyn-SAA positive      
## DX_APDPSP                     *
## scale(NFL_2)                   *
## scale(Onset_age):RTQUICaSyn-SAA positive *
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4591 on 58 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.263, Adjusted R-squared:  0.1995 
## F-statistic: 4.14 on 5 and 58 DF,  p-value: 0.002771

```

```
plot(stdmlr, which = 3) # 3 = Scale-Location plot. Variance of residuals
```



```
bptest(stdmlr) #Ok variance of residulas. Breusch-Pagan test for heterodasticity.
```

```
## 
## studentized Breusch-Pagan test
## 
## data: stdmlr
## BP = 7.4909, df = 5, p-value = 0.1866
```

```
durbinWatsonTest(stdmlr) #Ok autocorrelation of residuals
```

```
##   lag Autocorrelation D-W Statistic p-value
##   1      0.1083846     1.781022    0.36
## Alternative hypothesis: rho != 0
```

```
# Multicollinearity checks
car::vif(stdmlr) #no multicollinearity at all
```

```
## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif
```

```
## scale(Onset_age) RTQUIC DX_APD
## 1.459046 1.131585 1.142161
## scale(NFL_2) scale(Onset_age):RTQUIC
## 1.033238 1.443803
```

ABETA STATS: MULTIVARIABLE LINEAR REGRESSION MODEL SIMPLE SLOPES

```
emtrends(stdmlr, pairwise ~ RTQUIC, var="Onset_age") # for interaction of Onset age by R TQUIC
```

```
## $emtrends
## RTQUIC Onset_age.trend SE df lower.CL upper.CL
## aSyn-SAA negative 0.00706 0.00786 59 -0.00867 0.0228
## aSyn-SAA positive 0.04593 0.01429 59 0.01733 0.0745
##
## Results are averaged over the levels of: DX_APD
## Confidence level used: 0.95
##
## $contrasts
## contrast estimate SE df t.ratio p.value
## (aSyn-SAA negative) - (aSyn-SAA positive) -0.0389 0.016 59 -2.433 0.0180
##
## Results are averaged over the levels of: DX_APD
```

ABETA STATS: MULTIVARIABLE LINEAR REGRESSION MODEL MAIN EFFECTS

```
emmeans(stdmlr, ~ RTQUIC) #adjusted means: cannot be interpreted due to interaction (slopes crossing each other)
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
## RTQUIC emmean SE df lower.CL upper.CL
## aSyn-SAA negative 6.01 0.0693 59 5.87 6.15
## aSyn-SAA positive 6.04 0.1093 59 5.82 6.26
##
## Results are averaged over the levels of: DX_APD
## Confidence level used: 0.95
```

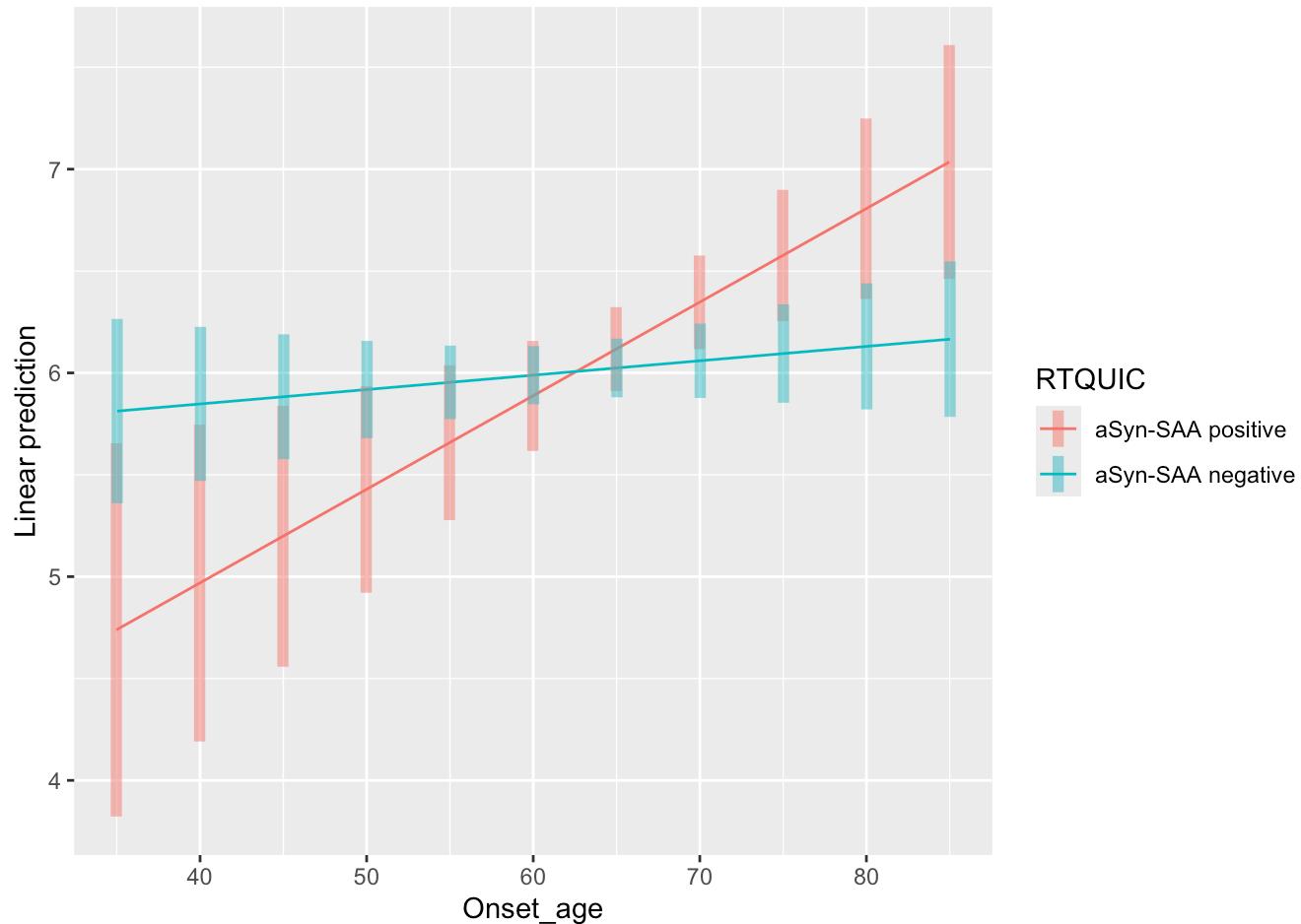
```
emmeans(stdmlr, ~ DX_APD) #adjusted means. Ok because no interaction.
```

```
## DX_APD emmean SE df lower.CL upper.CL
## CBS 5.99 0.0772 59 5.84 6.14
## PSP 6.07 0.1001 59 5.86 6.27
##
## Results are averaged over the levels of: RTQUIC
## Confidence level used: 0.95
```

7.2.2. FIGURE 1.B.

Figure 1.b.

```
##Create df for the figure based on the actual model  
mylist <- list(Onset_age=seq(35,85,by=5), RTQUIC=c("aSyn-SAA positive", "aSyn-SAA negative"))  
emmip(stdmlr, RTQUIC ~ Onset_age, at=mylist, CIs=TRUE)
```



```
#Dataset: data are from the model (predicted values)
figdf <- emmip(stdmlr, RTQUIC ~ Onset_age, at=mylist, CIs=TRUE, plotit=FALSE) #

label1 <- "paste(F*(5, 59)'==4.29*', ' ~ italic(p), \"< .01\"", " * italic(R)^2==20.47*%')" #First annotation of the plot: Model diagnostics. Comma has to be entered as text not in mathematical notation.
label2 <- "paste('*italic(p), \" < .05\"')" #Second annotation is p-value for the interaction

# Change name of variable RTQUIC
fig1b <- ggplot(data=figdf, aes(x=Onset_age,y=yvar, color=RTQUIC)) + #yvar is Abeta42 logged #Gplot figure basic layout

# Add the actual trend/slopes + CI around it
geom_line() + #DO NOT use show.legend as it will create a duplicate legend if combined with scale_color changes
  geom_ribbon(aes(ymax=UCL, ymin=LCL, fill=RTQUIC), alpha=0.2, show.legend=FALSE) + #show.legend=FALSE to avoid having both ribbon and line in legend

# Fix legends
scale_fill_manual(values=cbPalette_RTQUIC) +
  scale_color_manual(values=cbPalette_RTQUIC, name= "ASyn-SAA status", breaks=c("aSyn-SAA positive", "aSyn-SAA negative"), labels=c(expression(alpha*Syn-SAA+), expression(alpha*Syn-SAA-))) + #expression allows you to add greek letters
  scale_shape_manual(values=c(16,17), name="Diagnosis", breaks=c("CBS", "PSP"), labels=c("CBS", "PSP")) + #Color for the datapoints

# Add actual data
geom_point(data=df, aes(x=Onset_age, y=logabeta, shape=DX_APPD), size=4) + #Actual datapoints

# Annotate:
stat_regrline_equation(label.x=37, label.y=c(7.4, 7.2), size=6, show.legend=FALSE) + #shows the equation for each geom_line. Cannot use other options since it would not be based on the model

  annotate("text", size=5, x=70, y=4, label=label1, parse=TRUE)+ #label as defined above. Parse allows for use of mathematical notation
  annotate("text", size=6, x=40, y=7, label=label2, parse=TRUE)+

# Add the labels
  labs(title=expression(bold("Visual representation of the interaction of age at onset by "*alpha*Syn-SAA)), subtitle=expression("From the model: log(A"*beta*"42) ~ Age at onset * "*alpha*Syn-SAA + Diagnosis + NfL"), x="Age at onset (years)", y=expression(bold("CSF A"*beta*"42 levels (pg/mL) (log)")))+ #bold() is required otherwise the Y axis will not be bold in spite of element_text specification below

# Aesthetic only
  theme_classic() +
```

```

    theme(plot.title = element_text(size=16, hjust=0.5, face="bold")) +
    theme(plot.subtitle = element_text(size=14, hjust=0.5)) +
    theme(axis.text=element_text(size=16), axis.title=element_text(size=16,face="bold"))
+
    theme(legend.title = element_text(face="bold", size=16), legend.text= element_text(size=14))

fig1b

```

```

## Warning in summary.lm(res.lm): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(res.lm): essentially perfect fit: summary may be
## unreliable

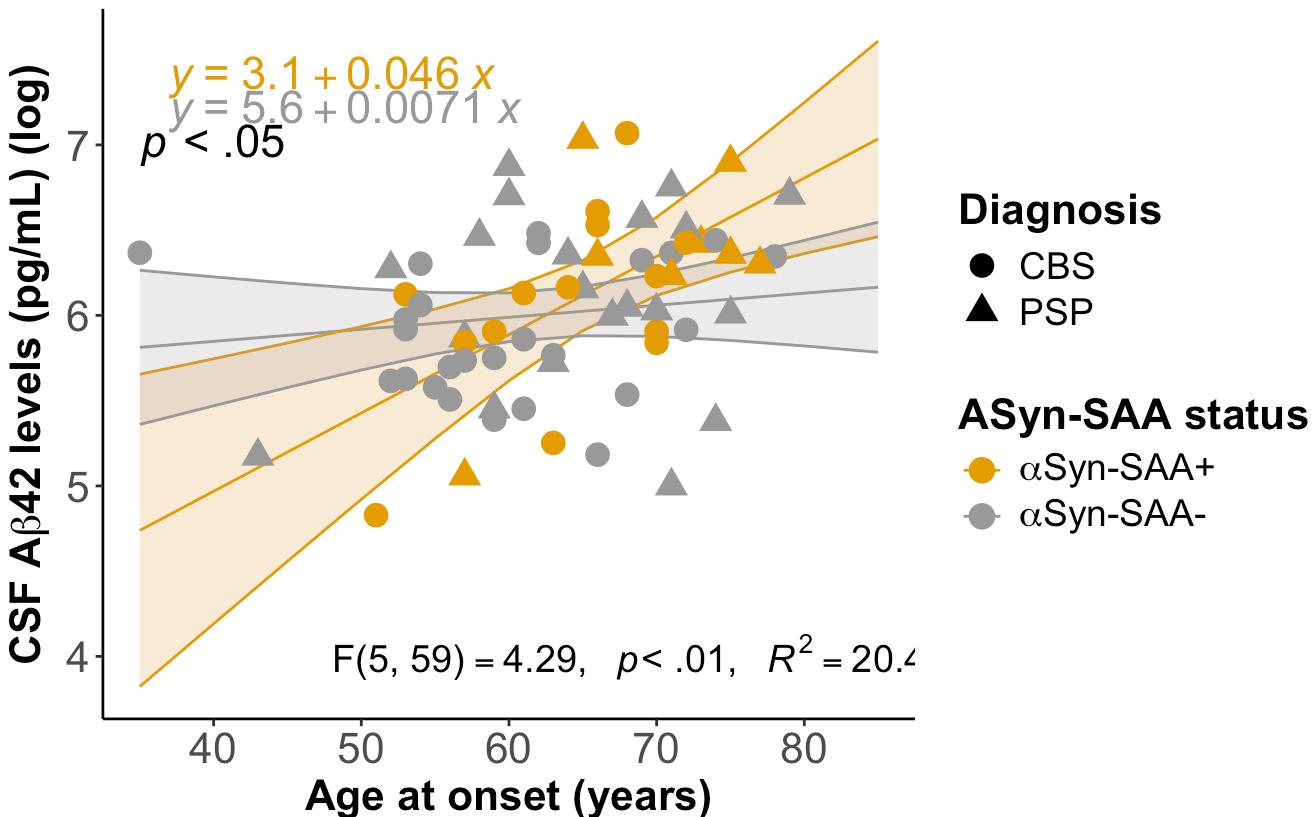
## Warning in summary.lm(res.lm): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(res.lm): essentially perfect fit: summary may be
## unreliable

```

presentation of the interaction of age at onset by α Syn-SAA

the model: $\log(A\beta 42) \sim \text{Age at onset} * \alpha\text{Syn-SAA} + \text{Diagnosis} + \text{NfL}$



```
ggsave(fig1b, filename = "Fig1b.png", bg= "transparent", width=9, height=10)
```

```
## Warning in summary.lm(res.lm): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(res.lm): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(res.lm): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(res.lm): essentially perfect fit: summary may be
## unreliable
```

8. ASYN-SAA+ & ALL VARIABLES

8.1. ASYN-SAA+ & NFL

8.1.1. NFL MLR

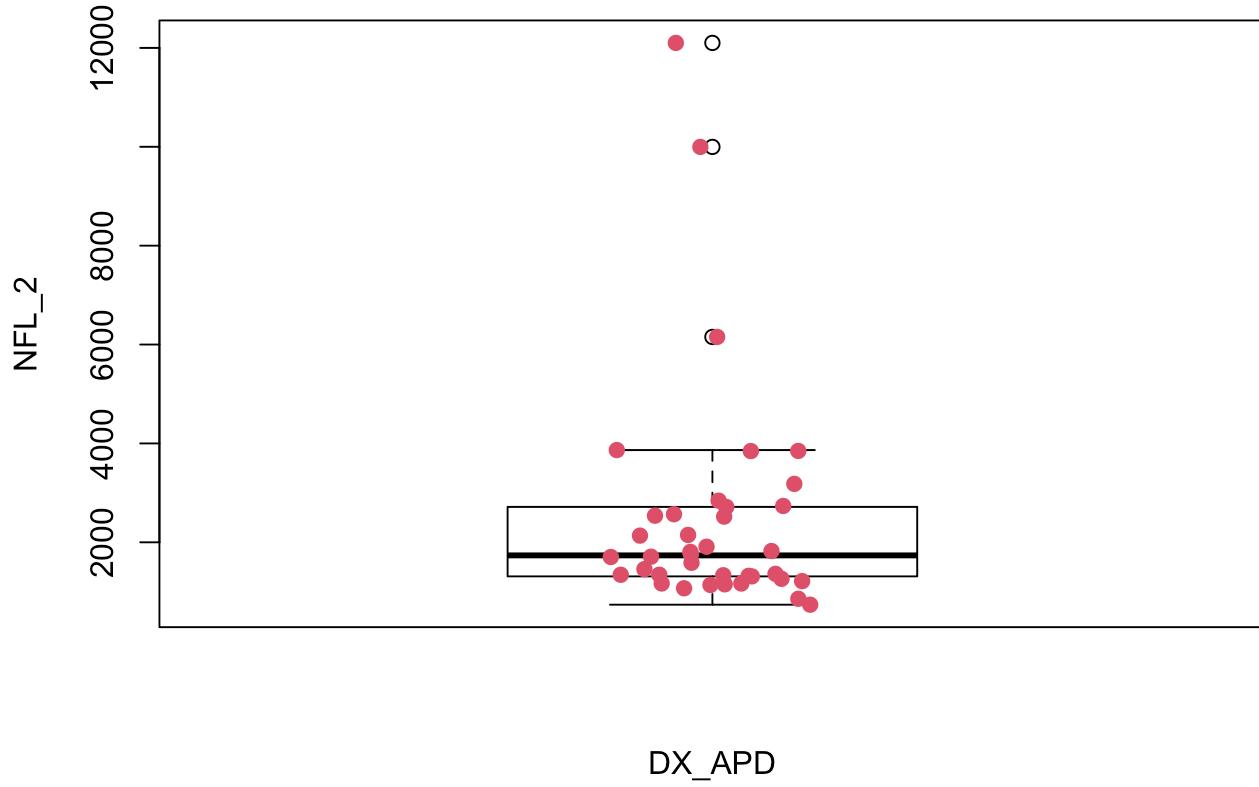
Comparisons shown in Results - ASYN-SAA+ & NFL

NFL STATISTICS: OUTLIERS

```
# Since the outcome of interest here is the extremely right skewed variable NfL, a tolerant approach to outlier removal was chosen ( $Q3 + 3 \times IQR$  instead of  $1.5 \times IQR$ . Reference for this is: https://www.nature.com/articles/s41598-020-66090-x).
# For reference, outliers are added to the notes of Tables.
boxplot(NFL_2 ~ DX_APD, data= CBSdf, col = "white")$out #identify outliers in each diagnosis. First, look at CBS: there is one so attribute its value to vector.
```

```
## [1] 12101.20 6153.82 9997.64
```

```
stripchart(NFL_2 ~ DX_APD, data = CBSdf, method = "jitter", pch = 19, col = 2:4, vertical = TRUE, add = TRUE)
```

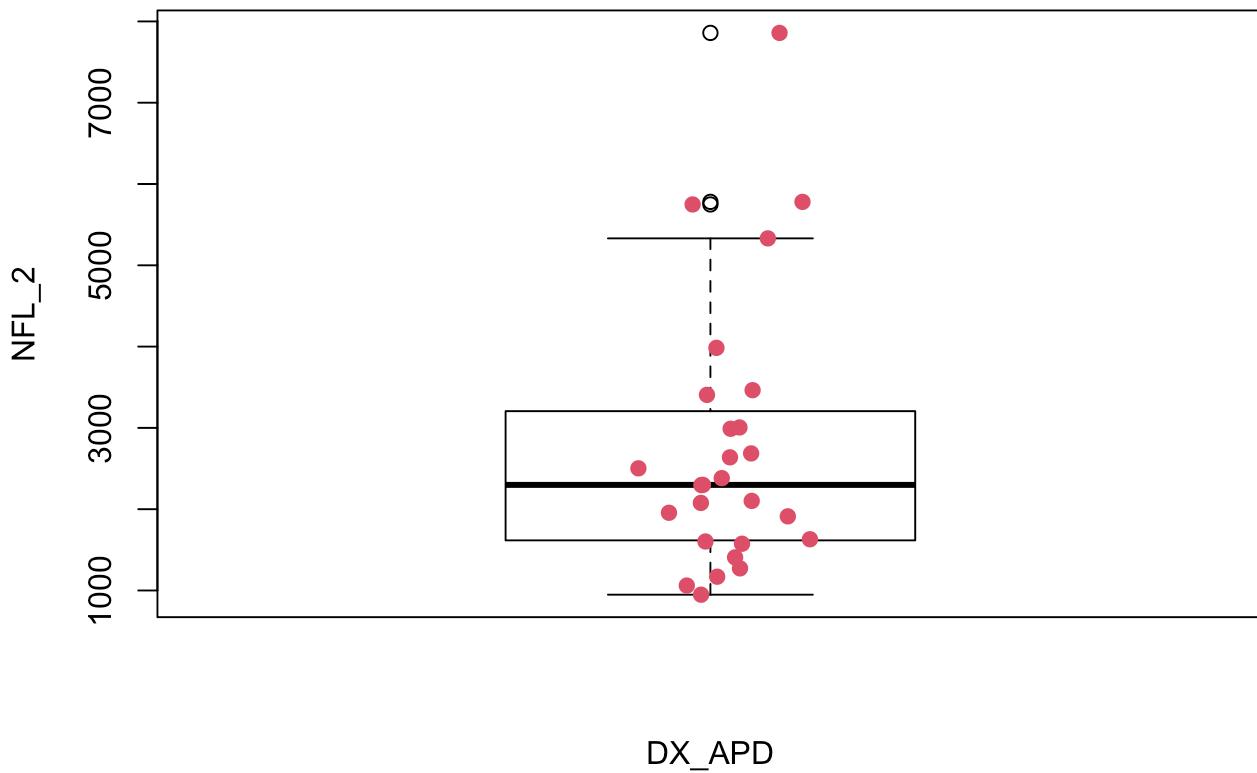


DX_APD

```
boxplot(NFL_2 ~ DX_APD, data= PSPdf, col = "white")$out #Now identify outliers in PSP: since there is none, no need to attribute to a vector.
```

```
## [1] 5780.13 5748.61 7858.61
```

```
stripchart(NFL_2 ~ DX_APD, data = PSPdf, method = "jitter", pch = 19, col = 2:4, vertical = TRUE, add = TRUE)
```



DX_APD

```
thresholdCBS <- min(max(CBSdf$NFL_2,na.rm=T), as.numeric(quantile(CBSdf$NFL_2, 0.75, na.rm=T)) + (IQR(na.rm=T, (CBSdf$NFL_2)*3))) #reports the value Q3+ IQR*3 (3 is very tolerant threshold)
thresholdPSP <- min(max(PSPdf$NFL_2,na.rm=T), as.numeric(quantile(PSPdf$NFL_2, 0.75, na.rm=T)) + (IQR(na.rm=T, (PSPdf$NFL_2)*3)))
cat("Outliers are values above ", thresholdCBS, " in CBS subset. \n")
```

```
## Outliers are values above 6775.568 in CBS subset.
```

```
cat("Outliers are values above ", thresholdPSP, " in PSP subset. \n")
```

```
## Outliers are values above 7858.61 in PSP subset.
```

```
dfnfl<- df[df$DX_APD=="PSP" | df$NFL_2 <= thresholdCBS, ] %>% remove_empty("rows") %>% data.frame() #Removes all subjects who are CBS and either have no NFL value or one over the threshold
dfnfl<- dfnfl[dfnfl$DX_APD=="CBS" | dfnfl$NFL_2 <= thresholdPSP, ] %>% remove_empty("rows") %>% data.frame() #Removes all subjects who are PSP and either have no NFL value or one over the threshold

removed <- setdiff(df, dfnfl)
cat("Following values were removed for the descriptive stats on NfL: ", removed$NFL_2,
"\n")
```

```
## Following values were removed for the descriptive stats on NfL: 12101.2 NA NA 9997.6
4
```

NFL STATISTICS: DISTRIBUTION

```
shapiro.test(dfnfl[dfnfl$DX_APD=="CBS", ]$logNFL) #normal
```

```
##
## Shapiro-Wilk normality test
##
## data: dfnfl[dfnfl$DX_APD == "CBS", ]$logNFL
## W = 0.96261, p-value = 0.2584
```

```
shapiro.test(dfnfl[dfnfl$DX_APD=="PSP", ]$logNFL) #normal
```

```
##
## Shapiro-Wilk normality test
##
## data: dfnfl[dfnfl$DX_APD == "PSP", ]$logNFL
## W = 0.97556, p-value = 0.7516
```

```
shapiro.test(dfnfl[dfnfl$RTQUIC=="aSyn-SAA positive", ]$logNFL) #borderline
```

```
##
## Shapiro-Wilk normality test
##
## data: dfnfl[dfnfl$RTQUIC == "aSyn-SAA positive", ]$logNFL
## W = 0.91836, p-value = 0.09216
```

```
shapiro.test(dfnfl[dfnfl$RTQUIC=="aSyn-SAA negative", ]$logNFL) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: dfnfl[dfnfl$RTQUIC == "aSyn-SAA negative", ]$logNFL  
## W = 0.97249, p-value = 0.3839
```

```
shapiro.test(dfnfl[dfnfl$AD=="AD Positive", ]$logNFL) #normal
```

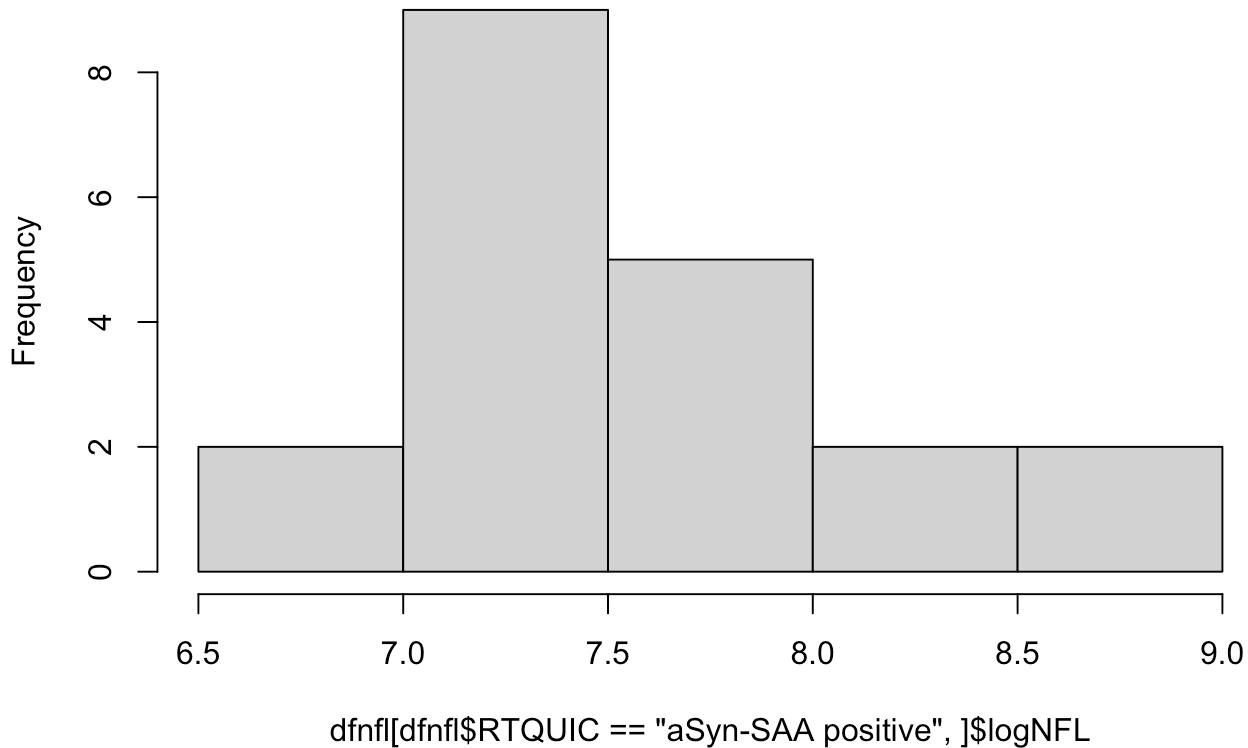
```
##  
## Shapiro-Wilk normality test  
##  
## data: dfnfl[dfnfl$AD == "AD Positive", ]$logNFL  
## W = 0.90972, p-value = 0.1561
```

```
shapiro.test(dfnfl[dfnfl$AD=="AD Negative", ]$logNFL) #borderline
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: dfnfl[dfnfl$AD == "AD Negative", ]$logNFL  
## W = 0.96128, p-value = 0.107
```

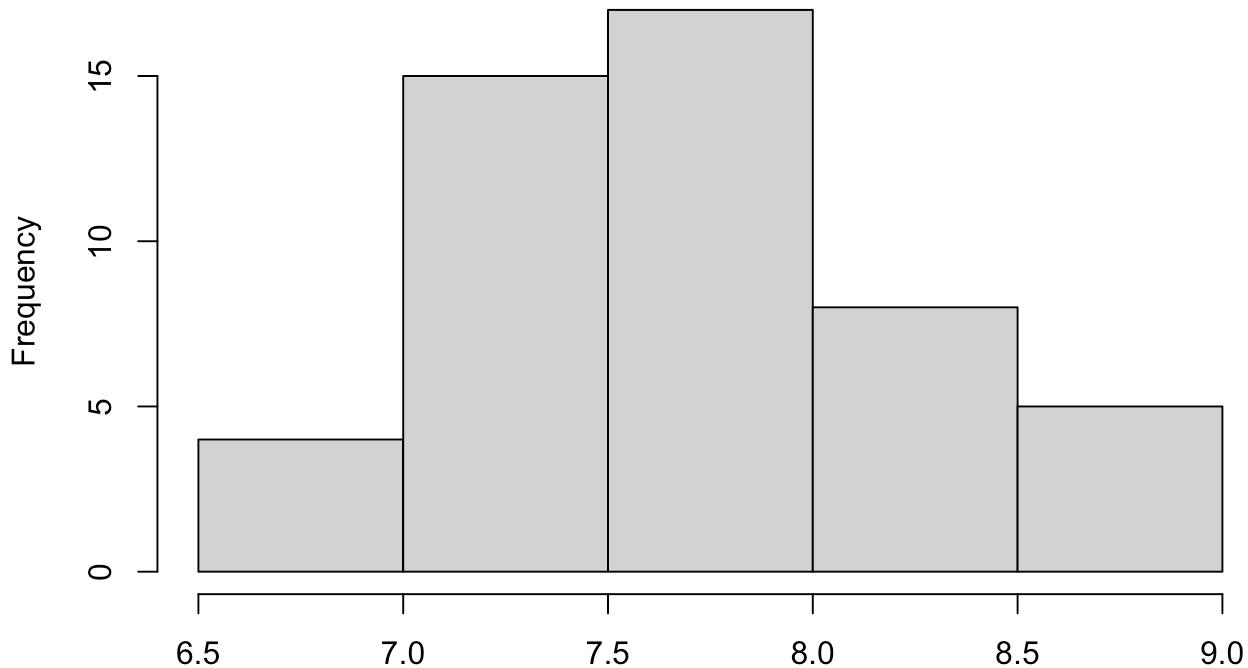
```
hist(dfnfl[dfnfl$RTQUIC=="aSyn-SAA positive", ]$logNFL)
```

Histogram of dfnfl[dfnfl\$RTQUIC == "aSyn-SAA positive",]\$logNFL



```
hist(dfnfl[dfnfl$AD=="AD Negative", ]$logNFL)
```

Histogram of dfnfl[dfnfl\$AD == "AD Negative",]\$logNFL



```
leveneTest(logNFL ~ DX_APD*AD*RTQUIC, data = dfnfl) #homoscedasticity
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  6  1.1766  0.332
##      56
```

NFL STATS: LINEAR REGRESSION MODEL SELECTION

```
# Compare models with Ftest
test1 <- lm(logNFL ~ RTQUIC, dfnfl)
test2 <- lm(logNFL ~ RTQUIC + DX_APD, dfnfl) #adding value to the model
test3 <- lm(logNFL ~ RTQUIC + AD, dfnfl) #not adding any value to the model
anova(test1, test2) #models that are nested in each other
```

```
## Analysis of Variance Table
##
## Model 1: logNFL ~ RTQUIC
## Model 2: logNFL ~ RTQUIC + DX_APPD
##   Res.Df   RSS Df Sum of Sq    F Pr(>F)
## 1     61 16.623
## 2     60 15.377  1      1.246 4.8619 0.0313 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(test1, test3)
```

```
## Analysis of Variance Table
##
## Model 1: logNFL ~ RTQUIC
## Model 2: logNFL ~ RTQUIC + AD
##   Res.Df   RSS Df Sum of Sq    F Pr(>F)
## 1     61 16.623
## 2     60 15.945  1      0.67755 2.5496 0.1156
```

Inclusion of other variables as covariate:
`summary(lm(dfnfl$logNFL ~ dfnfl$Age)) #no linear relationship`

```
##
## Call:
## lm(formula = dfnfl$logNFL ~ dfnfl$Age)
##
## Residuals:
##       Min     1Q     Median      3Q     Max 
## -0.96048 -0.42486 -0.04107  0.31145  1.23321 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 7.063196  0.482090 14.65   <2e-16 ***
## dfnfl$Age   0.008069  0.006956  1.16    0.251    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
##
## Residual standard error: 0.5178 on 61 degrees of freedom
## Multiple R-squared:  0.02158,    Adjusted R-squared:  0.005544 
## F-statistic: 1.346 on 1 and 61 DF,  p-value: 0.2506
```

`summary(lm(dfnfl$logNFL ~ dfnfl$LP2_Disease_Duration))#no linear relationship`

```

## 
## Call:
## lm(formula = dfnfl$logNFL ~ dfnfl$LP2_Disease_Duration)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -1.00926 -0.41465 -0.05653  0.28669  1.35379
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             7.60744   0.12119  62.776 <2e-16 ***
## dfnfl$LP2_Disease_Duration 0.00185   0.01912   0.097   0.923  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5234 on 61 degrees of freedom
## Multiple R-squared:  0.0001535, Adjusted R-squared:  -0.01624 
## F-statistic: 0.009365 on 1 and 61 DF,  p-value: 0.9232

```

```
summary(lm(dfnfl$logNFL ~ dfnfl$abeta_2)) #linear relationship
```

```

## 
## Call:
## lm(formula = dfnfl$logNFL ~ dfnfl$abeta_2)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -0.89492 -0.34549 -0.01374  0.21666  1.20997
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)    7.2573600  0.1468411 49.423 <2e-16 ***
## dfnfl$abeta_2 0.0007680  0.0002837   2.707   0.0088 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4946 on 61 degrees of freedom
## Multiple R-squared:  0.1072, Adjusted R-squared:  0.0926  
## F-statistic: 7.327 on 1 and 61 DF,  p-value: 0.008797

```

```
summary(lm(dfnfl$logNFL ~ dfnfl$ptau_2)) #no linear relationship
```

```

## 
## Call:
## lm(formula = dfnfl$logNFL ~ dfnfl$ptau_2)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -0.97886 -0.41905 -0.03975  0.29057  1.34893
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 7.713347  0.132485 58.220 <2e-16 ***
## dfnfl$ptau_2 -0.001804  0.002162 -0.835   0.407    
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5205 on 61 degrees of freedom
## Multiple R-squared:  0.01129, Adjusted R-squared: -0.004921
## F-statistic: 0.6964 on 1 and 61 DF, p-value: 0.4073

```

```
summary(lm(dfnfl$logNFL ~ dfnfl$ttau_2)) #no linear relationship
```

```

## 
## Call:
## lm(formula = dfnfl$logNFL ~ dfnfl$ttau_2)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -1.01292 -0.40270 -0.02267  0.29122  1.34669
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 7.6657704  0.1084848 70.662 <2e-16 ***
## dfnfl$ttau_2 -0.0001497  0.0002192 -0.683   0.497    
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5191 on 60 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.007718, Adjusted R-squared: -0.00882
## F-statistic: 0.4667 on 1 and 60 DF, p-value: 0.4972

```

NFL STATISTICS: LINEAR REGRESSION MODEL

```

# Based on clear linear relationship between abeta and NfL as seen above, it doesn't make sense to exclude Abeta from the model.
# Meanwhile, dx is an important covariate, and RTQUIC is the IV of interest. So just comparing two models: with or without interaction.
stdmlr <- lm(logNFL ~ RTQUIC*DX_APD + scale(abeta_2), dfnfl)
summary(stdmlr)

```

```

## 
## Call:
## lm(formula = logNFL ~ RTQUIC * DX_APD + scale(abeta_2), data = dfnfl)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -0.91541 -0.27896 -0.06221  0.27836  1.14385
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                7.50151   0.10223  73.380 <2e-16 ***
## RTQUICaSyn-SAA positive    0.08831   0.16702   0.529  0.5990
## DX_APDPSP                  0.33528   0.15024   2.232  0.0295 *
## scale(abeta_2)              0.16122   0.06517   2.474  0.0163 *
## RTQUICaSyn-SAA positive:DX_APDPSP -0.50362   0.27044  -1.862  0.0676 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4802 on 58 degrees of freedom
## Multiple R-squared:  0.1997, Adjusted R-squared:  0.1445
## F-statistic: 3.618 on 4 and 58 DF,  p-value: 0.01063

```

AIC(stdmlr) #Slightly better AIC

```
## [1] 93.15681
```

```
stdmlr2 <- lm(logNFL ~ RTQUIC + DX_APD + scale(abeta_2), dfnfl)
summary(stdmlr2)
```

```

## 
## Call:
## lm(formula = logNFL ~ RTQUIC + DX_APD + scale(abeta_2), data = dfnfl)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -0.90290 -0.33584 -0.09789  0.23158  1.26717
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                7.56632   0.09811  77.120 <2e-16 ***
## RTQUICaSyn-SAA positive   -0.10055   0.13545  -0.742  0.4608
## DX_APDPSP                  0.19339   0.13217   1.463  0.1487
## scale(abeta_2)              0.14780   0.06611   2.236  0.0292 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4902 on 59 degrees of freedom
## Multiple R-squared:  0.1518, Adjusted R-squared:  0.1087
## F-statistic: 3.521 on 3 and 59 DF,  p-value: 0.02037

```

```
AIC(stdmlr2)
```

```
## [1] 94.81533
```

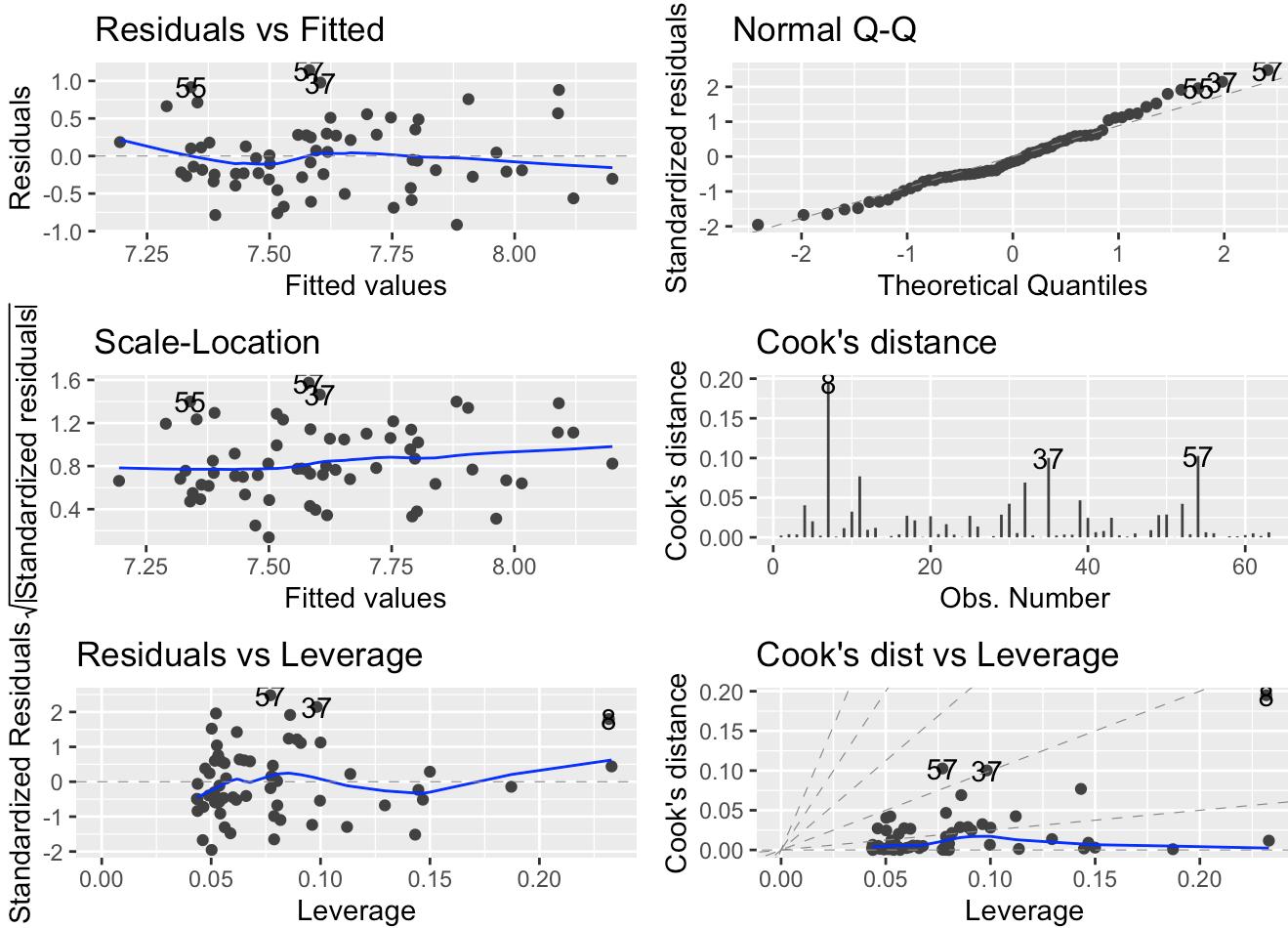
AIC is slightly better when including the interaction term between DX and RTQUIC, so we will keep it as more informative.
Moreover, it seems in model without interaction, the main effect of diagnosis is not significant. Suggests that not including the interaction term
would be misleading.

NFL STATISTICS: LINEAR REGRESSION MODEL DIAGNOSTICS

```
check_normality(stdmlr) #OK normality of residuals
```

```
## OK: residuals appear as normally distributed (p = 0.315).
```

```
autoplot(stdmlr, which = 1:6) #All plots including scale location plot
```



Visualize the values for each of the IDs that are indexed on above plots
dfnfl[8, c("abeta_2", "logabeta", "RTQUIC", "DX_APD", "NFL_2")]

```
##   abeta_2 logabeta          RTQUIC DX_APD   NFL_2
## 9 990.464 6.898174 aSyn-SAA positive    PSP 2296.71
```

```
dfnfl[37, c("abeta_2", "logabeta", "RTQUIC", "DX_APD","NFL_2")]
```

```
##   abeta_2 logabeta          RTQUIC DX_APD   NFL_2
## 39 233.053 5.451266 aSyn-SAA negative    PSP 2638.08
```

```
dfnfl[55, c("abeta_2", "logabeta", "RTQUIC", "DX_APD","NFL_2")]
```

```
##   abeta_2 logabeta          RTQUIC DX_APD   NFL_2
## 59 370.95 5.916067 aSyn-SAA negative    CBS 1137.16
```

```
dfnfl[57, c("abeta_2", "logabeta", "RTQUIC", "DX_APD","NFL_2")]
```

```
##   abeta_2 logabeta          RTQUIC DX_APD   NFL_2
## 62 428.65 6.060641 aSyn-SAA negative    CBS 1708.68
```

```
# for-loop to test the model without each of these values (ie once without outlier 1, then outlier 2, etc)
vecIDs <- dfnfl[c(8, 37, 55, 57), "ID"] #Create vector of each ID. For 1.5*IQR outlier threshold
for (i in vecIDs) {
  test <- subset(dfnfl, ID!=i) #for some analysis, need to exclude the potential false negative too
  teststdmlr <- lm(logNFL ~ RTQUIC*DX_APD + scale(abeta_2), test)
  print(summary(teststdmlr))
}
```

```

## 
## Call:
## lm(formula = logNFL ~ RTQUIC * DX_APD + scale(abeta_2), data = test)
## 
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -0.91565 -0.28194 -0.06793  0.27820  1.14366 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                7.49599   0.10272  72.977 <2e-16 ***
## RTQUICaSyn-SAA positive    0.08793   0.16847   0.522   0.6037    
## DX_APDPSP                  0.33421   0.15171   2.203   0.0317 *  
## scale(abeta_2)              0.15706   0.06487   2.421   0.0187 *  
## RTQUICaSyn-SAA positive:DX_APDPSP -0.49369   0.28152  -1.754   0.0849 .  
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.4843 on 57 degrees of freedom
## Multiple R-squared:  0.1993, Adjusted R-squared:  0.1431 
## F-statistic: 3.546 on 4 and 57 DF,  p-value: 0.01186 
## 
## 
## Call:
## lm(formula = logNFL ~ RTQUIC * DX_APD + scale(abeta_2), data = test)
## 
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -0.90445 -0.28094 -0.07794  0.27607  1.14340 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                7.50603   0.10323  72.708 <2e-16 *** 
## RTQUICaSyn-SAA positive    0.08740   0.16818   0.520   0.6053    
## DX_APDPSP                  0.32118   0.15438   2.081   0.0420 *  
## scale(abeta_2)              0.16627   0.06653   2.499   0.0154 *  
## RTQUICaSyn-SAA positive:DX_APDPSP -0.49450   0.27303  -1.811   0.0754 .  
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.4835 on 57 degrees of freedom
## Multiple R-squared:  0.1993, Adjusted R-squared:  0.1431 
## F-statistic: 3.547 on 4 and 57 DF,  p-value: 0.01184 
## 
## 
## Call:
## lm(formula = logNFL ~ RTQUIC * DX_APD + scale(abeta_2), data = test)
## 
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -0.9153  -0.2837 -0.0564  0.2625  1.1439 
## 
```

```

## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                7.52027   0.10472  71.811 <2e-16 ***
## RTQUICaSyn-SAA positive    0.07056   0.16879   0.418   0.6775
## DX_APDPSP                  0.31783   0.15207   2.090   0.0411 *
## scale(abeta_2)              0.16131   0.06578   2.452   0.0173 *
## RTQUICaSyn-SAA positive:DX_APDPSP -0.48526   0.27202  -1.784   0.0798 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4815 on 57 degrees of freedom
## Multiple R-squared:  0.1928, Adjusted R-squared:  0.1362
## F-statistic: 3.405 on 4 and 57 DF,  p-value: 0.01448
##
##
## Call:
## lm(formula = logNFL ~ RTQUIC * DX_APD + scale(abeta_2), data = test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.91542 -0.28162 -0.07366  0.27835  1.14385
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                7.50332   0.10546  71.147 <2e-16 ***
## RTQUICaSyn-SAA positive    0.08699   0.16987   0.512   0.6106
## DX_APDPSP                  0.33393   0.15316   2.180   0.0334 *
## scale(abeta_2)              0.16257   0.06627   2.453   0.0172 *
## RTQUICaSyn-SAA positive:DX_APDPSP -0.50234   0.27360  -1.836   0.0716 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4844 on 57 degrees of freedom
## Multiple R-squared:  0.1983, Adjusted R-squared:  0.142
## F-statistic: 3.524 on 4 and 57 DF,  p-value: 0.01223

```

bptest(stdmlr) #Ok variance of residulas. Breusch-Pagan test for heterodasticity.

```

##
## studentized Breusch-Pagan test
##
## data: stdmlr
## BP = 1.5504, df = 4, p-value = 0.8177

```

durbinWatsonTest(stdmlr) #Ok autocorrelation of residuals

```

## lag Autocorrelation D-W Statistic p-value
## 1     0.02629406     1.933946   0.724
## Alternative hypothesis: rho != 0

```

```
# Multicollinearity checks
car::vif(stdmlr) #no multicollinearity at all
```

```
## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif
```

```
##          RTQUIC      DX_APD scale(abeta_2)  RTQUIC:DX_APD
## 1.651222    1.510114     1.141793     1.973264
```

NFL STATISTICS: LINEAR REGRESSION MODEL MAIN EFFECTS

```
emmeans(stdmlr, ~ RTQUIC:DX_APD) #adjusted means.
```

```
##   RTQUIC      DX_APD emmean    SE df lower.CL upper.CL
## aSyn-SAA negative CBS    7.50 0.102 58     7.30     7.71
## aSyn-SAA positive CBS   7.59 0.134 58     7.32     7.86
## aSyn-SAA negative PSP   7.84 0.108 58     7.62     8.05
## aSyn-SAA positive PSP   7.42 0.189 58     7.04     7.80
##
## Confidence level used: 0.95
```

```
pairs(emmeans(stdmlr, ~ RTQUIC:DX_APD)) #adjusted means.
```

```
## contrast                                estimate    SE df t.ratio
## (aSyn-SAA negative CBS) - (aSyn-SAA positive CBS) -0.0883 0.167 58 -0.529
## (aSyn-SAA negative CBS) - (aSyn-SAA negative PSP) -0.3353 0.150 58 -2.232
## (aSyn-SAA negative CBS) - (aSyn-SAA positive PSP)  0.0800 0.220 58  0.364
## (aSyn-SAA positive CBS) - (aSyn-SAA negative PSP) -0.2470 0.172 58 -1.433
## (aSyn-SAA positive CBS) - (aSyn-SAA positive PSP)  0.1683 0.233 58  0.721
## (aSyn-SAA negative PSP) - (aSyn-SAA positive PSP)  0.4153 0.215 58  1.933
##
## p.value
## 0.9518
## 0.1268
## 0.9833
## 0.4841
## 0.8883
## 0.2259
##
## P value adjustment: tukey method for comparing a family of 4 estimates
```

8.1.2. FIGURE 1.D.

Figure 1.d.

FIG1D: NFL: Different between DX, but not RTQUIC, and linearly related to Abeta42. - Option 1: Plot Abeta42 by NFL relationship and add colors for DX and shapes for RTQUIC. This is done here but not in paper. - Option 2: Plot boxplot between RTQUIC status (see below this figure)

OPTION 1

```
# Create dataframe usable for plotting (ie kick out the scale())
plotmlr <- lm(logNFL ~ RTQUIC*DX_APD + abeta_2, dfnfl)
summary(plotmlr)
```

```
##
## Call:
## lm(formula = logNFL ~ RTQUIC * DX_APD + abeta_2, data = dfnfl)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -0.91541 -0.27896 -0.06221  0.27836  1.14385
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             7.1602333  0.1542851 46.409   <2e-16 ***
## RTQUICaSyn-SAA positive  0.0883131  0.1670216  0.529   0.5990    
## DX_APDPSP                0.3352832  0.1502418  2.232   0.0295 *  
## abeta_2                  0.0007282  0.0002944  2.474   0.0163 *  
## RTQUICaSyn-SAA positive:DX_APDPSP -0.5036172  0.2704387 -1.862   0.0676 .  
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4802 on 58 degrees of freedom
## Multiple R-squared:  0.1997, Adjusted R-squared:  0.1445 
## F-statistic: 3.618 on 4 and 58 DF,  p-value: 0.01063
```

```

label <- "paste('*italic(p), \" < .05\"') #Second annotation is p-value for the interaction

# General layout of the plot: x and y + stats_smooth for linear relationship
fig1d_ver1 <- ggplot(dfnfl, aes(x=abeta_2, y=logNFL)) + #No need for color or fill

  # Add actual datapoints
  geom_point(aes(color=RTQUIC, shape=DX_APD), size=4) +
  stat_smooth(aes(), method="lm", color="red", fill="red", linewidth=0.5, alpha=0.2, level = 0.95) + #No need to do separate lines for RTQUIC diagnosis. Lines are very similar and CIs encompass each other. Tried with geom_smooth too.

  # Fix legends and set up the appearance of the points
  scale_color_manual(values=cbPalette_RTQUIC, name="ASyn-SAA status", breaks=c("aSyn-SAA positive", "aSyn-SAA negative"), labels=c(expression(alpha*Syn-SAA+"), expression(alpha*Syn-SAA-")))) + #expression allows you to add greek letters
  scale_shape_manual(values=c(16,17), name="Diagnosis", breaks=c("CBS", "PSP"), labels=c("CBS", "PSP")) + #Color for the datapoints

  # Fix labs
  labs(title=expression(bold("Linear relationship between *Alpha*beta*42 and NfL")), subtitle=""),
  x=expression(bold("CSF A"*beta*"42 levels (pg/mL)")),
  y=expression(bold("CSF NfL levels (pg/mL) (log)")) + #bold() is required otherwise the Y axis will not be bold in spite of element_text specification below

  # Annotate:
  stat_regline_equation(label.x=120, label.y=8.7, color="red", size=6, show.legend=FALSE) + #shows the equation for each geom_line. Cannot use other options since it would not be based on the model

  annotate("text", size=6, x=175, y=8.5, color="red", label=label, parse=TRUE)+ #label as defined above. Parse allows for use of mathematical notation

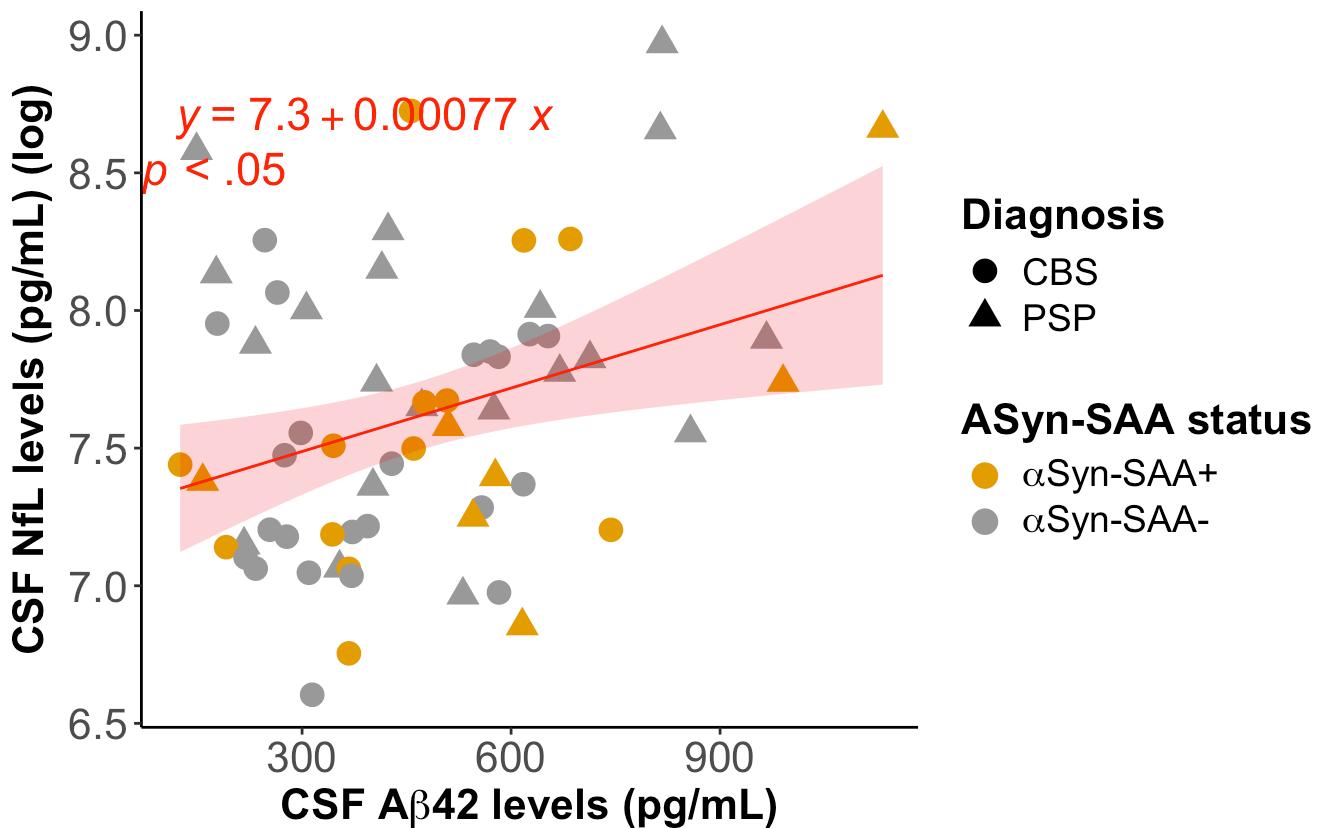
  # Aesthetic only
  theme_classic() +
  theme(plot.title = element_text(size=16, hjust=0.5, face="bold")) +
  theme(axis.text=element_text(size=16), axis.title=element_text(size=16, face="bold"))
+
  theme(legend.title = element_text(face="bold", size=16), legend.text= element_text(size=14))

  fig1d_ver1

```

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

Linear relationship between A β 42 and NfL



```
ggsave(fig1d_ver1, filename = "Fig1d_ver1.png", bg= "transparent", width=9, height=10)
```

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

OPTION 2

```

# General layout of the plot: boxplot by diagnosis where diagnosis is sig, within diagnosis rtquic groups are not sig
fig1d_ver2 <- ggplot(dfnfl, aes(x=DX_APD, y=logNFL, color=RTQUIC))+ #No need for color or fill

# Add actual datapoints
geom_boxplot() +
  geom_jitter(aes(color=RTQUIC), position=position_jitterdodge()) +

# Fix legends and set up the appearance of the points
  scale_color_manual(values=cbPalette_RTQUIC, name="ASyn-SAA status", breaks=c("aSyn-SAA positive", "aSyn-SAA negative"), labels=c(expression(alpha*Syn-SAA+"), expression(alpha*Syn-SAA-")))+ #expression allows you to add greek letters
  scale_fill_manual(values=cbPalette_RTQUIC, name="ASyn-SAA status", breaks=c("aSyn-SAA positive", "aSyn-SAA negative"), labels=c(expression(alpha*Syn-SAA+"), expression(alpha*Syn-SAA-")))+ #expression allows you to add greek letters
  scale_shape_manual(values=c(16,17), name="Diagnosis", breaks=c("CBS", "PSP"), labels=c("CBS", "PSP"))+ #Color for the datapoints

# Fix labs
  labs(title=expression("NFL levels by diagnosis and *alpha*Syn-SAA status"),
       subtitle="",
       y=expression(bold("CSF NFL levels (pg/mL) (log)")))+ #bold() is required otherwise the Y axis will not be bold in spite of element_text specification below

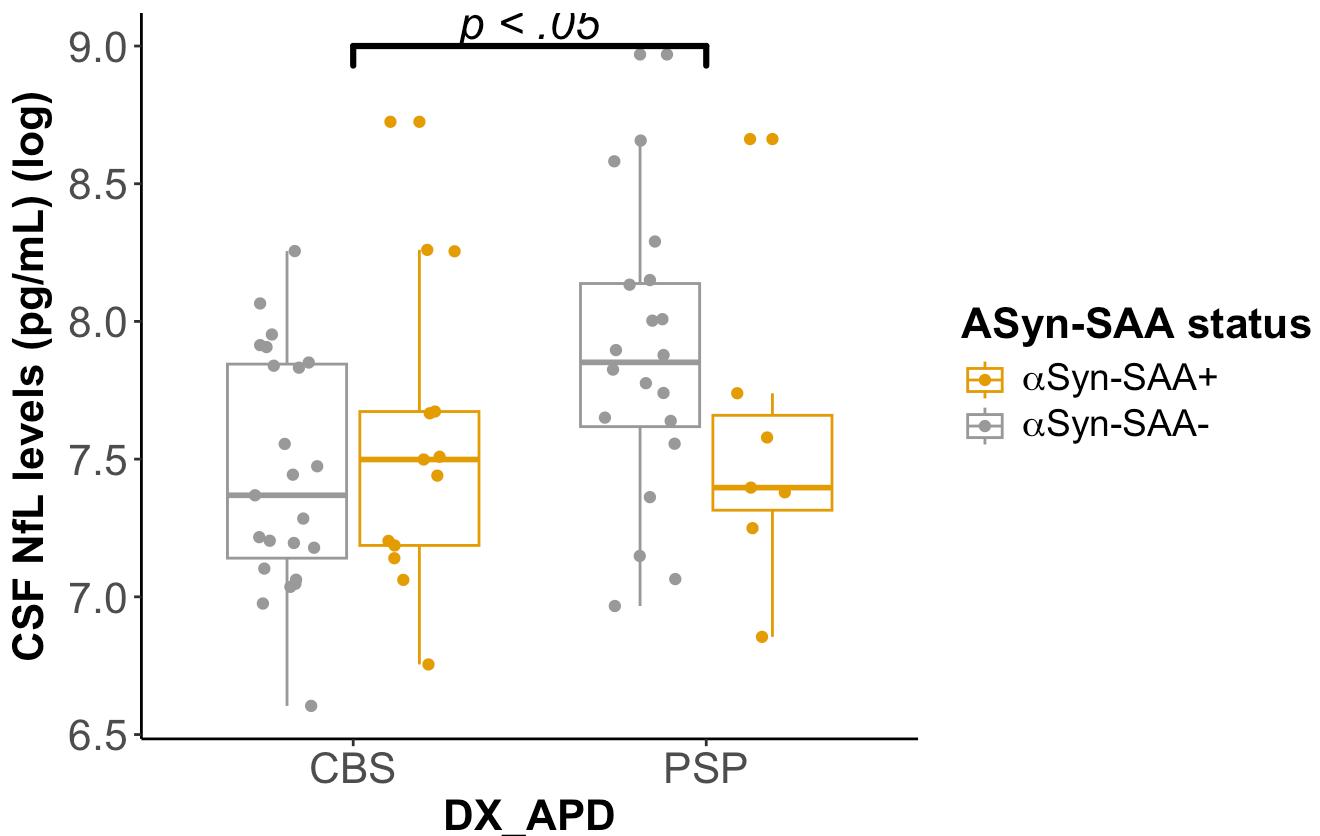
# Annotate:
  geom_signif(y_position = c(9), xmin = c(1), xmax = c(2), annotation = c("p < .05"),
  fontface = "italic",
  col="black", size=1.1, textsize=6) +

#Aesthetic only #Needs to be after Annotate() otherwise messes with the size of the asterisk
  theme_classic() +
  theme(plot.title = element_text(size=16, hjust=0.5, face="bold")) +
  theme(axis.text=element_text(size=16), axis.title=element_text(size=16, face="bold")) +
  theme(legend.title = element_text(face="bold", size=16), legend.text= element_text(size=14))

fig1d_ver2

```

NfL levels by diagnosis and α Syn-SAA status



```
ggsave(fig1d_ver2, filename = "Fig1d_ver2.png", bg= "transparent", width=9, height=10)
```

8.2. ASYN-SAA+ & SYMPTOMS

8.2.1. CBS-ONLY NUMERICAL VARIABLES

Comparisons shown in: eTable 1

CBS ONLY STATISTICS: DISTRIBUTION

```
shapiro.test(CBSdf[CBSdf$RTQUIC=="aSyn-SAA positive", ]$Age) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: CBSdf[CBSdf$RTQUIC == "aSyn-SAA positive", ]$Age  
## W = 0.95004, p-value = 0.5613
```

```
shapiro.test(CBSdf[CBSdf$RTQUIC=="aSyn-SAA negative", ]$Age) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: CBSdf[CBSdf$RTQUIC == "aSyn-SAA negative", ]$Age  
## W = 0.96496, p-value = 0.5217
```

```
var.test(Age ~ RTQUIC, data = CBSdf) #homoscedasticity
```

```
##  
## F test to compare two variances  
##  
## data: Age by RTQUIC  
## F = 1.3081, num df = 24, denom df = 13, p-value = 0.626  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.452126 3.272749  
## sample estimates:  
## ratio of variances  
## 1.308087
```

```
shapiro.test(CBSdf[CBSdf$RTQUIC=="aSyn-SAA positive", ]$Onset_age) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: CBSdf[CBSdf$RTQUIC == "aSyn-SAA positive", ]$Onset_age  
## W = 0.93077, p-value = 0.3128
```

```
shapiro.test(CBSdf[CBSdf$RTQUIC=="aSyn-SAA negative", ]$Onset_age) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: CBSdf[CBSdf$RTQUIC == "aSyn-SAA negative", ]$Onset_age  
## W = 0.94595, p-value = 0.2028
```

```
var.test(Onset_age ~ RTQUIC, data = CBSdf) #homoscedasticity
```

```

## 
## F test to compare two variances
##
## data: Onset_age by RTQUIC
## F = 1.8716, num df = 24, denom df = 13, p-value = 0.2403
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.6469066 4.6826837
## sample estimates:
## ratio of variances
## 1.871625

```

```
shapiro.test(CBSdf[CBSdf$RTQUIC=="aSyn-SAA positive", ]$Park_onset) #normal
```

```

## 
## Shapiro-Wilk normality test
##
## data: CBSdf[CBSdf$RTQUIC == "aSyn-SAA positive", ]$Park_onset
## W = 0.93165, p-value = 0.3218

```

```
shapiro.test(CBSdf[CBSdf$RTQUIC=="aSyn-SAA negative", ]$Park_onset) #normal
```

```

## 
## Shapiro-Wilk normality test
##
## data: CBSdf[CBSdf$RTQUIC == "aSyn-SAA negative", ]$Park_onset
## W = 0.94546, p-value = 0.2156

```

```
var.test(Park_onset ~ RTQUIC, data = CBSdf) #homoscedasticity
```

```

## 
## F test to compare two variances
##
## data: Park_onset by RTQUIC
## F = 1.9243, num df = 23, denom df = 13, p-value = 0.222
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.6623739 4.8700231
## sample estimates:
## ratio of variances
## 1.924299

```

```
shapiro.test(CBSdf[CBSdf$RTQUIC=="aSyn-SAA positive", ]$Age) #normal
```

```
## 
## Shapiro-Wilk normality test
##
## data: CBSdf[CBSdf$RTQUIC == "aSyn-SAA positive", ]$Age
## W = 0.95004, p-value = 0.5613
```

```
shapiro.test(CBSdf[CBSdf$RTQUIC=="aSyn-SAA negative", ]$Age) #normal
```

```
## 
## Shapiro-Wilk normality test
##
## data: CBSdf[CBSdf$RTQUIC == "aSyn-SAA negative", ]$Age
## W = 0.96496, p-value = 0.5217
```

```
var.test(Age ~ RTQUIC, data = CBSdf) #homoscedasticity
```

```
## 
## F test to compare two variances
##
## data: Age by RTQUIC
## F = 1.3081, num df = 24, denom df = 13, p-value = 0.626
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.452126 3.272749
## sample estimates:
## ratio of variances
## 1.308087
```

CBS ONLY STATISTICS: SUMMARY

```
CBSdf %>% group_by(RTQUIC) %>% summarize(format(round(mean(Age, na.rm=T),2),2), sd=sd(Age, na.rm=T))
```

```
## # A tibble: 2 × 3
##   RTQUIC           `format(round(mean(Age, na.rm = T), 2), 2)`     sd
##   <fct>             <chr>                                         <dbl>
## 1 aSyn-SAA negative 64.74                                         8.92
## 2 aSyn-SAA positive  68.02                                         7.80
```

```
CBSdf %>% group_by(RTQUIC) %>% summarize(format(round(mean(Onset_age, na.rm=T),2),2), sd=sd(Onset_age, na.rm=T))
```

```
## # A tibble: 2 × 3
##   RTQUIC      `format(round(mean(Onset_age, na.rm = T), 2), 2)`    sd
##   <fct>        <chr>
## 1 aSyn-SAA negative 60.12          9.04
## 2 aSyn-SAA positive  63.57          6.61
```

```
CBSdf %>% group_by(RTQUIC) %>% summarize(format(round(mean(Park_onset, na.rm=T),2),2), s=sd(Park_onset, na.rm=T))
```

```
## # A tibble: 2 × 3
##   RTQUIC      `format(round(mean(Park_onset, na.rm = T), 2), 2)`    sd
##   <fct>        <chr>
## 1 aSyn-SAA negative 61.75          8.94
## 2 aSyn-SAA positive  65.57          6.44
```

CBS ONLY STATISTICS: T-TEST

```
t.test(CBSdf$Age ~ CBSdf$RTQUIC, var.equal=TRUE)
```

```
##
## Two Sample t-test
##
## data: CBSdf$Age by CBSdf$RTQUIC
## t = -1.1501, df = 37, p-value = 0.2575
## alternative hypothesis: true difference in means between group aSyn-SAA negative and
## group aSyn-SAA positive is not equal to 0
## 95 percent confidence interval:
## -9.062894 2.499684
## sample estimates:
## mean in group aSyn-SAA negative mean in group aSyn-SAA positive
## 64.73973 68.02133
```

```
t.test(CBSdf$Onset_age ~ CBSdf$RTQUIC, var.equal=TRUE)
```

```
##
## Two Sample t-test
##
## data: CBSdf$Onset_age by CBSdf$RTQUIC
## t = -1.2509, df = 37, p-value = 0.2188
## alternative hypothesis: true difference in means between group aSyn-SAA negative and
## group aSyn-SAA positive is not equal to 0
## 95 percent confidence interval:
## -9.042210 2.139353
## sample estimates:
## mean in group aSyn-SAA negative mean in group aSyn-SAA positive
## 60.12000 63.57143
```

```
t.test(CBSdf$Park_onset ~ CBSdf$RTQUIC, var.equal=TRUE)
```

```
##
## Two Sample t-test
##
## data: CBSdf$Park_onset by CBSdf$RTQUIC
## t = -1.3987, df = 36, p-value = 0.1704
## alternative hypothesis: true difference in means between group aSyn-SAA negative and
## group aSyn-SAA positive is not equal to 0
## 95 percent confidence interval:
## -9.362269 1.719412
## sample estimates:
## mean in group aSyn-SAA negative mean in group aSyn-SAA positive
## 61.75000 65.57143
```

8.2.2. CBS-ONLY CATEGORICAL VARIABLES

Comparisons shown in: eTable 1

CBS ONLY STATISTICS: SUMMARY

```
CBSdf %>% group_by(RTQUIC) %>% count(Sex)
```

```
## # A tibble: 4 × 3
## # Groups: RTQUIC [2]
##   RTQUIC       Sex     n
##   <fct>      <chr> <int>
## 1 aSyn-SAA negative F     10
## 2 aSyn-SAA negative M     15
## 3 aSyn-SAA positive F      9
## 4 aSyn-SAA positive M      5
```

```
CBSdf %>% group_by(RTQUIC) %>% count(AP0Ee4)
```

```
## # A tibble: 6 × 3
## # Groups: RTQUIC [2]
##   RTQUIC       AP0Ee4     n
##   <fct>      <fct> <int>
## 1 aSyn-SAA negative Negative  18
## 2 aSyn-SAA negative Positive   6
## 3 aSyn-SAA negative <NA>      1
## 4 aSyn-SAA positive Negative  12
## 5 aSyn-SAA positive Positive   1
## 6 aSyn-SAA positive <NA>      1
```

```
CBSdf %>% group_by(RTQUIC) %>% count(Parkinsonian_onset)
```

```
## # A tibble: 4 × 3
## # Groups: RTQUIC [2]
##   RTQUIC          Parkinsonian_onset     n
##   <fct>            <chr>                <int>
## 1 aSyn-SAA negative No             17
## 2 aSyn-SAA negative Yes            8
## 3 aSyn-SAA positive No            10
## 4 aSyn-SAA positive Yes           4
```

CBSdf %>% group_by(RTQUIC) %>% count(Tremor_binary)

```
## # A tibble: 4 × 3
## # Groups: RTQUIC [2]
##   RTQUIC          Tremor_binary     n
##   <fct>            <chr>                <int>
## 1 aSyn-SAA negative No             15
## 2 aSyn-SAA negative Yes            10
## 3 aSyn-SAA positive No            10
## 4 aSyn-SAA positive Yes           4
```

CBSdf %>% group_by(RTQUIC) %>% count(RestTremor) *#Very rare symptom (<3 in both groups) so no stats comparison within CBS (ie CBS-aSyn+ vs CBS-aSyn-).*

```
## # A tibble: 4 × 3
## # Groups: RTQUIC [2]
##   RTQUIC          RestTremor     n
##   <fct>            <chr>                <int>
## 1 aSyn-SAA negative No             24
## 2 aSyn-SAA negative Yes            1
## 3 aSyn-SAA positive No            13
## 4 aSyn-SAA positive Yes           1
```

CBSdf %>% group_by(RTQUIC) %>% count(LimbRigidity)

```
## # A tibble: 4 × 3
## # Groups: RTQUIC [2]
##   RTQUIC          LimbRigidity     n
##   <fct>            <chr>                <int>
## 1 aSyn-SAA negative No             8
## 2 aSyn-SAA negative Yes            17
## 3 aSyn-SAA positive No            7
## 4 aSyn-SAA positive Yes           7
```

CBSdf %>% group_by(RTQUIC) %>% count(Slowness_binary)

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           Slowness_binary     n
##   <fct>            <chr>             <int>
## 1 aSyn-SAA negative No      5
## 2 aSyn-SAA negative Yes    20
## 3 aSyn-SAA positive No     4
## 4 aSyn-SAA positive Yes    10
```

CBSdf %>% group_by(RTQUIC) %>% count(LP2_falls_PI)

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           LP2_falls_PI     n
##   <fct>            <chr>            <int>
## 1 aSyn-SAA negative No      6
## 2 aSyn-SAA negative Yes    19
## 3 aSyn-SAA positive No     6
## 4 aSyn-SAA positive Yes    8
```

CBSdf %>% group_by(RTQUIC) %>% count(LP2_gait)

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           LP2_gait      n
##   <fct>            <chr>       <int>
## 1 aSyn-SAA negative No      8
## 2 aSyn-SAA negative Yes    17
## 3 aSyn-SAA positive No     9
## 4 aSyn-SAA positive Yes    5
```

CBSdf %>% group_by(RTQUIC) %>% count(RBD_binary) *#Very rare symptom (<3 in both groups) so no stats comparison within CBS (ie CBS-aSyn+ vs CBS-aSyn-).*

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           RBD_binary     n
##   <fct>            <chr>         <int>
## 1 aSyn-SAA negative No      24
## 2 aSyn-SAA negative Yes     1
## 3 aSyn-SAA positive No     12
## 4 aSyn-SAA positive Yes     2
```

CBSdf %>% group_by(RTQUIC) %>% count(Lifetime_Dopa_responder_true)

```
## # A tibble: 5 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           Lifetime_Dopa_responder_true     n
##   <fct>            <chr>                           <int>
## 1 aSyn-SAA negative No                      4
## 2 aSyn-SAA negative Yes                     2
## 3 aSyn-SAA negative <NA>                  19
## 4 aSyn-SAA positive No                     4
## 5 aSyn-SAA positive <NA>                  10
```

```
CBSdf %>% group_by(RTQUIC) %>% count(Lifetime_VisualHallucinations_binary)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           Lifetime_VisualHallucinations_binary     n
##   <fct>            <chr>                           <int>
## 1 aSyn-SAA negative No                      24
## 2 aSyn-SAA negative Yes                     1
## 3 aSyn-SAA positive No                     11
## 4 aSyn-SAA positive Yes                   3
```

```
CBSdf %>% group_by(RTQUIC) %>% count(Constipation_binary)
```

```
## # A tibble: 3 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           Constipation_binary     n
##   <fct>            <lgl>                           <int>
## 1 aSyn-SAA negative FALSE                 22
## 2 aSyn-SAA negative TRUE                  3
## 3 aSyn-SAA positive FALSE                14
```

CBSdf %>% group_by(RTQUIC) %>% count(Sexual_binary) *#Very rare symptom (<3 in both groups) so no stats comparison within CBS (ie CBS-aSyn+ vs CBS-aSyn-).*

```
## # A tibble: 3 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           Sexual_binary     n
##   <fct>            <lgl>                           <int>
## 1 aSyn-SAA negative FALSE                 24
## 2 aSyn-SAA negative TRUE                  1
## 3 aSyn-SAA positive FALSE                14
```

```
CBSdf %>% group_by(RTQUIC) %>% count(Urinary_binary)
```

```
## # A tibble: 4 × 3
## # Groups: RTQUIC [2]
##   RTQUIC          Urinary_binary     n
##   <fct>            <lgl>           <int>
## 1 aSyn-SAA negative FALSE        19
## 2 aSyn-SAA negative TRUE         6
## 3 aSyn-SAA positive FALSE       9
## 4 aSyn-SAA positive TRUE        5
```

```
CBSdf %>% group_by(RTQUIC) %>% count(Bowel_binary)
```

```
## # A tibble: 4 × 3
## # Groups: RTQUIC [2]
##   RTQUIC          Bowel_binary     n
##   <fct>            <lgl>           <int>
## 1 aSyn-SAA negative FALSE        22
## 2 aSyn-SAA negative TRUE         3
## 3 aSyn-SAA positive FALSE       12
## 4 aSyn-SAA positive TRUE        2
```

```
CBSdf %>% group_by(RTQUIC) %>% count(anyPPA)
```

```
## # A tibble: 4 × 3
## # Groups: RTQUIC [2]
##   RTQUIC          anyPPA     n
##   <fct>            <chr>    <int>
## 1 aSyn-SAA negative No        17
## 2 aSyn-SAA negative Yes      8
## 3 aSyn-SAA positive No       7
## 4 aSyn-SAA positive Yes      7
```

CBS ONLY STATISTICS: CHISQUARE OR FISHER TEST

```
if (chooseX2.func(table(CBSdf$Sex, CBSdf$RTQUIC)) == "chisquare") {
  chisq.test(table(CBSdf$Sex, CBSdf$RTQUIC))
} else fisher.test(CBSdf$Sex, CBSdf$RTQUIC)
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: table(CBSdf$Sex, CBSdf$RTQUIC)
## X-squared = 1.258, df = 1, p-value = 0.262
```

```
if (chooseX2.func(table(CBSdf$AP0Ee4, CBSdf$RTQUIC)) == "chisquare") {
  chisq.test(table(CBSdf$AP0Ee4, CBSdf$RTQUIC))
} else fisher.test(CBSdf$AP0Ee4, CBSdf$RTQUIC)
```

```
## 
## Fisher's Exact Test for Count Data
##
## data: CBSdf$AP0Ee4 and CBSdf$RTQUIC
## p-value = 0.3828
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.005029025 2.566732716
## sample estimates:
## odds ratio
## 0.2580524
```

```
if (chooseX2.func(table(CBSdf$Parkinsonian_onset, CBSdf$RTQUIC)) == "chisquare") {
  chisq.test(table(CBSdf$Parkinsonian_onset, CBSdf$RTQUIC))
} else fisher.test(CBSdf$Parkinsonian_onset, CBSdf$RTQUIC)
```

```
## 
## Fisher's Exact Test for Count Data
##
## data: CBSdf$Parkinsonian_onset and CBSdf$RTQUIC
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.1484647 4.2812730
## sample estimates:
## odds ratio
## 0.8535194
```

```
if (chooseX2.func(table(CBSdf$Tremor_binary, CBSdf$RTQUIC)) == "chisquare") {
  chisq.test(table(CBSdf$Tremor_binary, CBSdf$RTQUIC))
} else fisher.test(CBSdf$Tremor_binary, CBSdf$RTQUIC)
```

```
## 
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: table(CBSdf$Tremor_binary, CBSdf$RTQUIC)
## X-squared = 0.13379, df = 1, p-value = 0.7145
```

```
if (chooseX2.func(table(CBSdf$LimbRigidity, CBSdf$RTQUIC)) == "chisquare") {
  chisq.test(table(CBSdf$LimbRigidity, CBSdf$RTQUIC))
} else fisher.test(CBSdf$LimbRigidity, CBSdf$RTQUIC)
```

```
## 
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: table(CBSdf$LimbRigidity, CBSdf$RTQUIC)
## X-squared = 0.5857, df = 1, p-value = 0.4441
```

```
if (chooseX2.func(table(CBSdf$Slowness_binary, CBSdf$RTQUIC)) == "chisquare") {
  chisq.test(table(CBSdf$Slowness_binary, CBSdf$RTQUIC))
} else fisher.test(CBSdf$Slowness_binary, CBSdf$RTQUIC)
```

```
##
## Fisher's Exact Test for Count Data
##
## data: CBSdf$Slowness_binary and CBSdf$RTQUIC
## p-value = 0.6958
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.1074803 3.9350123
## sample estimates:
## odds ratio
## 0.6328383
```

```
if (chooseX2.func(table(CBSdf$LP2_falls_PI, CBSdf$RTQUIC)) == "chisquare") {
  chisq.test(table(CBSdf$LP2_falls_PI, CBSdf$RTQUIC))
} else fisher.test(CBSdf$LP2_falls_PI, CBSdf$RTQUIC)
```

```
##
## Fisher's Exact Test for Count Data
##
## data: CBSdf$LP2_falls_PI and CBSdf$RTQUIC
## p-value = 0.2869
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.08353211 2.15495833
## sample estimates:
## odds ratio
## 0.4311153
```

```
if (chooseX2.func(table(CBSdf$LP2_gait, CBSdf$RTQUIC)) == "chisquare") {
  chisq.test(table(CBSdf$LP2_gait, CBSdf$RTQUIC))
} else fisher.test(CBSdf$LP2_gait, CBSdf$RTQUIC)
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: table(CBSdf$LP2_gait, CBSdf$RTQUIC)
## X-squared = 2.6046, df = 1, p-value = 0.1066
```

```
if (chooseX2.func(table(CBSdf$Lifetime_Dopa_responder_true, CBSdf$RTQUIC)) == "chisquare") {
  chisq.test(table(CBSdf$Lifetime_Dopa_responder_true, CBSdf$RTQUIC))
} else fisher.test(CBSdf$Lifetime_Dopa_responder_true, CBSdf$RTQUIC)
```

```

## Fisher's Exact Test for Count Data
##
## data: CBSdf$Lifetime_Dopa_responder_true and CBSdf$RTQUIC
## p-value = 0.4667
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.000000 8.076414
## sample estimates:
## odds ratio
## 0

```

```

if (chooseX2.func(table(CBSdf$Lifetime_VisualHallucinations_binary, CBSdf$RTQUIC)) == "chisquare") {
  chisq.test(table(CBSdf$Lifetime_VisualHallucinations_binary, CBSdf$RTQUIC))
} else fisher.test(CBSdf$Lifetime_VisualHallucinations_binary, CBSdf$RTQUIC)

```

```

## Fisher's Exact Test for Count Data
##
## data: CBSdf$Lifetime_VisualHallucinations_binary and CBSdf$RTQUIC
## p-value = 0.1228
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.442479 356.700748
## sample estimates:
## odds ratio
## 6.21328

```

```

if (chooseX2.func(table(CBSdf$Constipation_binary, CBSdf$RTQUIC)) == "chisquare") {
  chisq.test(table(CBSdf$Constipation_binary, CBSdf$RTQUIC))
} else fisher.test(CBSdf$Constipation_binary, CBSdf$RTQUIC)

```

```

## Fisher's Exact Test for Count Data
##
## data: CBSdf$Constipation_binary and CBSdf$RTQUIC
## p-value = 0.5404
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.000000 4.314503
## sample estimates:
## odds ratio
## 0

```

```

if (chooseX2.func(table(CBSdf$Urinary_binary, CBSdf$RTQUIC)) == "chisquare") {
  chisq.test(table(CBSdf$Urinary_binary, CBSdf$RTQUIC))
} else fisher.test(CBSdf$Urinary_binary, CBSdf$RTQUIC)

```

```

## Fisher's Exact Test for Count Data
##
## data: CBSdf$Urinary_binary and CBSdf$RTQUIC
## p-value = 0.478
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.324475 9.090888
## sample estimates:
## odds ratio
## 1.732988

```

```

if (chooseX2.func(table(CBSdf$Bowel_binary, CBSdf$RTQUIC)) == "chisquare") {
  chisq.test(table(CBSdf$Bowel_binary, CBSdf$RTQUIC))
} else fisher.test(CBSdf$Bowel_binary, CBSdf$RTQUIC)

```

```

##
## Fisher's Exact Test for Count Data
##
## data: CBSdf$Bowel_binary and CBSdf$RTQUIC
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.08986523 12.22872401
## sample estimates:
## odds ratio
## 1.215804

```

```

if (chooseX2.func(table(CBSdf$anyPPA, CBSdf$RTQUIC)) == "chisquare") {
  chisq.test(table(CBSdf$anyPPA, CBSdf$RTQUIC))
} else fisher.test(CBSdf$anyPPA, CBSdf$RTQUIC)

```

```

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: table(CBSdf$anyPPA, CBSdf$RTQUIC)
## X-squared = 0.5857, df = 1, p-value = 0.4441

```

8.2.3. PSP-ONLY CATEGORICAL VARIABLES

Comparisons shown in: eTable 1

PSP ONLY STATISTICS: DISTRIBUTION

```
shapiro.test(PSPdf[PSPdf$RTQUIC=="aSyn-SAA positive", ]$Age) #normal
```

```
## 
## Shapiro-Wilk normality test
## 
## data: PSPdf[PSPdf$RTQUIC == "aSyn-SAA positive", ]$Age
## W = 0.96, p-value = 0.8101
```

```
shapiro.test(PSPdf[PSPdf$RTQUIC=="aSyn-SAA negative", ]$Age) #normal
```

```
## 
## Shapiro-Wilk normality test
## 
## data: PSPdf[PSPdf$RTQUIC == "aSyn-SAA negative", ]$Age
## W = 0.93967, p-value = 0.2363
```

```
var.test(Age ~ RTQUIC, data = PSPdf) #homoscedasticity
```

```
## 
## F test to compare two variances
## 
## data: Age by RTQUIC
## F = 1.9241, num df = 19, denom df = 7, p-value = 0.3827
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.4292154 5.8702728
## sample estimates:
## ratio of variances
## 1.924132
```

```
shapiro.test(PSPdf[PSPdf$RTQUIC=="aSyn-SAA positive", ]$Onset_age) #normal
```

```
## 
## Shapiro-Wilk normality test
## 
## data: PSPdf[PSPdf$RTQUIC == "aSyn-SAA positive", ]$Onset_age
## W = 0.90133, p-value = 0.297
```

```
shapiro.test(PSPdf[PSPdf$RTQUIC=="aSyn-SAA negative", ]$Onset_age) #normal
```

```
## 
## Shapiro-Wilk normality test
## 
## data: PSPdf[PSPdf$RTQUIC == "aSyn-SAA negative", ]$Onset_age
## W = 0.96336, p-value = 0.613
```

```
var.test(Onset_age ~ RTQUIC, data = PSPdf) #homoscedasticity
```

```

## 
## F test to compare two variances
##
## data: Onset_age by RTQUIC
## F = 1.6274, num df = 19, denom df = 7, p-value = 0.5252
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.3630251 4.9650044
## sample estimates:
## ratio of variances
## 1.627407

```

```
shapiro.test(PSPdf[PSPdf$RTQUIC=="aSyn-SAA positive", ]$Park_onset) #normal
```

```

## 
## Shapiro-Wilk normality test
##
## data: PSPdf[PSPdf$RTQUIC == "aSyn-SAA positive", ]$Park_onset
## W = 0.89162, p-value = 0.2423

```

```
shapiro.test(PSPdf[PSPdf$RTQUIC=="aSyn-SAA negative", ]$Park_onset) #normal
```

```

## 
## Shapiro-Wilk normality test
##
## data: PSPdf[PSPdf$RTQUIC == "aSyn-SAA negative", ]$Park_onset
## W = 0.9484, p-value = 0.3434

```

```
var.test(Park_onset ~ RTQUIC, data = PSPdf) #homoscedasticity
```

```

## 
## F test to compare two variances
##
## data: Park_onset by RTQUIC
## F = 1.7731, num df = 19, denom df = 7, p-value = 0.4485
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.3955307 5.4095760
## sample estimates:
## ratio of variances
## 1.773127

```

```
shapiro.test(PSPdf[PSPdf$RTQUIC=="aSyn-SAA positive", ]$Age) #normal
```

```
## 
## Shapiro-Wilk normality test
## 
## data: PSPdf[PSPdf$RTQUIC == "aSyn-SAA positive", ]$Age
## W = 0.96, p-value = 0.8101
```

```
shapiro.test(PSPdf[PSPdf$RTQUIC=="aSyn-SAA negative", ]$Age) #normal
```

```
## 
## Shapiro-Wilk normality test
## 
## data: PSPdf[PSPdf$RTQUIC == "aSyn-SAA negative", ]$Age
## W = 0.93967, p-value = 0.2363
```

```
var.test(Age ~ RTQUIC, data = PSPdf) #homoscedasticity
```

```
## 
## F test to compare two variances
## 
## data: Age by RTQUIC
## F = 1.9241, num df = 19, denom df = 7, p-value = 0.3827
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.4292154 5.8702728
## sample estimates:
## ratio of variances
## 1.924132
```

PSP ONLY STATISTICS: SUMMARY

```
PSPdf %>% group_by(RTQUIC) %>% summarize(format(round(mean(Age, na.rm=T),2),2), sd=sd(Age, na.rm=T))
```

```
## # A tibble: 2 × 3
##   RTQUIC           `format(round(mean(Age, na.rm = T), 2), 2)`     sd
##   <fct>             <chr>                                         <dbl>
## 1 aSyn-SAA negative 71.93                                         9.45
## 2 aSyn-SAA positive  73.95                                         6.81
```

```
PSPdf %>% group_by(RTQUIC) %>% summarize(format(round(mean(Onset_age, na.rm=T),2),2), sd =sd(Onset_age, na.rm=T))
```

```
## # A tibble: 2 × 3
##   RTQUIC      `format(round(mean(Onset_age, na.rm = T), 2), 2)`    sd
##   <fct>        <chr>
## 1 aSyn-SAA negative 64.85          8.61
## 2 aSyn-SAA positive 69.88          6.75
```

```
PSPdf %>% group_by(RTQUIC) %>% summarize(format(round(mean(Park_onset, na.rm=T),2),2), s=sd(Park_onset, na.rm=T))
```

```
## # A tibble: 2 × 3
##   RTQUIC      `format(round(mean(Park_onset, na.rm = T), 2), 2)`    sd
##   <fct>        <chr>
## 1 aSyn-SAA negative 65.9          8.68
## 2 aSyn-SAA positive 70.25         6.52
```

PSP ONLY STATISTICS: T-TEST

```
t.test(PSPdf$Age ~ PSPdf$RTQUIC, var.equal=TRUE)
```

```
##
## Two Sample t-test
##
## data: PSPdf$Age by PSPdf$RTQUIC
## t = -0.54813, df = 26, p-value = 0.5883
## alternative hypothesis: true difference in means between group aSyn-SAA negative and
## group aSyn-SAA positive is not equal to 0
## 95 percent confidence interval:
## -9.601622 5.558882
## sample estimates:
## mean in group aSyn-SAA negative mean in group aSyn-SAA positive
## 71.93205 73.95342
```

```
t.test(PSPdf$Onset_age ~ PSPdf$RTQUIC, var.equal=TRUE)
```

```
##
## Two Sample t-test
##
## data: PSPdf$Onset_age by PSPdf$RTQUIC
## t = -1.4737, df = 26, p-value = 0.1526
## alternative hypothesis: true difference in means between group aSyn-SAA negative and
## group aSyn-SAA positive is not equal to 0
## 95 percent confidence interval:
## -12.033993 1.983993
## sample estimates:
## mean in group aSyn-SAA negative mean in group aSyn-SAA positive
## 64.850 69.875
```

```
t.test(PSPdf$Park_onset ~ PSPdf$RTQUIC, var.equal=TRUE)
```

```
## 
## Two Sample t-test
##
## data: PSPdf$Park_onset by PSPdf$RTQUIC
## t = -1.275, df = 26, p-value = 0.2136
## alternative hypothesis: true difference in means between group aSyn-SAA negative and
## group aSyn-SAA positive is not equal to 0
## 95 percent confidence interval:
## -11.362796 2.662796
## sample estimates:
## mean in group aSyn-SAA negative mean in group aSyn-SAA positive
## 65.90 70.25
```

8.2.4. PSP-ONLY CATEGORICAL VARIABLES

Comparisons shown in: eTable 1

PSP ONLY STATISTICS: SUMMARY

```
PSPdf %>% group_by(RTQUIC) %>% count(Sex)
```

```
## # A tibble: 4 × 3
## # Groups: RTQUIC [2]
##   RTQUIC       Sex     n
##   <fct>      <chr> <int>
## 1 aSyn-SAA negative F     10
## 2 aSyn-SAA negative M     10
## 3 aSyn-SAA positive F     3
## 4 aSyn-SAA positive M     5
```

```
PSPdf %>% group_by(RTQUIC) %>% count(AP0Ee4)
```

```
## # A tibble: 6 × 3
## # Groups: RTQUIC [2]
##   RTQUIC       AP0Ee4     n
##   <fct>      <fct> <int>
## 1 aSyn-SAA negative Negative 13
## 2 aSyn-SAA negative Positive  5
## 3 aSyn-SAA negative <NA>     2
## 4 aSyn-SAA positive Negative  4
## 5 aSyn-SAA positive Positive  2
## 6 aSyn-SAA positive <NA>     2
```

```
PSPdf %>% group_by(RTQUIC) %>% count(Parkinsonian_onset)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC          Parkinsonian_onset     n
##   <fct>            <chr>              <int>
## 1 aSyn-SAA negative      No           6
## 2 aSyn-SAA negative      Yes          14
## 3 aSyn-SAA positive      No           2
## 4 aSyn-SAA positive      Yes          6
```

```
PSPdf %>% group_by(RTQUIC) %>% count(Tremor_binary)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC          Tremor_binary     n
##   <fct>            <chr>          <int>
## 1 aSyn-SAA negative      No        11
## 2 aSyn-SAA negative      Yes       9
## 3 aSyn-SAA positive      No        3
## 4 aSyn-SAA positive      Yes       5
```

PSPdf %>% group_by(RTQUIC) %>% count(RestTremor) *#Very rare symptom (<3 in both groups) so no stats comparison within PSP (ie PSP-aSyn+ vs PSP-aSyn-).*

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC          RestTremor     n
##   <fct>            <chr>          <int>
## 1 aSyn-SAA negative      No       18
## 2 aSyn-SAA negative      Yes       2
## 3 aSyn-SAA positive      No       7
## 4 aSyn-SAA positive      Yes       1
```

```
PSPdf %>% group_by(RTQUIC) %>% count(LimbRigidity)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC          LimbRigidity     n
##   <fct>            <chr>          <int>
## 1 aSyn-SAA negative      No       3
## 2 aSyn-SAA negative      Yes      17
## 3 aSyn-SAA positive      No       3
## 4 aSyn-SAA positive      Yes      5
```

```
PSPdf %>% group_by(RTQUIC) %>% count(Slowness_binary)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           Slowness_binary     n
##   <fct>            <chr>             <int>
## 1 aSyn-SAA negative No          1
## 2 aSyn-SAA negative Yes        19
## 3 aSyn-SAA positive No         1
## 4 aSyn-SAA positive Yes        7
```

```
PSPdf %>% group_by(RTQUIC) %>% count(LP2_falls_PI)
```

```
## # A tibble: 3 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           LP2_falls_PI     n
##   <fct>            <chr>             <int>
## 1 aSyn-SAA negative No          1
## 2 aSyn-SAA negative Yes        19
## 3 aSyn-SAA positive Yes        8
```

```
PSPdf %>% group_by(RTQUIC) %>% count(LP2_gait)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           LP2_gait      n
##   <fct>            <chr>       <int>
## 1 aSyn-SAA negative No          2
## 2 aSyn-SAA negative Yes        18
## 3 aSyn-SAA positive No         1
## 4 aSyn-SAA positive Yes        7
```

```
PSPdf %>% group_by(RTQUIC) %>% count(RBD_binary)
```

```
## # A tibble: 3 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           RBD_binary     n
##   <fct>            <chr>             <int>
## 1 aSyn-SAA negative No          20
## 2 aSyn-SAA positive No          4
## 3 aSyn-SAA positive Yes         4
```

```
PSPdf %>% group_by(RTQUIC) %>% count(Lifetime_Dopa_responder_true)
```

```
## # A tibble: 6 × 3
## # Groups: RTQUIC [2]
##   RTQUIC           Lifetime_Dopa_responder_true     n
##   <fct>            <chr>                           <int>
## 1 aSyn-SAA negative No                      10
## 2 aSyn-SAA negative Yes                     4
## 3 aSyn-SAA negative <NA>                   6
## 4 aSyn-SAA positive No                     2
## 5 aSyn-SAA positive Yes                    3
## 6 aSyn-SAA positive <NA>                   3
```

```
PSPdf %>% group_by(RTQUIC) %>% count(Lifetime_VisualHallucinations_binary)
```

```
## # A tibble: 4 × 3
## # Groups: RTQUIC [2]
##   RTQUIC           Lifetime_VisualHallucinations_binary     n
##   <fct>            <chr>                           <int>
## 1 aSyn-SAA negative No                      17
## 2 aSyn-SAA negative Yes                     3
## 3 aSyn-SAA positive No                     6
## 4 aSyn-SAA positive Yes                    2
```

```
PSPdf %>% group_by(RTQUIC) %>% count(Constipation_binary)
```

```
## # A tibble: 4 × 3
## # Groups: RTQUIC [2]
##   RTQUIC           Constipation_binary     n
##   <fct>            <lgl>                           <int>
## 1 aSyn-SAA negative FALSE                  14
## 2 aSyn-SAA negative TRUE                   6
## 3 aSyn-SAA positive FALSE                 6
## 4 aSyn-SAA positive TRUE                  2
```

PSPdf %>% group_by(RTQUIC) %>% count(Sexual_binary) #Very rare symptom (<3 in both groups) so no stats comparison within PSP (ie PSP-aSyn+ vs PSP-aSyn-).

```
## # A tibble: 4 × 3
## # Groups: RTQUIC [2]
##   RTQUIC           Sexual_binary     n
##   <fct>            <lgl>                           <int>
## 1 aSyn-SAA negative FALSE                  18
## 2 aSyn-SAA negative TRUE                   2
## 3 aSyn-SAA positive FALSE                 7
## 4 aSyn-SAA positive TRUE                  1
```

```
PSPdf %>% group_by(RTQUIC) %>% count(Urinary_binary)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           Urinary_binary     n
##   <fct>            <lgl>          <int>
## 1 aSyn-SAA negative FALSE         3
## 2 aSyn-SAA negative TRUE        17
## 3 aSyn-SAA positive FALSE       2
## 4 aSyn-SAA positive TRUE        6
```

```
PSPdf %>% group_by(RTQUIC) %>% count(Bowel_binary)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           Bowel_binary     n
##   <fct>            <lgl>          <int>
## 1 aSyn-SAA negative FALSE        16
## 2 aSyn-SAA negative TRUE         4
## 3 aSyn-SAA positive FALSE        7
## 4 aSyn-SAA positive TRUE         1
```

PSP ONLY STATISTICS: CHISQUARE OR FISHER TEST

```
if (chooseX2.func(table(PSPdf$Sex, PSPdf$RTQUIC)) == "chisquare") {
  chisq.test(table(PSPdf$Sex, PSPdf$RTQUIC))
} else fisher.test(PSPdf$Sex, PSPdf$RTQUIC)
```

```
##
## Fisher's Exact Test for Count Data
##
## data: PSPdf$Sex and PSPdf$RTQUIC
## p-value = 0.686
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.2384413 13.5110071
## sample estimates:
## odds ratio
## 1.636637
```

```
if (chooseX2.func(table(PSPdf$AP0Ee4, PSPdf$RTQUIC)) == "chisquare") {
  chisq.test(table(PSPdf$AP0Ee4, PSPdf$RTQUIC))
} else fisher.test(PSPdf$AP0Ee4, PSPdf$RTQUIC)
```

```
## 
## Fisher's Exact Test for Count Data
##
## data: PSPdf$AP0Ee4 and PSPdf$RTQUIC
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.08937451 12.95765657
## sample estimates:
## odds ratio
## 1.285414
```

```
if (chooseX2.func(table(PSPdf$Parkinsonian_onset, PSPdf$RTQUIC)) == "chisquare") {
  chisq.test(table(PSPdf$Parkinsonian_onset, PSPdf$RTQUIC))
} else fisher.test(PSPdf$Parkinsonian_onset, PSPdf$RTQUIC)
```

```
## 
## Fisher's Exact Test for Count Data
##
## data: PSPdf$Parkinsonian_onset and PSPdf$RTQUIC
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.157035 16.475021
## sample estimates:
## odds ratio
## 1.274496
```

```
if (chooseX2.func(table(PSPdf$Tremor_binary, PSPdf$RTQUIC)) == "chisquare") {
  chisq.test(table(PSPdf$Tremor_binary, PSPdf$RTQUIC))
} else fisher.test(PSPdf$Tremor_binary, PSPdf$RTQUIC)
```

```
## 
## Fisher's Exact Test for Count Data
##
## data: PSPdf$Tremor_binary and PSPdf$RTQUIC
## p-value = 0.6776
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.2897147 16.4540308
## sample estimates:
## odds ratio
## 1.98554
```

```
if (chooseX2.func(table(PSPdf$LimbRigidity, PSPdf$RTQUIC)) == "chisquare") {
  chisq.test(table(PSPdf$LimbRigidity, PSPdf$RTQUIC))
} else fisher.test(PSPdf$LimbRigidity, PSPdf$RTQUIC)
```

```

## 
## Fisher's Exact Test for Count Data
##
## data: PSPdf$LimbRigidity and PSPdf$RTQUIC
## p-value = 0.3107
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.03036191 3.04869702
## sample estimates:
## odds ratio
## 0.3095039

```

```

if (chooseX2.func(table(PSPdf$Slowness_binary, PSPdf$RTQUIC)) == "chisquare") {
  chisq.test(table(PSPdf$Slowness_binary, PSPdf$RTQUIC))
} else fisher.test(PSPdf$Slowness_binary, PSPdf$RTQUIC)

```

```

## 
## Fisher's Exact Test for Count Data
##
## data: PSPdf$Slowness_binary and PSPdf$RTQUIC
## p-value = 0.4974
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.00447247 33.01630326
## sample estimates:
## odds ratio
## 0.3838855

```

```

if (chooseX2.func(table(PSPdf$LP2_falls_PI, PSPdf$RTQUIC)) == "chisquare") {
  chisq.test(table(PSPdf$LP2_falls_PI, PSPdf$RTQUIC))
} else fisher.test(PSPdf$LP2_falls_PI, PSPdf$RTQUIC)

```

```

## 
## Fisher's Exact Test for Count Data
##
## data: PSPdf$LP2_falls_PI and PSPdf$RTQUIC
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.01027188      Inf
## sample estimates:
## odds ratio
##      Inf

```

```

if (chooseX2.func(table(PSPdf$LP2_gait, PSPdf$RTQUIC)) == "chisquare") {
  chisq.test(table(PSPdf$LP2_gait, PSPdf$RTQUIC))
} else fisher.test(PSPdf$LP2_gait, PSPdf$RTQUIC)

```

```

## Fisher's Exact Test for Count Data
##
## data: PSPdf$LP2_gait and PSPdf$RTQUIC
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.03528132 52.36689579
## sample estimates:
## odds ratio
## 0.7850664

```

```

if (chooseX2.func(table(PSPdf$RBD_binary, PSPdf$RTQUIC)) == "chisquare") {
  chisq.test(table(PSPdf$RBD_binary, PSPdf$RTQUIC))
} else fisher.test(PSPdf$RBD_binary, PSPdf$RTQUIC)

```

```

##
## Fisher's Exact Test for Count Data
##
## data: PSPdf$RBD_binary and PSPdf$RTQUIC
## p-value = 0.003419
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 2.179296      Inf
## sample estimates:
## odds ratio
##      Inf

```

```

if (chooseX2.func(table(PSPdf$Lifetime_Dopa_responder_true, PSPdf$RTQUIC)) == "chisquare") {
  chisq.test(table(PSPdf$Lifetime_Dopa_responder_true, PSPdf$RTQUIC))
} else fisher.test(PSPdf$Lifetime_Dopa_responder_true, PSPdf$RTQUIC)

```

```

##
## Fisher's Exact Test for Count Data
##
## data: PSPdf$Lifetime_Dopa_responder_true and PSPdf$RTQUIC
## p-value = 0.3047
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.2826482 57.3397574
## sample estimates:
## odds ratio
## 3.470449

```

```
if (chooseX2.func(table(PSPdf$Lifetime_VisualHallucinations_binary, PSPdf$RTQUIC)) == "chisquare") {
  chisq.test(table(PSPdf$Lifetime_VisualHallucinations_binary, PSPdf$RTQUIC))
} else fisher.test(PSPdf$Lifetime_VisualHallucinations_binary, PSPdf$RTQUIC)
```

```
##
## Fisher's Exact Test for Count Data
##
## data: PSPdf$Lifetime_VisualHallucinations_binary and PSPdf$RTQUIC
## p-value = 0.6056
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.1252442 20.7164164
## sample estimates:
## odds ratio
## 1.8428
```

```
if (chooseX2.func(table(PSPdf$Constipation_binary, PSPdf$RTQUIC)) == "chisquare") {
  chisq.test(table(PSPdf$Constipation_binary, PSPdf$RTQUIC))
} else fisher.test(PSPdf$Constipation_binary, PSPdf$RTQUIC)
```

```
##
## Fisher's Exact Test for Count Data
##
## data: PSPdf$Constipation_binary and PSPdf$RTQUIC
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.06069795 6.36800803
## sample estimates:
## odds ratio
## 0.7846237
```

```
if (chooseX2.func(table(PSPdf$Urinary_binary, PSPdf$RTQUIC)) == "chisquare") {
  chisq.test(table(PSPdf$Urinary_binary, PSPdf$RTQUIC))
} else fisher.test(PSPdf$Urinary_binary, PSPdf$RTQUIC)
```

```
##
## Fisher's Exact Test for Count Data
##
## data: PSPdf$Urinary_binary and PSPdf$RTQUIC
## p-value = 0.6056
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.0482709 7.9844016
## sample estimates:
## odds ratio
## 0.5426526
```

```
if (chooseX2.func(table(PSPdf$Bowel_binary, PSPdf$RTQUIC)) == "chisquare") {
  chisq.test(table(PSPdf$Bowel_binary, PSPdf$RTQUIC))
} else fisher.test(PSPdf$Bowel_binary, PSPdf$RTQUIC)
```

```
##
## Fisher's Exact Test for Count Data
##
## data: PSPdf$Bowel_binary and PSPdf$RTQUIC
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.01017649 7.47489134
## sample estimates:
## odds ratio
## 0.5820007
```

8.2.5. BOTH DX NUMERICAL VARIABLES

GET FROM SECTIONS 5.2.

8.2.6. BOTH DX CATEGORICAL VARIABLES

Comparisons shown in: eTable 1

BOTH DX STATISTICS: SUMMARY

```
df %>% group_by(RTQUIC) %>% count(Sex)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           Sex     n
##   <fct>         <chr> <int>
## 1 aSyn-SAA negative F      20
## 2 aSyn-SAA negative M      25
## 3 aSyn-SAA positive F      12
## 4 aSyn-SAA positive M      10
```

```
df %>% group_by(RTQUIC) %>% count(AP0Ee4)
```

```
## # A tibble: 6 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC          AP0Ee4      n
##   <fct>          <fct>    <int>
## 1 aSyn-SAA negative Negative     31
## 2 aSyn-SAA negative Positive     11
## 3 aSyn-SAA negative <NA>        3
## 4 aSyn-SAA positive Negative    16
## 5 aSyn-SAA positive Positive     3
## 6 aSyn-SAA positive <NA>        3
```

```
df %>% group_by(RTQUIC) %>% count(Parkinsonian_onset)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC          Parkinsonian_onset      n
##   <fct>          <chr>            <int>
## 1 aSyn-SAA negative No             23
## 2 aSyn-SAA negative Yes           22
## 3 aSyn-SAA positive No            12
## 4 aSyn-SAA positive Yes           10
```

```
df %>% group_by(RTQUIC) %>% count(Tremor_binary)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC          Tremor_binary      n
##   <fct>          <chr>            <int>
## 1 aSyn-SAA negative No             26
## 2 aSyn-SAA negative Yes           19
## 3 aSyn-SAA positive No            13
## 4 aSyn-SAA positive Yes           9
```

```
df %>% group_by(RTQUIC) %>% count(RestTremor)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC          RestTremor      n
##   <fct>          <chr>            <int>
## 1 aSyn-SAA negative No             42
## 2 aSyn-SAA negative Yes           3
## 3 aSyn-SAA positive No            20
## 4 aSyn-SAA positive Yes           2
```

```
df %>% group_by(RTQUIC) %>% count(LimbRigidity)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           LimbRigidity     n
##   <fct>            <chr>          <int>
## 1 aSyn-SAA negative      No        11
## 2 aSyn-SAA negative      Yes       34
## 3 aSyn-SAA positive      No        10
## 4 aSyn-SAA positive      Yes       12
```

```
df %>% group_by(RTQUIC) %>% count(Slowness_binary)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           Slowness_binary     n
##   <fct>            <chr>          <int>
## 1 aSyn-SAA negative      No        6
## 2 aSyn-SAA negative      Yes      39
## 3 aSyn-SAA positive      No        5
## 4 aSyn-SAA positive      Yes      17
```

```
df %>% group_by(RTQUIC) %>% count(LP2_falls_PI)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           LP2_falls_PI     n
##   <fct>            <chr>          <int>
## 1 aSyn-SAA negative      No        7
## 2 aSyn-SAA negative      Yes      38
## 3 aSyn-SAA positive      No        6
## 4 aSyn-SAA positive      Yes      16
```

```
df %>% group_by(RTQUIC) %>% count(LP2_gait)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           LP2_gait      n
##   <fct>            <chr>        <int>
## 1 aSyn-SAA negative      No      10
## 2 aSyn-SAA negative      Yes     35
## 3 aSyn-SAA positive      No      10
## 4 aSyn-SAA positive      Yes     12
```

```
df %>% group_by(RTQUIC) %>% count(RBD_binary)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           RBD_binary     n
##   <fct>            <chr>        <int>
## 1 aSyn-SAA negative No          44
## 2 aSyn-SAA negative Yes         1
## 3 aSyn-SAA positive No         16
## 4 aSyn-SAA positive Yes         6
```

```
df %>% group_by(RTQUIC) %>% count(Lifetime_Dopa_responder_true)
```

```
## # A tibble: 6 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           Lifetime_Dopa_responder_true     n
##   <fct>            <chr>                    <int>
## 1 aSyn-SAA negative No          14
## 2 aSyn-SAA negative Yes         6
## 3 aSyn-SAA negative <NA>       25
## 4 aSyn-SAA positive No         6
## 5 aSyn-SAA positive Yes         3
## 6 aSyn-SAA positive <NA>       13
```

```
df %>% group_by(RTQUIC) %>% count(Lifetime_VisualHallucinations_binary)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           Lifetime_VisualHallucinations_binary     n
##   <fct>            <chr>                    <int>
## 1 aSyn-SAA negative No          41
## 2 aSyn-SAA negative Yes         4
## 3 aSyn-SAA positive No         17
## 4 aSyn-SAA positive Yes         5
```

```
df %>% group_by(RTQUIC) %>% count(Constipation_binary)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC           Constipation_binary     n
##   <fct>            <lgl>        <int>
## 1 aSyn-SAA negative FALSE      36
## 2 aSyn-SAA negative TRUE       9
## 3 aSyn-SAA positive FALSE     20
## 4 aSyn-SAA positive TRUE      2
```

```
df %>% group_by(RTQUIC) %>% count(Sexual_binary)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC      Sexual_binary     n
##   <fct>        <lgl>       <int>
## 1 aSyn-SAA negative FALSE     42
## 2 aSyn-SAA negative TRUE      3
## 3 aSyn-SAA positive FALSE    21
## 4 aSyn-SAA positive TRUE     1
```

```
df %>% group_by(RTQUIC) %>% count(Urinary_binary)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC      Urinary_binary     n
##   <fct>        <lgl>       <int>
## 1 aSyn-SAA negative FALSE     22
## 2 aSyn-SAA negative TRUE     23
## 3 aSyn-SAA positive FALSE    11
## 4 aSyn-SAA positive TRUE     11
```

```
df %>% group_by(RTQUIC) %>% count(Bowel_binary)
```

```
## # A tibble: 4 × 3
## # Groups:   RTQUIC [2]
##   RTQUIC      Bowel_binary     n
##   <fct>        <lgl>       <int>
## 1 aSyn-SAA negative FALSE     38
## 2 aSyn-SAA negative TRUE      7
## 3 aSyn-SAA positive FALSE    19
## 4 aSyn-SAA positive TRUE     3
```

BOTH DX STATISTICS: CHISQUARE OR FISHER TEST

```
if (chooseX2.func(table(df$Sex, df$RTQUIC)) == "chisquare") {
  chisq.test(table(df$Sex, df$RTQUIC))
} else fisher.test(df$Sex, df$RTQUIC)
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: table(df$Sex, df$RTQUIC)
## X-squared = 0.26722, df = 1, p-value = 0.6052
```

```
if (chooseX2.func(table(df$AP0Ee4, df$RTQUIC)) == "chisquare") {
  chisq.test(table(df$AP0Ee4, df$RTQUIC))
} else fisher.test(df$AP0Ee4, df$RTQUIC)
```

```
##  
## Fisher's Exact Test for Count Data  
##  
## data: df$APOEe4 and df$RTQUIC  
## p-value = 0.5164  
## alternative hypothesis: true odds ratio is not equal to 1  
## 95 percent confidence interval:  
## 0.08370541 2.43483909  
## sample estimates:  
## odds ratio  
## 0.5335897
```

```
if (chooseX2.func(table(df$Parkinsonian_onset, df$RTQUIC)) == "chisquare") {  
  chisq.test(table(df$Parkinsonian_onset, df$RTQUIC))  
} else fisher.test(df$Parkinsonian_onset, df$RTQUIC)
```

```
##  
## Pearson's Chi-squared test with Yates' continuity correction  
##  
## data: table(df$Parkinsonian_onset, df$RTQUIC)  
## X-squared = 1.5106e-05, df = 1, p-value = 0.9969
```

```
if (chooseX2.func(table(df$Tremor_binary, df$RTQUIC)) == "chisquare") {  
  chisq.test(table(df$Tremor_binary, df$RTQUIC))  
} else fisher.test(df$Tremor_binary, df$RTQUIC)
```

```
##  
## Pearson's Chi-squared test with Yates' continuity correction  
##  
## data: table(df$Tremor_binary, df$RTQUIC)  
## X-squared = 0, df = 1, p-value = 1
```

```
if (chooseX2.func(table(df$LimbRigidity, df$RTQUIC)) == "chisquare") {  
  chisq.test(table(df$LimbRigidity, df$RTQUIC))  
} else fisher.test(df$LimbRigidity, df$RTQUIC)
```

```
##  
## Pearson's Chi-squared test with Yates' continuity correction  
##  
## data: table(df$LimbRigidity, df$RTQUIC)  
## X-squared = 2.1333, df = 1, p-value = 0.1441
```

```
if (chooseX2.func(table(df$Slowness_binary, df$RTQUIC)) == "chisquare") {  
  chisq.test(table(df$Slowness_binary, df$RTQUIC))  
} else fisher.test(df$Slowness_binary, df$RTQUIC)
```

```

## Fisher's Exact Test for Count Data
##
## data: df$Slowness_binary and df$RTQUIC
## p-value = 0.4834
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.115735 2.509391
## sample estimates:
## odds ratio
## 0.5285138

```

```

if (chooseX2.func(table(df$LP2_falls_PI, df$RTQUIC)) == "chisquare") {
  chisq.test(table(df$LP2_falls_PI, df$RTQUIC))
} else fisher.test(df$LP2_falls_PI, df$RTQUIC)

```

```

##
## Fisher's Exact Test for Count Data
##
## data: df$LP2_falls_PI and df$RTQUIC
## p-value = 0.3271
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.1204831 2.0925592
## sample estimates:
## odds ratio
## 0.4968569

```

```

if (chooseX2.func(table(df$LP2_gait, df$RTQUIC)) == "chisquare") {
  chisq.test(table(df$LP2_gait, df$RTQUIC))
} else fisher.test(df$LP2_gait, df$RTQUIC)

```

```

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: table(df$LP2_gait, df$RTQUIC)
## X-squared = 2.7799, df = 1, p-value = 0.09545

```

```

if (chooseX2.func(table(df$RBD_binary, df$RTQUIC)) == "chisquare") {
  chisq.test(table(df$RBD_binary, df$RTQUIC))
} else fisher.test(df$RBD_binary, df$RTQUIC)

```

```
## 
## Fisher's Exact Test for Count Data
##
## data: df$RBD_binary and df$RTQUIC
## p-value = 0.004057
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##     1.715434 772.058733
## sample estimates:
## odds ratio
##    15.75251
```

```
if (chooseX2.func(table(df$Lifetime_Dopa_responder_true, df$RTQUIC)) == "chisquare") {
  chisq.test(table(df$Lifetime_Dopa_responder_true, df$RTQUIC))
} else fisher.test(df$Lifetime_Dopa_responder_true, df$RTQUIC)
```

```
## 
## Fisher's Exact Test for Count Data
##
## data: df$Lifetime_Dopa_responder_true and df$RTQUIC
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##    0.1403631 8.0655487
## sample estimates:
## odds ratio
##    1.160399
```

```
if (chooseX2.func(table(df$Lifetime_VisualHallucinations_binary, df$RTQUIC)) == "chisquare") {
  chisq.test(table(df$Lifetime_VisualHallucinations_binary, df$RTQUIC))
} else fisher.test(df$Lifetime_VisualHallucinations_binary, df$RTQUIC)
```

```
## 
## Fisher's Exact Test for Count Data
##
## data: df$Lifetime_VisualHallucinations_binary and df$RTQUIC
## p-value = 0.1415
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##    0.5616313 16.8749320
## sample estimates:
## odds ratio
##    2.959418
```

```
if (chooseX2.func(table(df$Constipation_binary, df$RTQUIC)) == "chisquare") {
  chisq.test(table(df$Constipation_binary, df$RTQUIC))
} else fisher.test(df$Constipation_binary, df$RTQUIC)
```

```

## Fisher's Exact Test for Count Data
##
## data: df$Constipation_binary and df$RTQUIC
## p-value = 0.3165
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.03891295 2.24031974
## sample estimates:
## odds ratio
## 0.4048595

```

```

if (chooseX2.func(table(df$Sexual_binary, df$RTQUIC)) == "chisquare") {
  chisq.test(table(df$Sexual_binary, df$RTQUIC))
} else fisher.test(df$Sexual_binary, df$RTQUIC)

```

```

##
## Fisher's Exact Test for Count Data
##
## data: df$Sexual_binary and df$RTQUIC
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.01213466 8.94481037
## sample estimates:
## odds ratio
## 0.6704391

```

```

if (chooseX2.func(table(df$Urinary_binary, df$RTQUIC)) == "chisquare") {
  chisq.test(table(df$Urinary_binary, df$RTQUIC))
} else fisher.test(df$Urinary_binary, df$RTQUIC)

```

```

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: table(df$Urinary_binary, df$RTQUIC)
## X-squared = 0, df = 1, p-value = 1

```

```

if (chooseX2.func(table(df$Bowel_binary, df$RTQUIC)) == "chisquare") {
  chisq.test(table(df$Bowel_binary, df$RTQUIC))
} else fisher.test(df$Bowel_binary, df$RTQUIC)

```

```
##  
## Fisher's Exact Test for Count Data  
##  
## data: df$Bowel_binary and df$RTQUIC  
## p-value = 1  
## alternative hypothesis: true odds ratio is not equal to 1  
## 95 percent confidence interval:  
## 0.1289181 4.3125694  
## sample estimates:  
## odds ratio  
## 0.8590504
```

8.2.7. FIGURE 1.C. RADAR PLOTS

Fig 1.c.

CBS

List = list() #Create an empty list to which you will assign the values from the for-loop. By creating this list outside of the main for-loop, you allow for the data to be entered under different entries which means you can call all the values you need.

```

rt.value <- c("aSyn-SAA positive", "aSyn-SAA negative")
for (rt in rt.value) { #for-loop that tests each value of the variable

  rtCBSdf <- CBSdf[CBSdf$RTQUIC== rt, ] #Within subset of CBS for eg, look for subset of RT+
  
  print(rt)

  # SUMMARIZE THE CATEGORICAL VARIABLES INTO ONE SINGLE COUNT OF "YES"
  Tremor_perc <- rtCBSdf %>% summarise(Tremor_count= sum(Tremor_binary == "Yes")) %>% mutate(Tremor_count= (as.numeric(Tremor_count)/nrow(rtCBSdf)*100))
  RestTremor_perc <- rtCBSdf %>% summarise(RestTremor_count= sum(RestTremor == "Yes")) %>% mutate(RestTremor_count= (as.numeric(RestTremor_count)/nrow(rtCBSdf)*100))
  LimbRigidity_perc <- rtCBSdf %>% summarise(LimbRigidity_count= sum(LimbRigidity == "Yes")) %>% mutate(LimbRigidity_count= (as.numeric(LimbRigidity_count)/nrow(rtCBSdf)*100))
  Slowness_perc <- rtCBSdf %>% summarise(Slowness_count= sum(Slowness_binary == "Yes")) %>% mutate(Slowness_count= (as.numeric(Slowness_count)/nrow(rtCBSdf)*100))
  Apraxia_perc <- rtCBSdf %>% summarise(Apraxia_count= sum(LP2_apraxia == "Yes")) %>% mutate(Apraxia_count= (as.numeric(Apraxia_count)/nrow(rtCBSdf)*100))
  Gait_perc <- rtCBSdf %>% summarise(Gait_count= sum(LP2_gait == "Yes")) %>% mutate(Gait_count= (as.numeric(Gait_count)/nrow(rtCBSdf)*100))
  FallsPI_perc <- rtCBSdf %>% summarise(FallsPI_count= sum(LP2_falls_PI == "Yes")) %>% mutate(FallsPI_count= (as.numeric(FallsPI_count)/nrow(rtCBSdf)*100))

  # Assign these counts to a variable
  Tremor_perc <- Tremor_perc[, 1]
  RestTremor_perc <- RestTremor_perc[, 1]
  LimbRigidity_perc <- LimbRigidity_perc[, 1]
  Slowness_perc <- Slowness_perc[, 1]
  Apraxia_perc <- Apraxia_perc[, 1]
  Gait_perc <- Gait_perc[, 1]
  FallsPI_perc <- FallsPI_perc[, 1]

  # Save each value in a list which grows with each iteration of the for-loop (just need to be mindful of the values you are entering here)
  List[[length(List)+1]] = c(Tremor_perc, RestTremor_perc, LimbRigidity_perc, Slowness_perc, Apraxia_perc, Gait_perc, FallsPI_perc)

} #end of loop

```

```

## [1] "aSyn-SAA positive"
## [1] "aSyn-SAA negative"

```

```

# To understand the structure of List: ##[[1]] is CBS asyn - [[2]] is CBS asyn + [[3]] is PSP asyn -[[4]] is PSP asyn +
print(List[[1]])

```

```
## [1] 28.571429 7.142857 50.000000 71.428571 64.285714 35.714286 57.142857
```

```
print(List[[2]])
```

```
## [1] 40 4 68 80 64 68 76
```

```
radar.rtCBSdf <- data.frame(row.names = c("aSyn-SAA negative", "aSyn-SAA positive"),
  Tremor = c(List[[1]][[1]], List[[2]][[1]]),
  Rest = c(List[[1]][[2]], List[[2]][[2]]),
  Limb = c(List[[1]][[3]], List[[2]][[3]]),
  Slowness = c(List[[1]][[4]], List[[2]][[4]]),
  Apraxia = c(List[[1]][[5]], List[[2]][[5]]),
  Gait = c(List[[1]][[6]], List[[2]][[6]]),
  Falls = c(List[[1]][[7]], List[[2]][[7]]))

max_mindf <- data.frame(Tremor = c(100, 0), Rest = c(100, 0), Limb = c(100, 0), Slowness =
= c(100, 0), Apraxia = c(100, 0), Gait = c(100, 0), Falls = c(100, 0))
rownames(max_mindf) <- c("Max", "Min")

# Bind the variable ranges to the data
radar.rtCBSdf <- rbind(max_mindf, radar.rtCBSdf)
radar.rtCBSdf
```

```
##          Tremor      Rest Limb Slowness Apraxia      Gait
## Max     100.00000 100.00000 100 100.00000 100.00000 100.00000
## Min      0.00000  0.00000   0  0.00000  0.00000  0.00000
## aSyn-SAA negative 28.57143 7.142857 50 71.42857 64.28571 35.71429
## aSyn-SAA positive 40.00000 4.000000 68 80.00000 64.00000 68.00000
##          Falls
## Max     100.00000
## Min      0.00000
## aSyn-SAA negative 57.14286
## aSyn-SAA positive 76.00000
```

```
# Rename some variables for presentation purposes
colnames(radar.rtCBSdf)[which(names(radar.rtCBSdf) == "Rest")] <- "Rest tremor"
colnames(radar.rtCBSdf)[which(names(radar.rtCBSdf) == "Limb")] <- "Limb rigidity"
colnames(radar.rtCBSdf)[which(names(radar.rtCBSdf) == "Falls")] <- "Falls & instability"
colnames(radar.rtCBSdf)[which(names(radar.rtCBSdf) == "Gait")] <- "Gait \n p<0.1"

# Create the radar charts
png(filename ="Fig1c_CBS.png")

create_beautiful_radarchart(data= radar.rtCBSdf, color= cbPalette_RTQUIC, vlcex=1, pltty=1, title="Motor symptoms in CBS")
    # Add an horizontal legend
    # legend(-0.65, -1.2, legend=c(expression(alpha*"Syn-SAA+"),expression(alpha*"Syn-SAA-")), horiz=TRUE, bty= "o", pch= 15 , col= cbPalette_RTQUIC, text.col= "black", cex = 1, pt.cex= 1.5)
```

PSP

```

List = list() #Create an empty list to which you will assign the values from the for-loop. By creating this list outside of the main for-loop, you allow for the data to be entered under different entries which means you can call all the values you need.

rt.value <- c("aSyn-SAA positive", "aSyn-SAA negative")
for (rt in rt.value) { #for-loop that tests each value of the variable

  rtPSPdf <- PSPdf[PSPdf$RTQUIC== rt, ] #Within subset of CBS for eg, look for subset of RT+

  print(rt)

#SUMMARIZE THE CATEGORICAL VARIABLES INTO ONE SINGLE COUNT OF "YES"
Tremor_perc <- rtPSPdf %>% summarise(Tremor_count= sum(Tremor_binary == "Yes")) %>% mutate(Tremor_count= (as.numeric(Tremor_count)/nrow(rtPSPdf)*100))
RestTremor_perc <- rtPSPdf %>% summarise(RestTremor_count= sum(RestTremor == "Yes")) %>% mutate(RestTremor_count= (as.numeric(RestTremor_count)/nrow(rtPSPdf)*100))
LimbRigidity_perc <- rtPSPdf %>% summarise(LimbRigidity_count= sum(LimbRigidity == "Yes")) %>% mutate(LimbRigidity_count= (as.numeric(LimbRigidity_count)/nrow(rtPSPdf)*100))
AxialRigidity_perc <- rtPSPdf %>% summarise(AxialRigidity_count= sum(AxialRigidity == "Yes")) %>% mutate(AxialRigidity_count= (as.numeric(AxialRigidity_count)/nrow(rtPSPdf)*100))
Slowness_perc <- rtPSPdf %>% summarise(Slowness_count= sum(Slowness_binary == "Yes")) %>% mutate(Slowness_count= (as.numeric(Slowness_count)/nrow(rtPSPdf)*100))
OM_perc <- rtPSPdf %>% summarise(OM_count= sum(VerticalOM == "Yes")) %>% mutate(OM_count = (as.numeric(OM_count)/nrow(rtPSPdf)*100))
Gait_perc <- rtPSPdf %>% summarise(Gait_count= sum(LP2_gait == "Yes")) %>% mutate(Gait_count= (as.numeric(Gait_count)/nrow(rtPSPdf)*100))
FallsPI_perc <- rtPSPdf %>% summarise(FallsPI_count= sum(LP2_falls_PI == "Yes")) %>% mutate(FallsPI_count= (as.numeric(FallsPI_count)/nrow(rtPSPdf)*100))

#Assign these counts to a variable
Tremor_perc <- Tremor_perc[, 1]
RestTremor_perc <- RestTremor_perc[, 1]
LimbRigidity_perc <- LimbRigidity_perc[, 1]
AxialRigidity_perc <- AxialRigidity_perc[, 1]
Slowness_perc <- Slowness_perc[, 1]
OM_perc <- OM_perc[, 1]
Gait_perc <- Gait_perc[, 1]
FallsPI_perc <- FallsPI_perc[, 1]

# #Save each value in a list which grows with each iteration of the for-loop (just need to be mindful of the values you are entering here)
List[[length(List)+1]] = c(Tremor_perc, RestTremor_perc, LimbRigidity_perc, AxialRigidity_perc, Slowness_perc, OM_perc, Gait_perc, FallsPI_perc)

} #end of loop

```

```

## [1] "aSyn-SAA positive"
## [1] "aSyn-SAA negative"

```

```
# To understand the structure of List: ##[[1]] is CBS asyn - [[2]] is CBS asyn + [[3]] is PSP asyn -[[4]] is PSP asyn +
print(List[[1]])
```

```
## [1] 62.5 12.5 62.5 87.5 87.5 87.5 87.5 100.0
```

```
print(List[[2]])
```

```
## [1] 45 10 85 90 95 95 90 95
```

```
radar.rtPSPdf <- data.frame(row.names = c("aSyn-SAA negative", "aSyn-SAA positive"),
  Tremor = c(List[[1]][[1]], List[[2]][[1]]),
  Rest = c(List[[1]][[2]], List[[2]][[2]]),
  Limb = c(List[[1]][[3]], List[[2]][[3]]),
  Axial = c(List[[1]][[4]], List[[2]][[4]]),
  Slowness = c(List[[1]][[5]], List[[2]][[5]]),
  OM = c(List[[1]][[6]], List[[2]][[6]]),
  Gait = c(List[[1]][[7]], List[[2]][[7]]),
  Falls = c(List[[1]][[8]], List[[2]][[8]]))

max_mindf <- data.frame(Tremor= c(100, 0), Rest= c(100, 0), Limb= c(100, 0), Axial=c(100, 0), Slowness = c(100, 0), OM= c(100, 0), Gait= c(100, 0), Falls= c(100, 0))
rownames(max_mindf) <- c("Max", "Min")

# Bind the variable ranges to the data
radar.rtPSPdf <- rbind(max_mindf, radar.rtPSPdf)
radar.rtPSPdf
```

	Tremor	Rest	Limb	Axial	Slowness	OM	Gait	Falls
## Max	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100
## Min	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
## aSyn-SAA negative	62.5	12.5	62.5	87.5	87.5	87.5	87.5	100
## aSyn-SAA positive	45.0	10.0	85.0	90.0	95.0	95.0	90.0	95

```
#Rename some variables for presentation purposes
colnames(radar.rtPSPdf)[which(names(radar.rtPSPdf) == "Rest")] <- "Rest tremor"
colnames(radar.rtPSPdf)[which(names(radar.rtPSPdf) == "Limb")] <- "Limb rigidity \n p<0.
1"
colnames(radar.rtPSPdf)[which(names(radar.rtPSPdf) == "Axial")] <- "Axial rigidity"
colnames(radar.rtPSPdf)[which(names(radar.rtPSPdf) == "OM")] <- "Oculomotor"
colnames(radar.rtPSPdf)[which(names(radar.rtPSPdf) == "Falls")] <- "Falls & instability"

# Create the radar charts
png(filename ="Fig1c_PSP.png")

create_beautiful_radarchart(data= radar.rtPSPdf, color= cbPalette_RTQUIC, vlcex=1, plty=
1, title="Motor symptoms in PSP")
    # Add an horizontal legend
    # legend(-0.65, -1.2, legend=c(expression(alpha*"Syn-SAA+"),expression(alpha*"Sy
n-SAA-")), horiz=TRUE, bty= "o", pch= 15 , col= cbPalette_RTQUIC, text.col= "black", cex
= 1, pt.cex= 1.5)

dev.off()#Not needed?
```

```
## quartz_off_screen
##                 2
```

9. BINARY LOGISTIC REGRESSION

9.1. BLR STATS: LOGISTIC REGRESSION MODEL SELECTION

Model is selected based on previous analyses (logabeta*Onset) + the simple comparisons in Supp material (Gait/RBD_binary). Model was run with and without AD/DX_APD status and all values remained very similar.

```
dfblr<- df[!which(is.na(df$logNFL)), ] #Remove NAs in df for the predictor NfL (anyway w
ould have been automatically excluded)

blr <- glm(RTQUIC_BLR ~ DX_APD + scale(Onset_age)*scale(logabeta) + RBD_binary + LP2_gai
t + scale(logNFL), data= dfblr, family = "binomial"(logit))
summary(blr)
```

```

## 
## Call:
## glm(formula = RTQUIC_BLR ~ DX_APD + scale(Onset_age) * scale(logabeta) +
##       RBD_binary + LP2_gait + scale(logNFL), family = binomial(logit),
##       data = dfblr)
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.2373   0.6490   0.366  0.71459
## DX_APDPSP                  -1.4575   0.9908  -1.471  0.14128
## scale(Onset_age)             1.5575   0.6583   2.366  0.01799 *
## scale(logabeta)              0.7127   0.5850   1.218  0.22304
## RBD_binaryYes                5.1058   1.6350   3.123  0.00179 **
## LP2_gaitYes                 -2.7869   1.0824  -2.575  0.01003 *
## scale(logNFL)                -0.7345   0.4956  -1.482  0.13831
## scale(Onset_age):scale(logabeta) 1.0211   0.5067   2.015  0.04387 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 81.792 on 64 degrees of freedom
## Residual deviance: 49.759 on 57 degrees of freedom
## AIC: 65.759
##
## Number of Fisher Scoring iterations: 6

```

AIC(blr) #lowest AIC but Dx and NFL are not significant. So should consider excluding them.

```
## [1] 65.75874
```

```

blr2 <- glm(RTQUIC_BLR ~ DX_APD + scale(Onset_age) + scale(logabeta) + RBD_binary + LP2_gait + scale(logNFL), data= dfblr, family = "binomial"(logit))
AIC(blr2) #Slightly better to keep interaction term

```

```
## [1] 68.55218
```

```

blr3 <- glm(RTQUIC_BLR ~  scale(Onset_age)*scale(logabeta) + RBD_binary + LP2_gait + scale(logNFL), data= dfblr, family = "binomial"(logit))
AIC(blr3) #Slightly better with DX

```

```
## [1] 66.2534
```

```

blr4 <- glm(RTQUIC_BLR ~  DX_APD + scale(Onset_age)*scale(logabeta) + RBD_binary + LP2_gait, data= dfblr, family = "binomial"(logit))
AIC(blr4) #Slightly better with DX

```

```
## [1] 66.5125
```

```
blr5 <- glm(RTQUIC_BLR ~ DX_APD + scale(Onset_age)*scale(logabeta) + RBD_binary + LP2_gait, data= dfblr, family = "binomial"(logit))
AIC(blr5) #Slightly better with NfL
```

```
## [1] 66.5125
```

```
blr6 <- glm(RTQUIC_BLR ~ DX_APD + scale(Onset_age)*scale(logabeta) + RBD_binary + scale(logNFL), data= dfblr, family = "binomial"(logit))
AIC(blr6) #Better with gait
```

```
## [1] 72.87027
```

```
blr7 <- glm(RTQUIC_BLR ~ DX_APD + scale(Onset_age)*scale(logabeta) + LP2_gait + scale(logNFL), data= dfblr, family = "binomial"(logit))
AIC(blr7) #Better with RBD
```

```
## [1] 79.97648
```

IMPORTANT NOTE: As mentioned in Results, we know that the generalizability of this model is likely to be limited due to the relatively low sample size considering the number of predictors. Therefore, the model is mostly meant to give additional information. For ex, RBD is a rare symptom so it would be very unlikely that the OR would remain stable across multiple iterations of the model. Similarly, we preferred to keep outliers for the run, but below the model without these outliers is also shown.

9.2. BLR STATS: LOGISTIC REGRESSION MODEL DIAGNOSTICS

```
# MCFADDEN INDEX
pscl:::pR2(blr) ["McFadden"]
```

```
## fitting null model for pseudo-r2
```

```
## McFadden
## 0.3916408
```

```
# COLLINEARITY CHECK WITH VARIABLE INFLATION FACTOR
# First look at Onset_age and logged Abeta42 separately due to interaction
blronset <- glm(RTQUIC_BLR ~ DX_APD + scale(Onset_age) + RBD_binary + LP2_gait + scale(logNFL), data= dfblr, family = "binomial") #Check individual relationship of Onset with RTQUIC
car::vif(blronset)
```

```
##          DX_APD scale(Onset_age)      RBD_binary      LP2_gait
##        1.347620       2.441283       1.503927       2.124081
##      scale(logNFL)
##        1.068010
```

```
blrabeta <- glm(RTQUIC_BLR ~ DX_APD + scale(logabeta) + RBD_binary + LP2_gait + scale(logNFL), data= dfblr, family = "binomial") #Check individual relationship of Abeta with RTQUIC
car::vif(blrabeta)
```

	DX_APD	scale(logabeta)	RBD_binary	LP2_gait	scale(logNFL)
##	1.233808	1.569275	1.247782	1.164635	1.289880

```
blr_nointerac <- glm(RTQUIC_BLR ~ DX_APD + scale(Onset_age)+ scale(logabeta) + RBD_binary + LP2_gait + scale(logNFL), data= dfblr, family = "binomial") #Check individual relationship of Abeta with RTQUIC
car::vif(blr_nointerac)
```

	DX_APD	scale(Onset_age)	scale(logabeta)	RBD_binary
##	1.309351	2.522389	1.749046	1.587635
##	LP2_gait	scale(logNFL)		
##	2.057364	1.370489		

```
# With interaction and scale
car::vif(blr)
```

```
## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif
```

	DX_APD	scale(Onset_age)
##	1.627432	3.128605
##	scale(logabeta)	RBD_binary
##	2.676131	1.734627
##	LP2_gait	scale(logNFL)
##	2.290486	1.874543
## scale(Onset_age):scale(logabeta)		
##	1.568962	

```
# AUTOCORRELATION RESIDUALS
durbinWatsonTest(blr)
```

```
## lag Autocorrelation D-W Statistic p-value
##    1     -0.0592418     2.116274   0.774
## Alternative hypothesis: rho != 0
```

```
# Variable importance
caret::varImp(blr) #Similar, so all variables are of comparable importance in this mode
l.
```

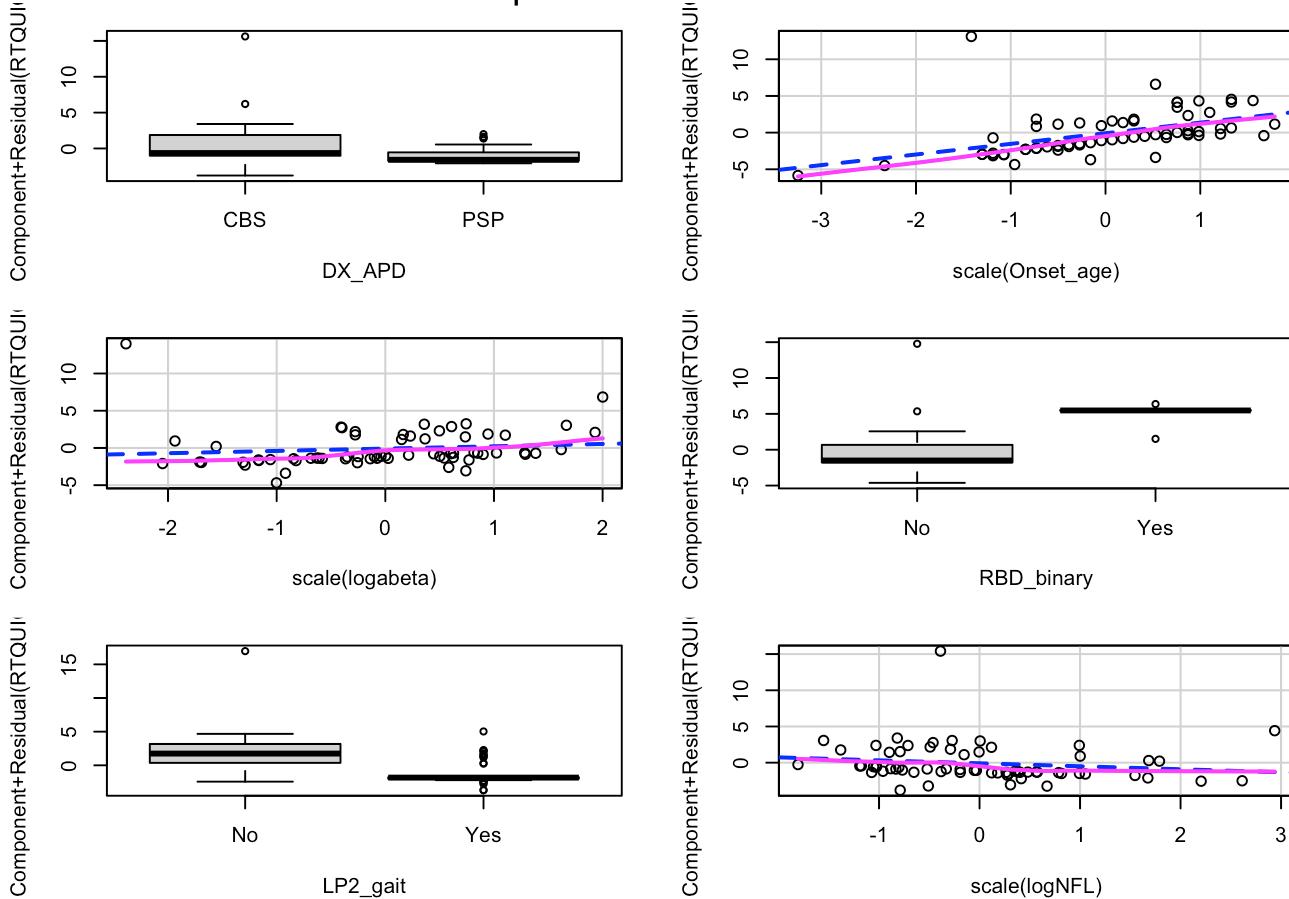
```
## Overall
## DX_APDPSP 1.471025
## scale(Onset_age) 2.365896
## scale(logabeta) 1.218474
## RBD_binaryYes 3.122778
## LP2_gaitYes 2.574706
## scale(logNFL) 1.482103
## scale(Onset_age):scale(logabeta) 2.015318
```

LINEARITY BETWEEN LOGIT AND IVS

In logistic regression, we assume the relationship is linear on the logit scale. This is assessed with component-plus-residual plots. The “component” is the values of a variable multiplied by its estimated coefficient (meaning that each predictor has its own component vector), and the “residual” is the working residuals, a type of residuals in generalized linear models. https://sscc.wisc.edu/sscc/pubs/RegDiag-R/logistic-regression.html#log_lin

```
crPlots(blr2) #without the interaction
```

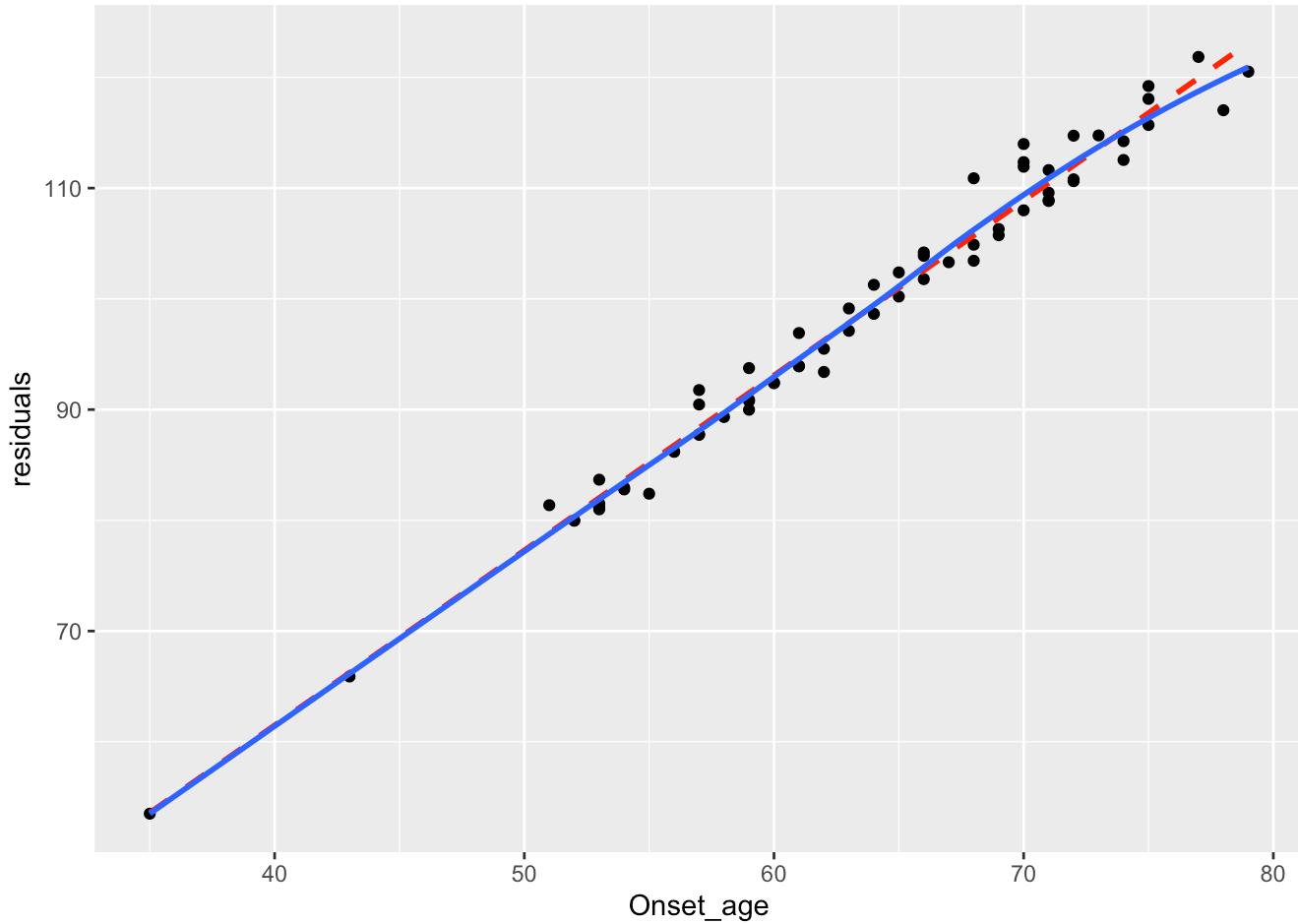
Component + Residual Plots



```
dfblr |> #Good
  mutate(residuals = coef(blr)[3]*Onset_age + residuals(blr, type="working")) |>
  ggplot(aes(x = Onset_age, y = residuals)) +
  geom_point() +
  geom_smooth(color = "red", method = "lm", linetype = 2, se = F) +
  geom_smooth(se = F)
```

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

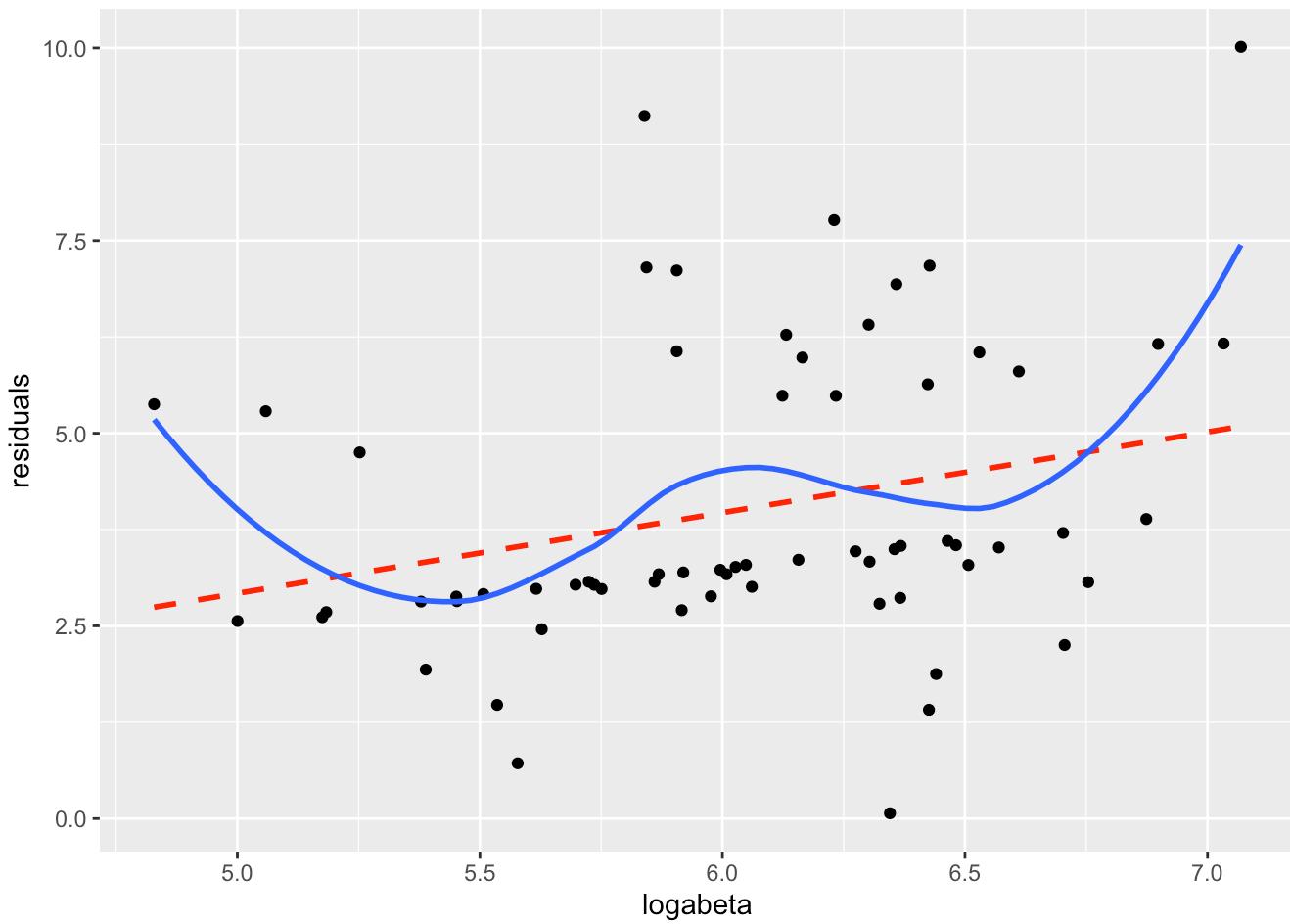
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
dfblr |> #Not linear
  mutate(residuals = coef(blr)[4]*logabeta + residuals(blr, type="working")) |>
  ggplot(aes(x = logabeta, y = residuals)) +
  geom_point() +
  geom_smooth(color = "red", method = "lm", linetype = 2, se = F) +
  geom_smooth(se = F)
```

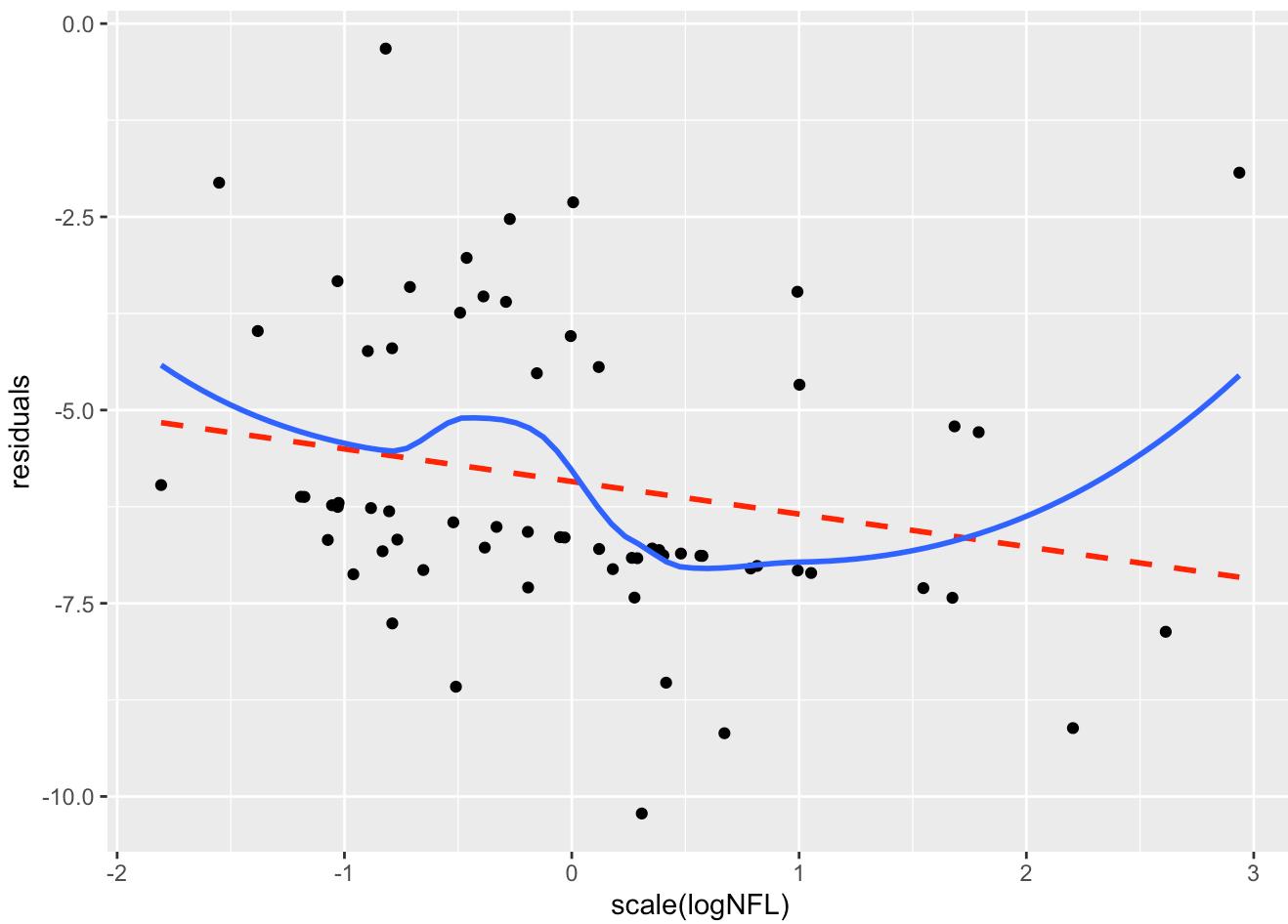
```
## `geom_smooth()` using formula = 'y ~ x'
```

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



```
dfblr |> #Not linear
  mutate(residuals = coef(blr)[7]*logNFL + residuals(blr, type="working")) |>
  ggplot(aes(x = scale(logNFL), y = residuals)) +
  geom_point() +
  geom_smooth(color = "red", method = "lm", linetype = 2, se = F) +
  geom_smooth(se = F)
```

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



These plots indicate that there are important outliers. Since this analysis is just exploratory and underpowered to begin with, the results below are presented for informative purposes to evaluate the robustness of findings.

9.3. MODEL SELECTION CONTINUED - TESTING OTHER APPROACHES

As noted in the manuscript, the binary logistic regression modeling is underpowered due to the heterogeneity of the sample.

9.3.1. APPROACH 1: REMOVING ABETA OUTLIERS

```
dfabetablr <- dfabeta[!(dfabeta$logabeta < 5), ] #Remove the low Abeta outlier
dfabetablr<- dfabetablr[-which(is.na(dfabetablr$logNFL)), ] #Remove NAs in df for the predictor NFL (anyway would have been automatically excluded)

blrtest <- glm(RTQUIC_BLR ~ DX_APD + scale(Onset_age)*scale(logabeta) + RBD_binary + LP2_gait + scale(logNFL), data= dfabetablr, family = "binomial"(logit))
summary(blrtest)
```

```

## 
## Call:
## glm(formula = RTQUIC_BLR ~ DX_APD + scale(Onset_age) * scale(logabeta) +
##       RBD_binary + LP2_gait + scale(logNFL), family = binomial(logit),
##       data = dfabetablr)
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                0.2304    0.7137   0.323  0.74679
## DX_APDPSP                 -0.8659    1.0131  -0.855  0.39273
## scale(Onset_age)            2.0910    0.8201   2.550  0.01078 *
## scale(logabeta)             0.9847    0.6890   1.429  0.15296
## RBD_binaryYes              7.0648    2.3168   3.049  0.00229 **
## LP2_gaitYes                -3.5014    1.3505  -2.593  0.00953 **
## scale(logNFL)               -1.3361    0.6520  -2.049  0.04043 *
## scale(Onset_age):scale(logabeta) 0.3437    0.6383   0.538  0.59029
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 77.138 on 62 degrees of freedom
## Residual deviance: 41.885 on 55 degrees of freedom
## AIC: 57.885
##
## Number of Fisher Scoring iterations: 6

```

AIC(blrtest) #lowest AIC but Dx and NFL are not significant. So should consider excluding them.

```
## [1] 57.8853
```

```

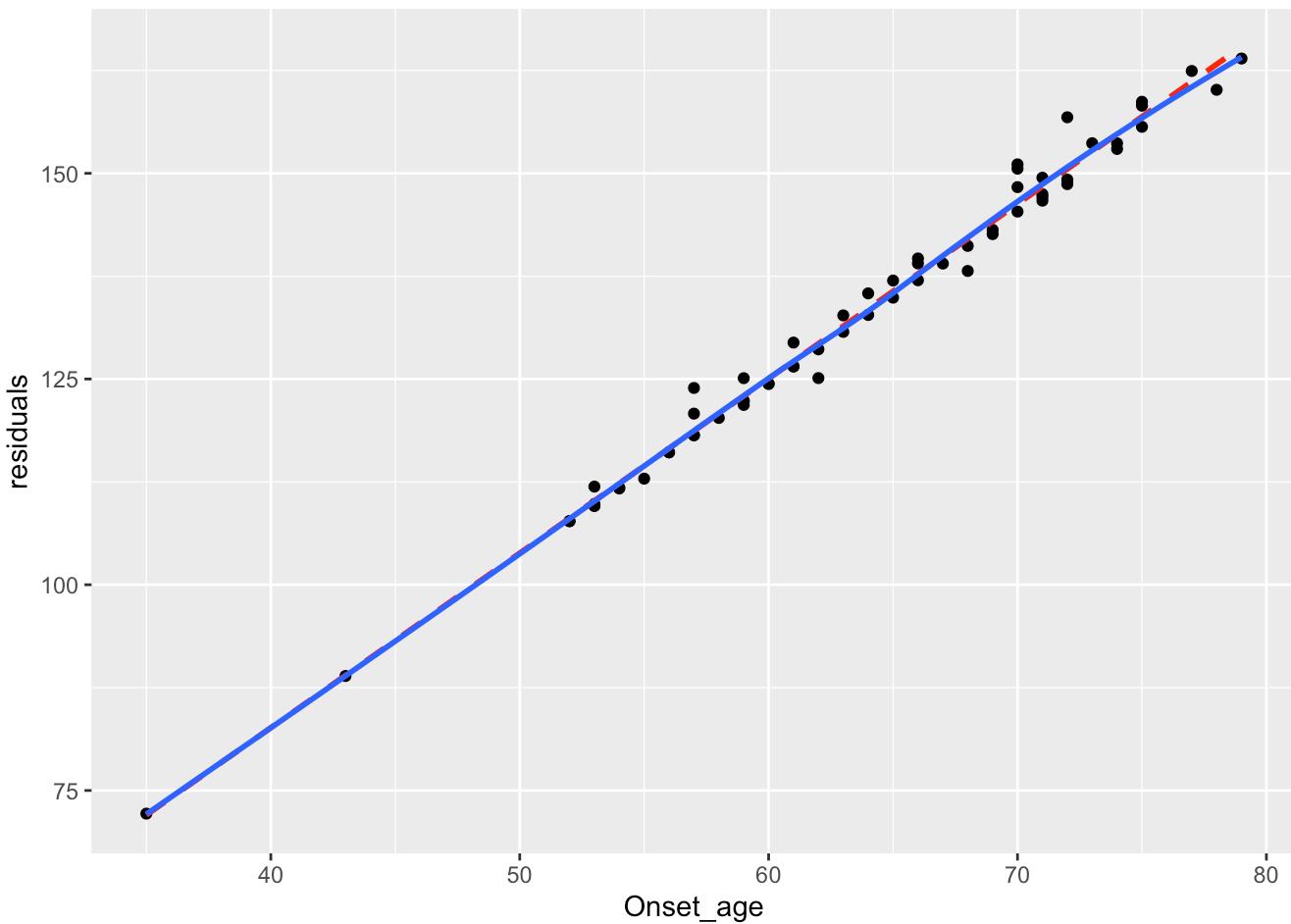
dfabetablr |>
  mutate(residuals = coef(blrtest)[3]*Onset_age + residuals(blrtest, type="working"))
|>
  ggplot(aes(x = Onset_age, y = residuals)) +
  geom_point() +
  geom_smooth(color = "red", method = "lm", linetype = 2, se = F) +
  geom_smooth(se = F)

```

```

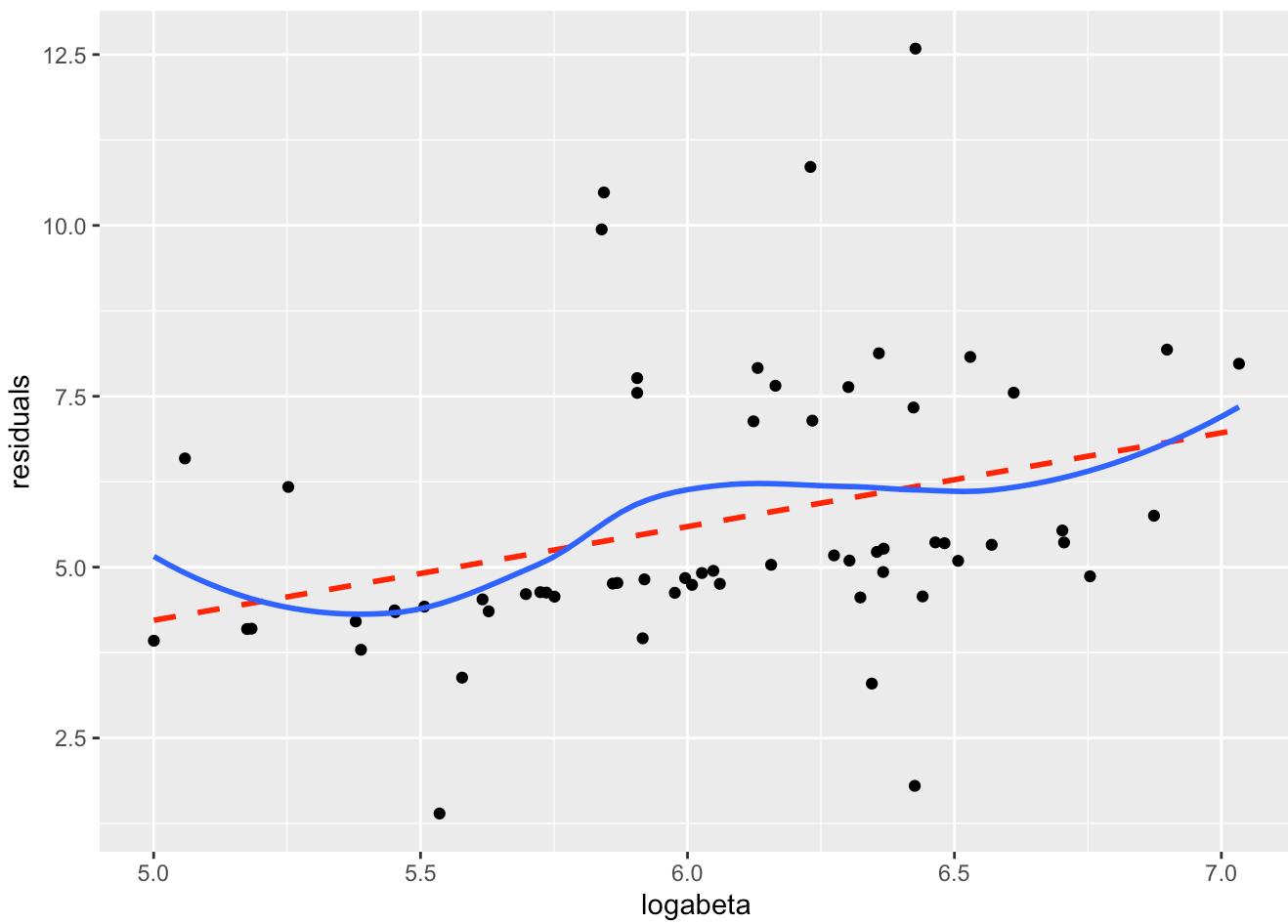
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'

```



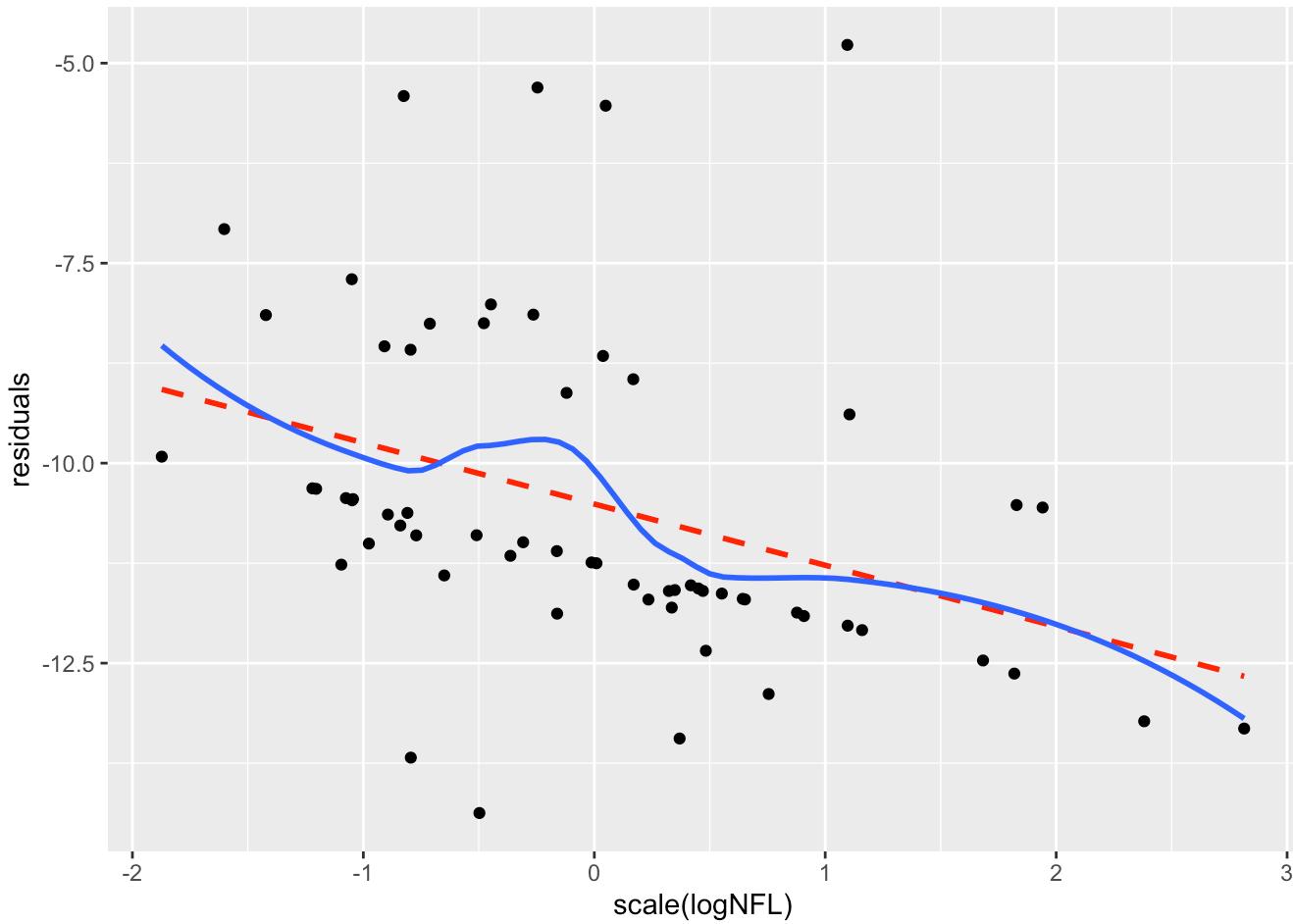
```
dfabetablr |>
  mutate(residuals = coef(blrtest)[4]*logabeta + residuals(blrtest, type="working")) |>
  ggplot(aes(x = logabeta, y = residuals)) +
  geom_point() +
  geom_smooth(color = "red", method = "lm", linetype = 2, se = F) +
  geom_smooth(se = F)
```

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



```
dfabetablr |>
  mutate(residuals = coef(blrtest)[7]*logNFL + residuals(blrtest, type="working")) |>
  ggplot(aes(x = scale(logNFL), y = residuals)) +
  geom_point() +
  geom_smooth(color = "red", method = "lm", linetype = 2, se = F) +
  geom_smooth(se = F)
```

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



```
removed <- setdiff(dfblr, dfabetablr)
removed[, c("RTQUIC", "AD", "Early_onset", "logabeta")]
```

```
##           RTQUIC          AD Early_onset logabeta
## 1 aSyn-SAA positive AD Positive Late-onset 7.068870
## 2 aSyn-SAA positive AD Positive Young-onset 4.828314
```

```
cat("The following IDs are removed based on Abeta42 values for the corrected logistic regression model with better diagnostics: ", removed$logabeta, "\n")
```

```
## The following IDs are removed based on Abeta42 values for the corrected logistic regression model with better diagnostics: 7.06887 4.828314
```

The issue is, both are aSyn-SAA+ (bringing down total number of aSyn-SAA+ to 20 instead of 22 (aSyn-SAA+ is the outcome measure for this model), but 1 especially is one of the 8 aSyn-SAA+/young-onset, 7 aSyn-SAA+/AD+, and 5 aSyn-SAA+/AD+. We already know from Fig 1A that these combinations are rare but they are of interest in our cohort due to 1. hypothesis 2. results of frequency analyses (categorical tests on AD+ vs Young-onset, so removing a subject that meets all these criteria would in itself be an issue. Removing extreme Abeta42 values (even not identified by Tukey method) improves assumption testing output. It removes important information from the model though as it reduces significantly groups that were already underrepresented but of interest. In this iteration, Onset age, Gait, RBD, and NfL are significantly predictors of aSyn-SAA+, but not Abeta 42.

9.3.2. APPROACH 2: REMOVING VARIABLES/SIMPLIFYING THE MODEL

```
# NFL and DX can be removed as they are not significant in blr model. However: DX is clinically really important hence its inclusion.
# NFL is significant in some iterations of the model, such as the one above - it could be a significant covariate/confounding factor
# that "competes" with Abeta.
blrtest2 <- glm(RTQUIC_BLR ~ scale(Onset_age)*scale(logabeta) + RBD_binary + LP2_gait, data= dfblr, family = "binomial"(logit))
summary(blrtest2)
```

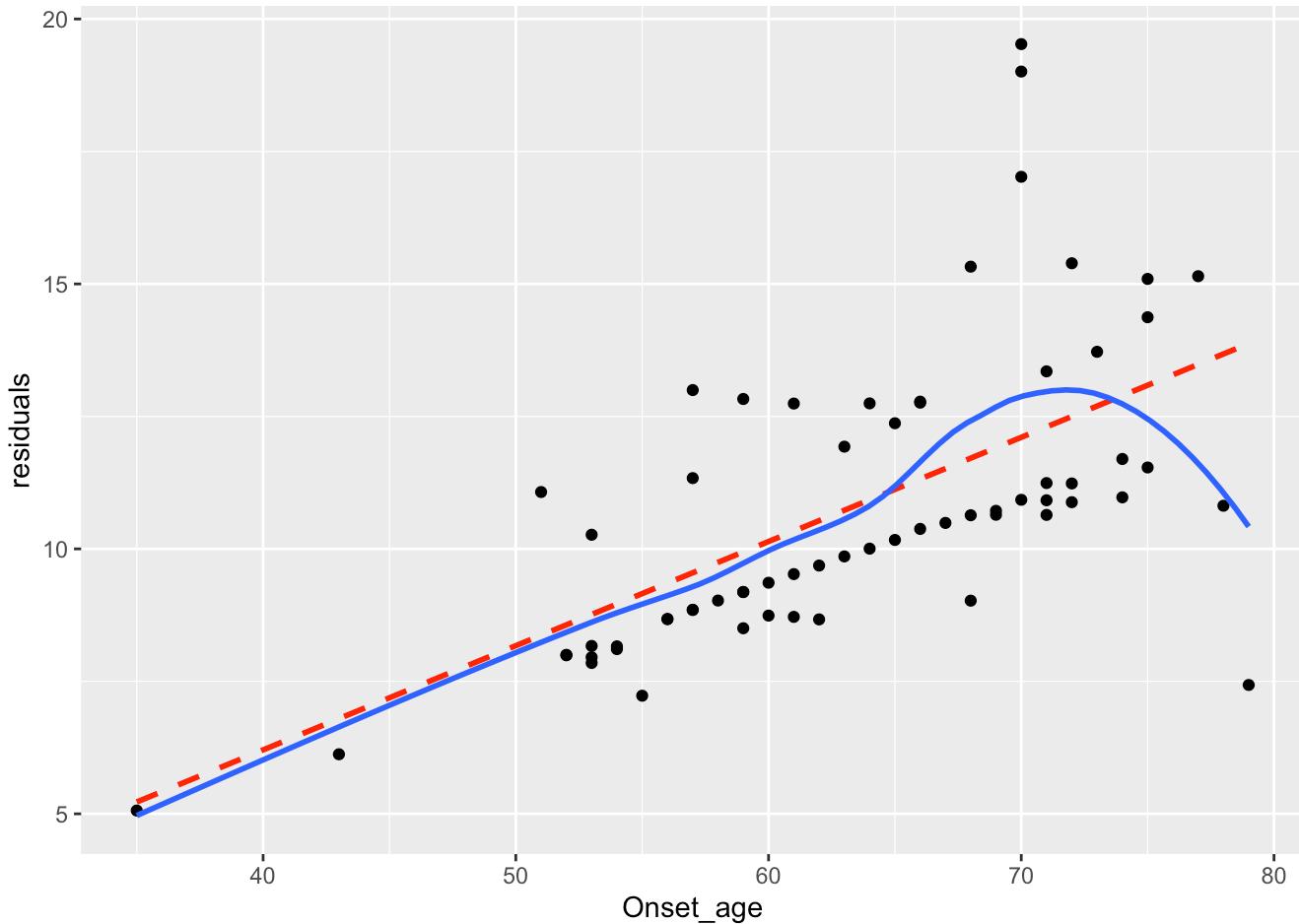
```
##
## Call:
## glm(formula = RTQUIC_BLR ~ scale(Onset_age) * scale(logabeta) +
##     RBD_binary + LP2_gait, family = binomial(logit), data = dfblr)
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.2561    0.5874   0.436  0.66288
## scale(Onset_age)            1.5387    0.6084   2.529  0.01144 *
## scale(logabeta)             0.1732    0.4037   0.429  0.66792
## RBD_binaryYes              4.0981    1.4490   2.828  0.00468 **
## LP2_gaitYes                -3.0127   1.0204  -2.952  0.00315 **
## scale(Onset_age):scale(logabeta) 0.6276    0.3666   1.712  0.08690 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 81.792 on 64 degrees of freedom
## Residual deviance: 54.768 on 59 degrees of freedom
## AIC: 66.768
##
## Number of Fisher Scoring iterations: 6
```

AIC(blrtest2) #lowest AIC but Dx and NFL are not significant. So should consider excluding them.

```
## [1] 66.76798
```

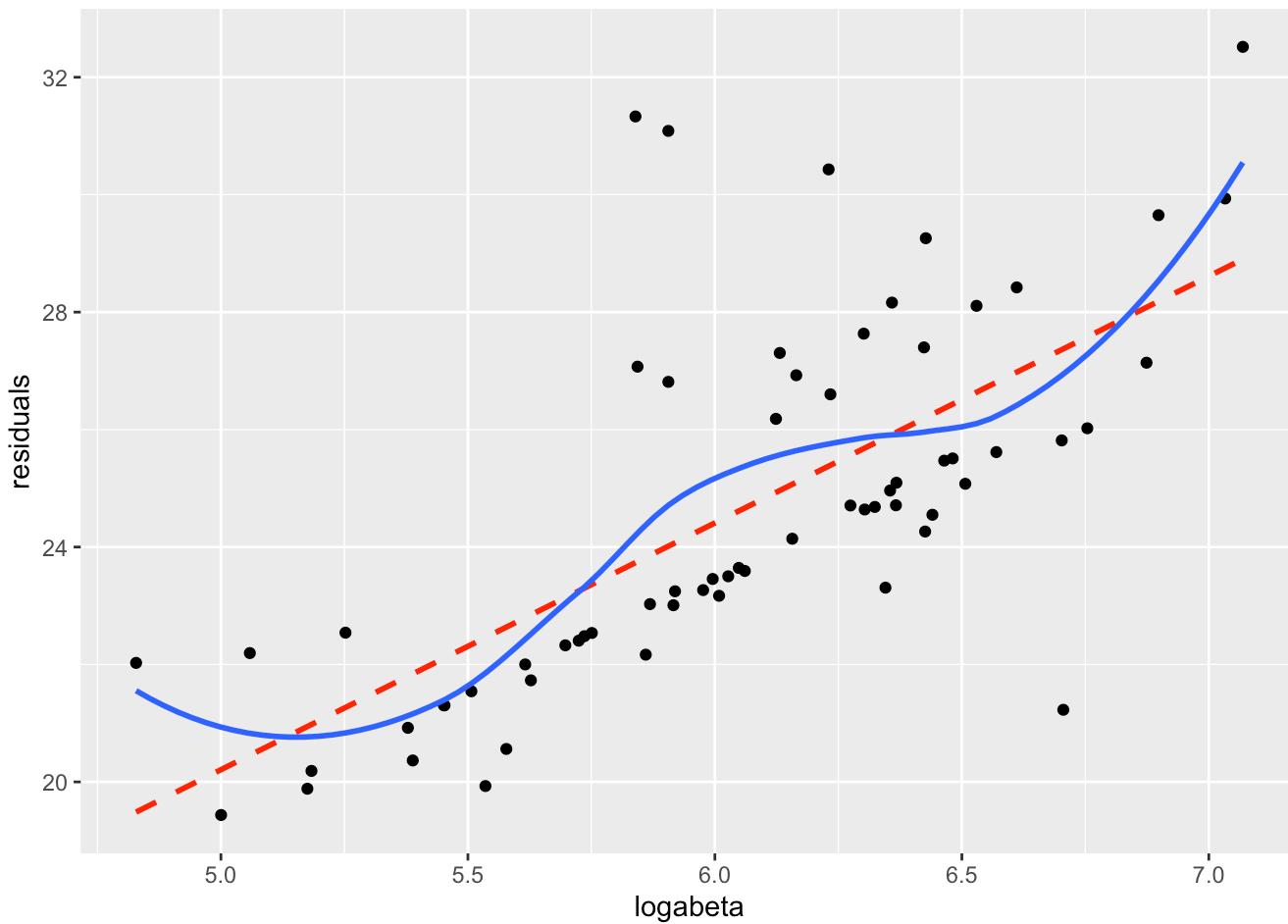
```
dfblr |>
  mutate(residuals = coef(blrtest2)[3]*Onset_age + residuals(blrtest2, type="working")) |>
  ggplot(aes(x = Onset_age, y = residuals)) +
  geom_point() +
  geom_smooth(color = "red", method = "lm", linetype = 2, se = F) +
  geom_smooth(se = F)
```

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



```
dfblr |>
  mutate(residuals = coef(blrtest2)[4]*logabeta + residuals(blrtest2, type="working"))
|>
  ggplot(aes(x = logabeta, y = residuals)) +
  geom_point() +
  geom_smooth(color = "red", method = "lm", linetype = 2, se = F) +
  geom_smooth(se = F)
```

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



Removing variables overall improves assumption testing output, except at very high age at onset(>80). In this iteration, Onset age, Gait, and RBD are significantly predictors of aSyn-SAA+. Onset age by Abeta42 is significant at a 90% confidence level. It is the best compromise between reducing violation of assumptions and representativity of the sample. It also reduces the issue of overfitting.

Conclusion: The logit linearity assumption was violated so we examined multiple possibilities. Transforming data was not appropriate due to the complexity of the model. Removing outliers was promising based on graph: Removing NfL outliers (not shown) fixed the assumption for NfL but not Abeta42. Removing outliers of Abeta42 based on Tukey method (eg using dfbeta dataset) did not fix the assumption for either variable (not shown). Removing the lowest value of Abeta42 (<5 logged Abeta42 value) in addition to the positive outliers of Abeta42 (not of NfL) reduced violation of the linearity assumption. However, the low Abeta42 subject is not an outlier based on Tukey method (ie not unreasonably low Abeta42 compared to other AD+/diagnostic group/RT-QUIC group). On the other hand, this low Abeta42 subject is one of few subjects who are RTQUIC+ (rarer outcome, moreover outcome to be predicted), AD+ (<20% of whole dataset), and young-onset RTQUIC+ (8 people total, of which 5 are AD+). The last approach tested was removing variables that were not significant or of interest from the final model. In that model, the whole dataset could be including without violating the linearity assumption (except at very high age at onset, which is easier to interpret).

For the reasons above, no further efforts towards prediction are performed. O.R. will be given for scale. Visualization of the model is kept here for interpretation but not included in the manuscript. Influence of outliers should be highlighted, but to remove them entirely would also be inappropriate. Consistently significant effect of: Age at onset; Gait; and RBD.

9.4. MODEL COEFFICIENTS (FOR SCALE ONLY)

```
coef(blr)
```

```

##                               (Intercept)                               DX_APDPSP
##                               0.2373303                            -1.4575350
## scale(Onset_age)                               scale(logabeta)
##                               1.5575150                             0.7127483
## RBD_binaryYes                               LP2_gaitYes
##                               5.1058142                            -2.7869085
## scale(logNFL) scale(Onset_age):scale(logabeta)
##                               -0.7344995                           1.0210788
##
```

```
exp(coef(blr))
```

```

##                               (Intercept)                               DX_APDPSP
##                               1.26785980                            0.23280943
## scale(Onset_age)                               scale(logabeta)
##                               4.74701012                            2.03958891
## RBD_binaryYes                               LP2_gaitYes
##                               164.97833640                           0.06161139
## scale(logNFL) scale(Onset_age):scale(logabeta)
##                               0.47974549                           2.77618822
##
```

`cbind(coef(blr),odds_ratio=exp(coef(blr)),exp(confint(blr, level=0.95)))` #it says 2.5% and 97.5% because these are the two borders to end up with the cnetral 95% of your distribution

```
## Waiting for profiling to be done...
```

	odds_ratio	2.5 %
## (Intercept)	0.2373303	1.26785980 0.352389256
## DX_APDPSP	-1.4575350	0.23280943 0.025719856
## scale(Onset_age)	1.5575150	4.74701012 1.505991556
## scale(logabeta)	0.7127483	2.03958891 0.717141165
## RBD_binaryYes	5.1058142	164.97833640 10.502297767
## LP2_gaitYes	-2.7869085	0.06161139 0.005408024
## scale(logNFL)	-0.7344995	0.47974549 0.153426284
## scale(Onset_age):scale(logabeta)	1.0210788	2.77618822 1.109929368
		97.5 %
## (Intercept)	4.8004636	
## DX_APDPSP	1.3915122	
## scale(Onset_age)	21.0904655	
## scale(logabeta)	7.4954446	
## RBD_binaryYes	8048.0626595	
## LP2_gaitYes	0.4092119	
## scale(logNFL)	1.1280402	
## scale(Onset_age):scale(logabeta)	8.8301725	

```
coef(blrtest2)
```

```

##          (Intercept)           scale(Onset_age)
##            0.2560521            1.5387101
##      scale(logabeta)           RBD_binaryYes
##            0.1732093            4.0981456
## LP2_gaitYes scale(Onset_age):scale(logabeta)
##            -3.0126710            0.6276367

```

```
exp(coef(blrtest2))
```

```

##          (Intercept)           scale(Onset_age)
##            1.2918200            4.6585773
##      scale(logabeta)           RBD_binaryYes
##            1.1891150            60.2284990
## LP2_gaitYes scale(Onset_age):scale(logabeta)
##            0.0491602            1.8731784

```

```
cbind(coef(blrtest2), odds_ratio=exp(coef(blrtest2)), exp(confint(blrtest2, level=0.95)))
#it says 2.5% and 97.5% because these are the two borders to end up with the cnetal 95%
of your distribution
```

```
## Waiting for profiling to be done...
```

	odds_ratio	2.5 %	97.5 %
## (Intercept)	0.2560521	1.2918200	0.411869833
## scale(Onset_age)	1.5387101	4.6585773	1.602984464
## scale(logabeta)	0.1732093	1.1891150	0.543441718
## RBD_binaryYes	4.0981456	60.2284990	5.233289945
## LP2_gaitYes	-3.0126710	0.0491602	0.005025569
## scale(Onset_age):scale(logabeta)	0.6276367	1.8731784	0.909549983

Example of interpretation: $\exp(4.88278)$ #RBD. 81.60169. Odds of someone with RBD being RTquic+ increases by 8000%. IE x80. $\exp(-2.85663)$ #Gait. 0.09253855. Odds of someone with gait issues being RTQUIC+ decreases by 10% compared to someone without. $\exp(0.15122)$ #Age: 1.16. #The value indicates that as age increase by one more unit, then the odds of being SAA+ increases by 16%

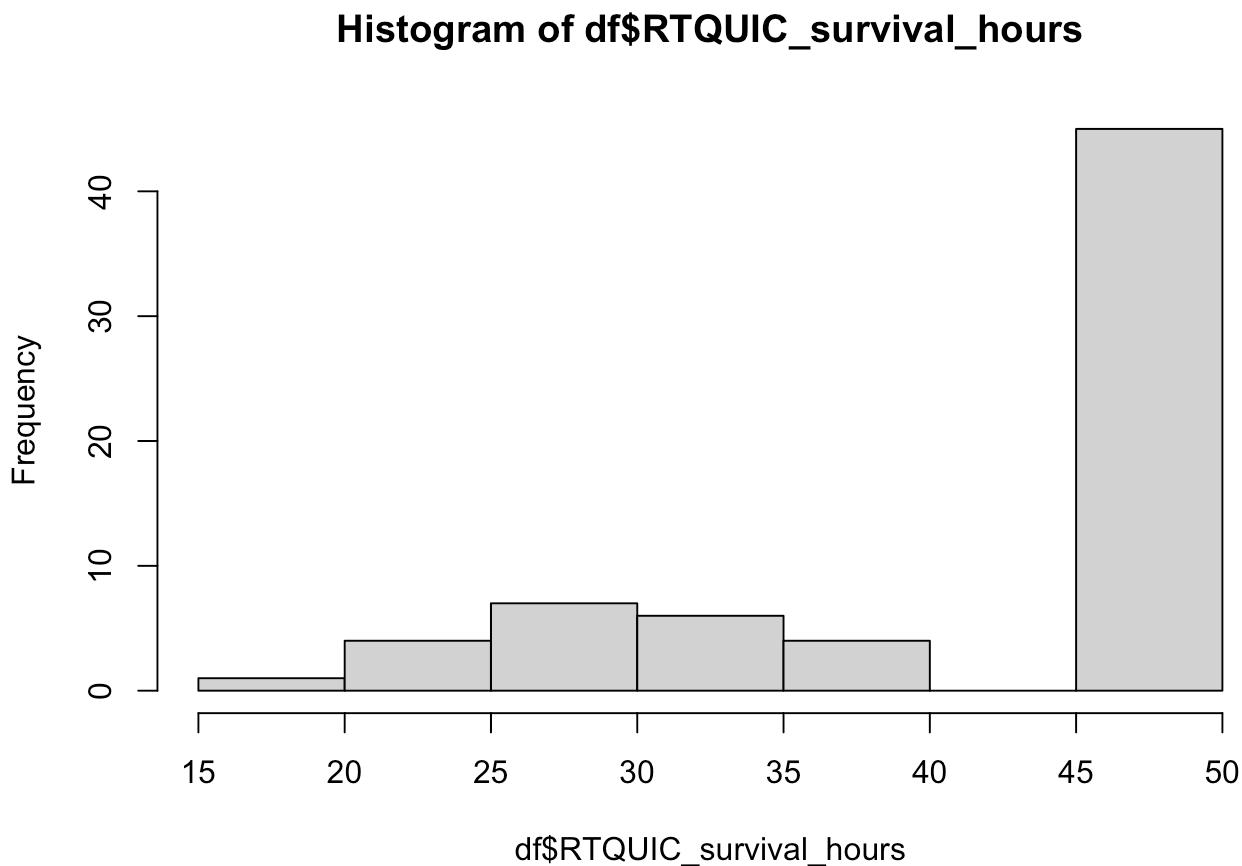
10. RTQUIC PARAMETERS SUPP ANALYSES

10.1. RTQUIC PARAMETERS SUPP: LAG

Lag hours is the #hours required to reach max ThT. It makes the most sense to think about it as data suited for survival analysis. For that purpose, we are censoring the subjects who never reached positivity (RTQUIC negative) as max ThT is not something that exists in this case they plateau early on in the analysis as their curve never rises - so comparign them to RT-QUIC+ subjects would not make sense.

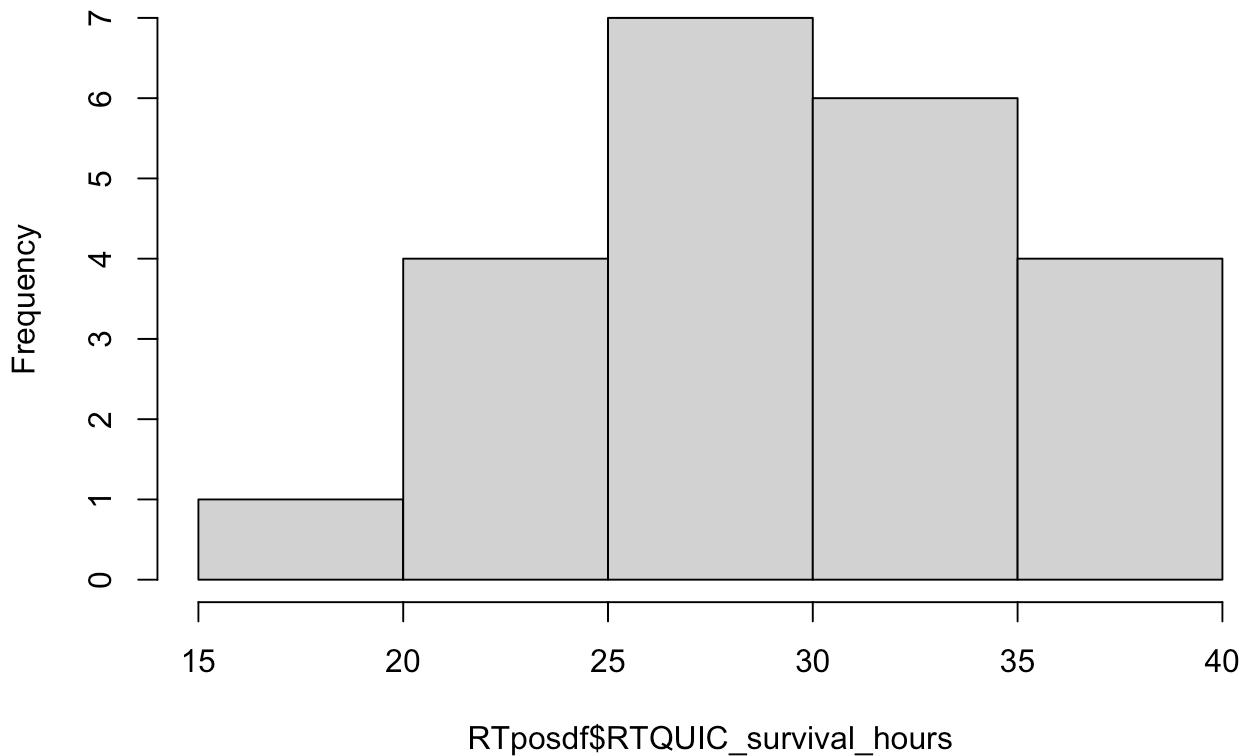
LAG STATISTICS: DISTRIBUTION

```
hist(df$RTQUIC_survival_hours)
```



```
hist(RTposdf$RTQUIC_survival_hours)
```

Histogram of RTposdf\$RTQUIC_survival_hours



```
shapiro.test(df[RTposdf$DX_APD == "CBS", ]$RTQUIC_survival_hours) #nonnormal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: df[RTposdf$DX_APD == "CBS", ]$RTQUIC_survival_hours  
## W = 0.73185, p-value = 2.072e-07
```

```
shapiro.test(df[RTposdf$DX_APD == "PSP", ]$RTQUIC_survival_hours) #nonnormal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: df[RTposdf$DX_APD == "PSP", ]$RTQUIC_survival_hours  
## W = 0.52728, p-value = 7.026e-08
```

```
shapiro.test(df[RTposdf$Early_onset == "Young-onset", ]$RTQUIC_survival_hours) #nonnormal
```

```
## 
## Shapiro-Wilk normality test
##
## data: df[RTposdf$Early_onset == "Young-onset", ]$RTQUIC_survival_hours
## W = 0.80118, p-value = 0.000308
```

```
shapiro.test(df[RTposdf$Early_onset == "Late-onset", ]$RTQUIC_survival_hours) #nonnormal
```

```
## 
## Shapiro-Wilk normality test
##
## data: df[RTposdf$Early_onset == "Late-onset", ]$RTQUIC_survival_hours
## W = 0.56702, p-value = 4.661e-10
```

```
shapiro.test(df[RTposdf$AD == "AD Positive", ]$RTQUIC_survival_hours) #nonnormal
```

```
## 
## Shapiro-Wilk normality test
##
## data: df[RTposdf$AD == "AD Positive", ]$RTQUIC_survival_hours
## W = 0.8159, p-value = 0.00116
```

```
shapiro.test(df[RTposdf$AD == "AD Negative", ]$RTQUIC_survival_hours) #nonnormal
```

```
## 
## Shapiro-Wilk normality test
##
## data: df[RTposdf$AD == "AD Negative", ]$RTQUIC_survival_hours
## W = 0.57171, p-value = 2.279e-10
```

```
leveneTest(RTQUIC_survival_hours ~ DX_APD, data = RTposdf) #homoscedasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value Pr(>F)
## group    1  0.6338 0.4353
##          20
```

```
leveneTest(RTQUIC_survival_hours ~ Early_onset, data = RTposdf) #homoscedasticity
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.3208 0.5775
##       20
```

```
leveneTest(RTQUIC_survival_hours ~ AD, data = RTposdf) #homoscedasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.2545 0.6194
##       20
```

LAG STATISTICS: SUMMARY

```
RTposdf %>% summarize(count=n(), format(round(median(RTQUIC_survival_hours, na.rm=T),2),
2), IQR=IQR(RTQUIC_survival_hours, na.rm=T), min=min(RTQUIC_survival_hours, na.rm=T), ma
x=max(RTQUIC_survival_hours, na.rm=T))
```

```
##   count format(round(median(RTQUIC_survival_hours, na.rm = T), 2), 2)    IQR
## 1     22                           27.88  6.625
##   min max
## 1 16.5 40
```

```
RTposdf %>% group_by(DX_APD) %>% summarize(count=n(), format(round(median(RTQUIC_surviva
l_hours, na.rm=T),2),2), IQR=IQR(RTQUIC_survival_hours, na.rm=T), min=min(RTQUIC_surviva
l_hours, na.rm=T), max=max(RTQUIC_survival_hours, na.rm=T))
```

```
## # A tibble: 2 × 6
##   DX_APD count format(round(median(RTQUIC_survival_hours, na...¹  IQR   min   max
##   <chr>  <int> <chr>                               <dbl> <dbl> <dbl>
## 1 CBS      14 31.5                                8     26     40
## 2 PSP      8 24.75                               3.12  16.5  31.5
## # i abbreviated name:
## #   ¹`format(round(median(RTQUIC_survival_hours, na.rm = T), 2), 2)`
```

```
RTposdf %>% summarize(count=n(), format(round(median(RTQUIC_survival_hours, na.rm=T),2),
2), IQR=IQR(RTQUIC_survival_hours, na.rm=T), min=min(RTQUIC_survival_hours, na.rm=T), ma
x=max(RTQUIC_survival_hours, na.rm=T))
```

```
##   count format(round(median(RTQUIC_survival_hours, na.rm = T), 2), 2)    IQR
## 1     22                           27.88  6.625
##   min max
## 1 16.5 40
```

```
RTposdf %>% group_by(AD) %>% summarize(count=n(), format(round(median(RTQUIC_survival_hours, na.rm=T),2),2), IQR=IQR(RTQUIC_survival_hours, na.rm=T), min=min(RTQUIC_survival_hours, na.rm=T), max=max(RTQUIC_survival_hours, na.rm=T))
```

```
## # A tibble: 2 × 6
##   AD           count format(round(median(RTQUIC_survival_hours, na.rm = T), 2), 2)    IQR     min     max
##   <chr>        <int> <chr>
## 1 AD Negative    15  26.25          6.75  16.5  40
## 2 AD Positive     7  32.25          5.25  26    39.8
## # i abbreviated name:
## #   `format(round(median(RTQUIC_survival_hours, na.rm = T), 2), 2)`
```

```
RTposdf %>% summarize(count=n(), format(round(median(RTQUIC_survival_hours, na.rm=T),2),2), IQR=IQR(RTQUIC_survival_hours, na.rm=T), min=min(RTQUIC_survival_hours, na.rm=T), max=max(RTQUIC_survival_hours, na.rm=T))
```

```
##   count format(round(median(RTQUIC_survival_hours, na.rm = T), 2), 2)    IQR
## 1     22                      27.88 6.625
##   min max
## 1 16.5 40
```

```
RTposdf %>% group_by(Early_onset) %>% summarize(count=n(), format(round(median(RTQUIC_survival_hours, na.rm=T),2),2), IQR=IQR(RTQUIC_survival_hours, na.rm=T), min=min(RTQUIC_survival_hours, na.rm=T), max=max(RTQUIC_survival_hours, na.rm=T))
```

```
## # A tibble: 2 × 6
##   Early_onset count format(round(median(RTQUIC_survival_hours, na.rm = T), 2), 2)    IQR     min     max
##   <fct>        <int> <chr>
## 1 Late-onset     14  27.12          6.62  16.5  40
## 2 Young-onset     8  30.38          10.4   24.8  39.8
## # i abbreviated name:
## #   `format(round(median(RTQUIC_survival_hours, na.rm = T), 2), 2)`
```

LAG STATISTICS: CORRELATIONS

```
cor.test(RTposdf$RTQUIC_survival_hours, RTposdf$Onset_age, method="spearman") #correlate d
```

```
## Warning in cor.test.default(RTposdf$RTQUIC_survival_hours, RTposdf$Onset_age, : 
## Cannot compute exact p-value with ties
```

```
##  
## Spearman's rank correlation rho  
##  
## data: RTposdf$RTQUIC_survival_hours and RTposdf$Onset_age  
## S = 2528.7, p-value = 0.047  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho  
## -0.427841
```

```
cor.test(RTposdf$RTQUIC_survival_hours, RTposdf$Age, method="spearman")
```

```
## Warning in cor.test.default(RTposdf$RTQUIC_survival_hours, RTposdf$Age, :  
## Cannot compute exact p-value with ties
```

```
##  
## Spearman's rank correlation rho  
##  
## data: RTposdf$RTQUIC_survival_hours and RTposdf$Age  
## S = 2351, p-value = 0.1368  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho  
## -0.3274806
```

```
cor.test(RTposdf$RTQUIC_survival_hours, RTposdf$logabeta, method="spearman")
```

```
## Warning in cor.test.default(RTposdf$RTQUIC_survival_hours, RTposdf$logabeta, :  
## Cannot compute exact p-value with ties
```

```
##  
## Spearman's rank correlation rho  
##  
## data: RTposdf$RTQUIC_survival_hours and RTposdf$logabeta  
## S = 1925.5, p-value = 0.6994  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho  
## -0.08725263
```

```
cor.test(RTposdf$RTQUIC_survival_hours, RTposdf$logNFL, method="spearman")
```

```
## Warning in cor.test.default(RTposdf$RTQUIC_survival_hours, RTposdf$logNFL, :  
## Cannot compute exact p-value with ties
```

```
##  
## Spearman's rank correlation rho  
##  
## data: RTposdf$RTQUIC_survival_hours and RTposdf$logNFL  
## S = 1620.3, p-value = 0.8224  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho  
## -0.05215164
```

```
cor.test(RTposdf$RTQUIC_survival_hours, RTposdf$logptau, method="spearman") #correlated
```

```
## Warning in cor.test.default(RTposdf$RTQUIC_survival_hours, RTposdf$logptau, :  
## Cannot compute exact p-value with ties
```

```
##  
## Spearman's rank correlation rho  
##  
## data: RTposdf$RTQUIC_survival_hours and RTposdf$logptau  
## S = 856.9, p-value = 0.01393  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho  
## 0.5161503
```

```
cor.test(RTposdf$RTQUIC_survival_hours, RTposdf$logtau, method="spearman")
```

```
## Warning in cor.test.default(RTposdf$RTQUIC_survival_hours, RTposdf$logtau, :  
## Cannot compute exact p-value with ties
```

```
##  
## Spearman's rank correlation rho  
##  
## data: RTposdf$RTQUIC_survival_hours and RTposdf$logtau  
## S = 1002, p-value = 0.1206  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho  
## 0.3493826
```

```
cor.test(RTposdf$RTQUIC_survival_hours, RTposdf$ATI_2, method="spearman")
```

```
## Warning in cor.test.default(RTposdf$RTQUIC_survival_hours, RTposdf$ATI_2, :  
## Cannot compute exact p-value with ties
```

```

## 
## Spearman's rank correlation rho
##
## data: RTposdf$RTQUIC_survival_hours and RTposdf$ATI_2
## S = 2052, p-value = 0.1409
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## -0.3324665

```

LAG STATISTICS: CORRELATION PLOTS

```

# Age at onset
ggscatter(RTposdf, x="Onset_age", y= "RTQUIC_survival_hours", fill="DX_APD",
           size=5, shape=21,
           palette = c(CBS= "#56B4E9", PSP = "#CC79A7"),
           add = "reg.line", cor.coef = TRUE, cor.method = "spearman",
           title="Correlation of lag with age at onset (Spearman)",
           xlab="Age at onset (years)", ylab = "Lag (hours)") +
           scale_fill_discrete(name = 'Diagnosis',labels = c("CBS", "PSP"))

```

```

## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.

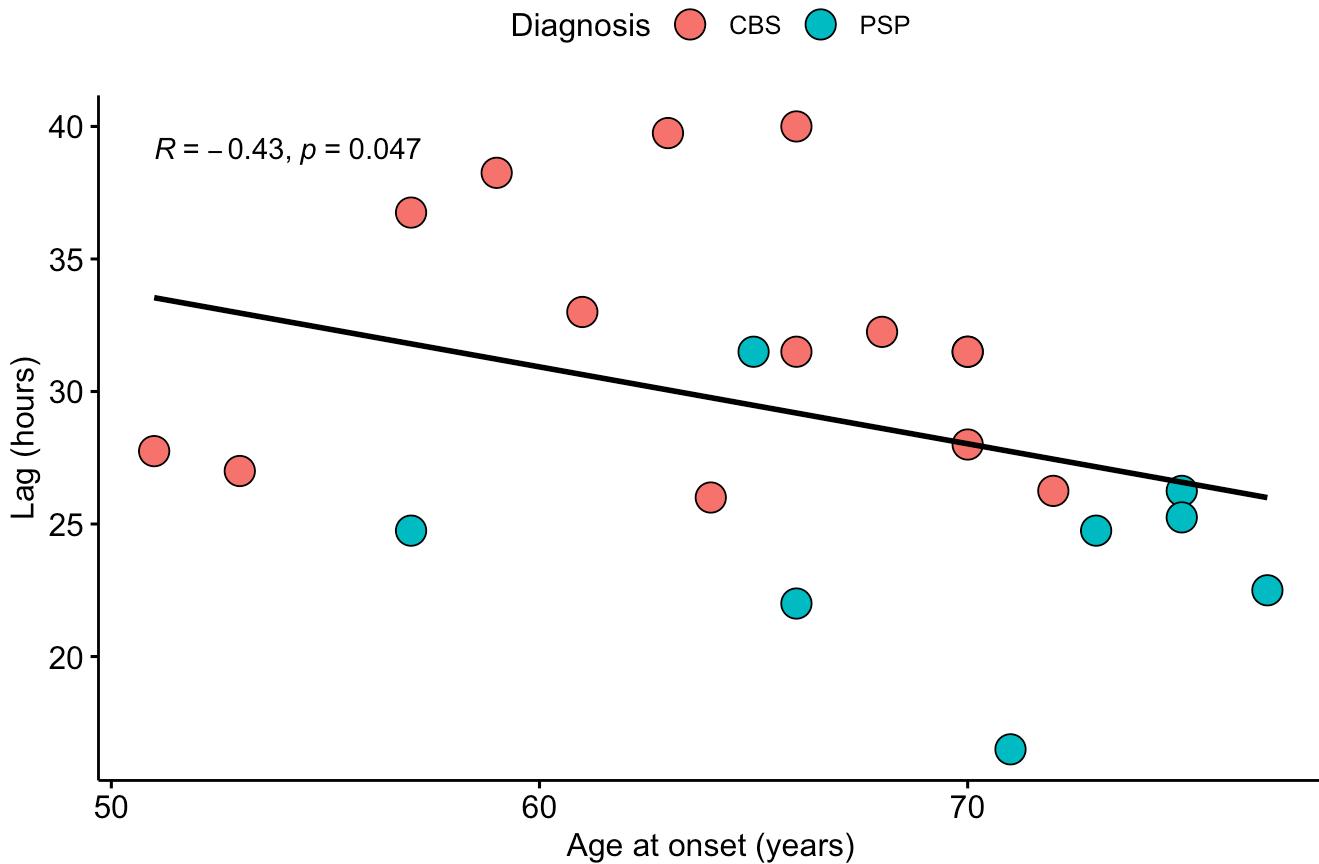
```

```

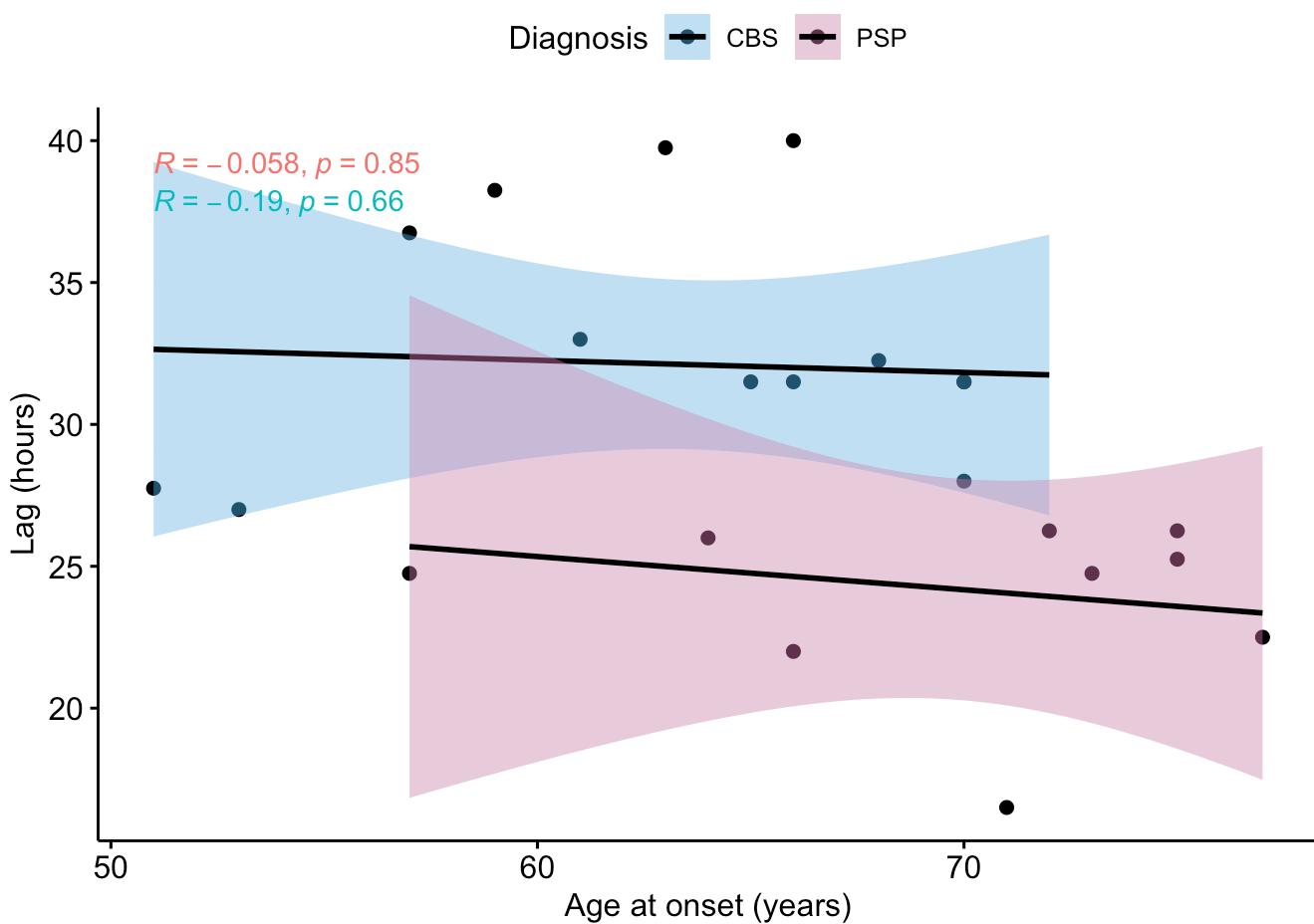
## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's colour values.

```

Correlation of lag with age at onset (Spearman)



```
ggscatter(RTposdf, x="Onset_age", y= "RTQUIC_survival_hours", fill="DX_APD",
  xlab="Age at onset (years)", ylab = "Lag (hours)",
  add = "reg.line", conf.int = TRUE) +
  stat_cor(aes(color = DX_APD, label = paste("R^2~=", ~))), show.legend = FALSE) +
  scale_fill_manual(values=cbPalette_DX_APD, name="Diagnosis", labels=c("CBS", "PS
P"))
```

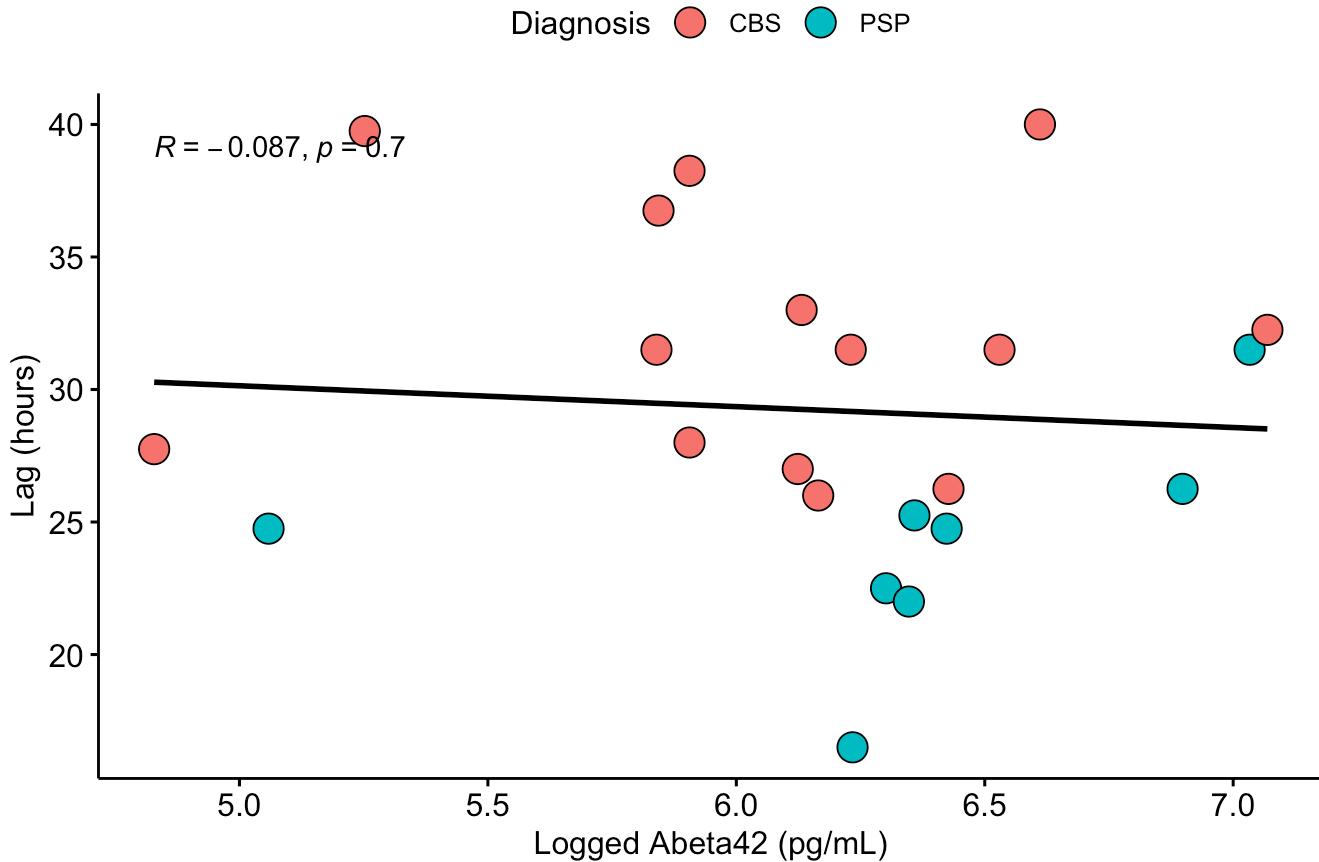


```
# A beta42
ggscatter(RTposdf, x="logabeta", y= "RTQUIC_survival_hours", fill="DX_APD",
           size=5, shape=21,
           palette = c(CBS= "#56B4E9", PSP = "#CC79A7"),
           add = "reg.line", cor.coef = TRUE, cor.method = "spearman",
           title="Correlation of lag with Abeta42 levels (Spearman)",
           xlab="Logged Abeta42 (pg/mL)", ylab = "Lag (hours)") +
           scale_fill_discrete(name = 'Diagnosis', labels = c("CBS", "PSP"))
```

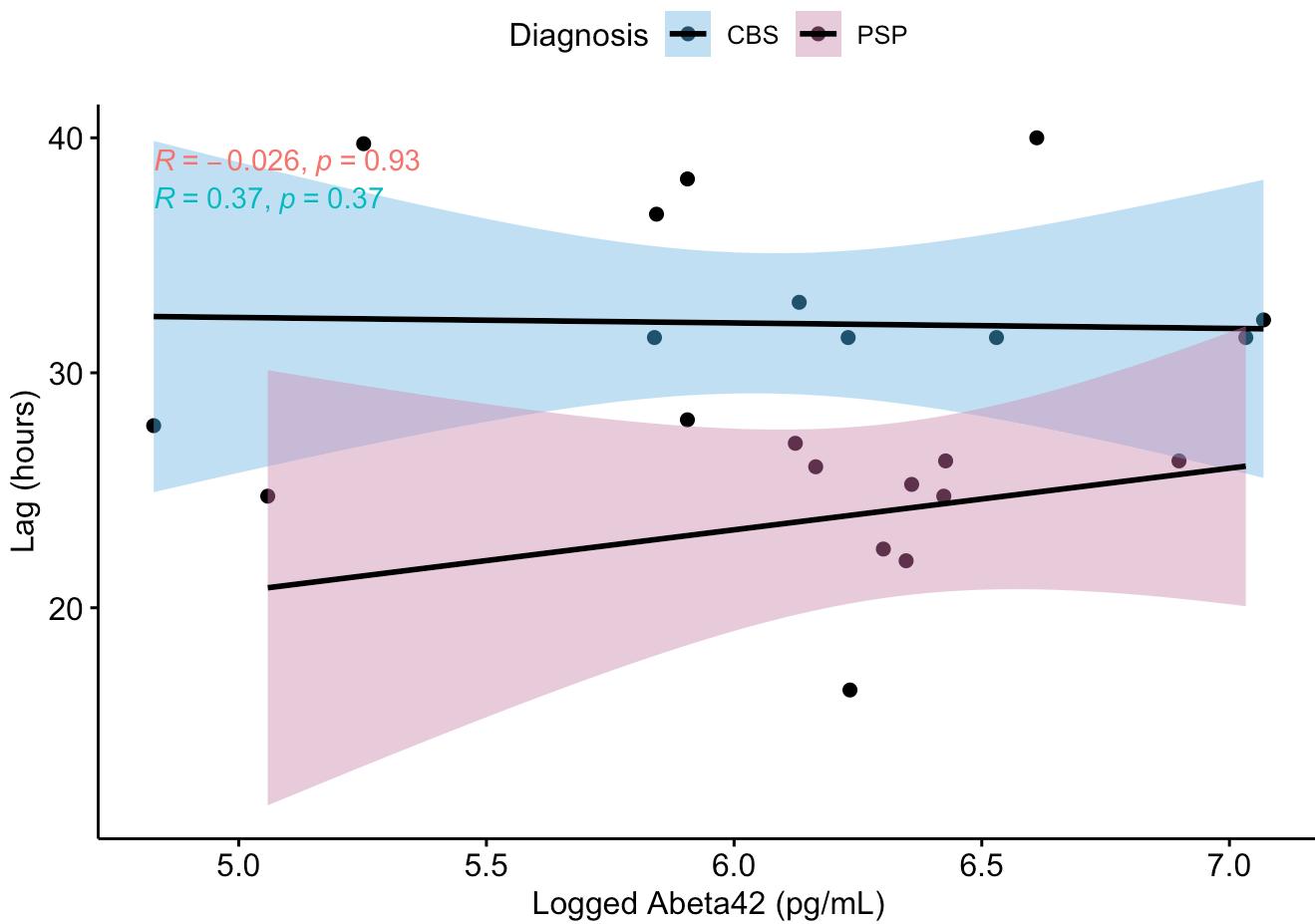
```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```
## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's colour values.
```

Correlation of lag with Abeta42 levels (Spearman)



```
ggscatter(RTposdf, x="logabeta", y= "RTQUIC_survival_hours", fill="DX_APD",
  xlab="Logged Abeta42 (pg/mL)", ylab = "Lag (hours)",
  add = "reg.line", conf.int = TRUE) +
  stat_cor(aes(color = DX_APD, label = paste("R^2~=", ~))), show.legend = FALSE) +
  scale_fill_manual(values=cbPalette_DX_APD, name="Diagnosis", labels=c("CBS", "PS
P"))
```

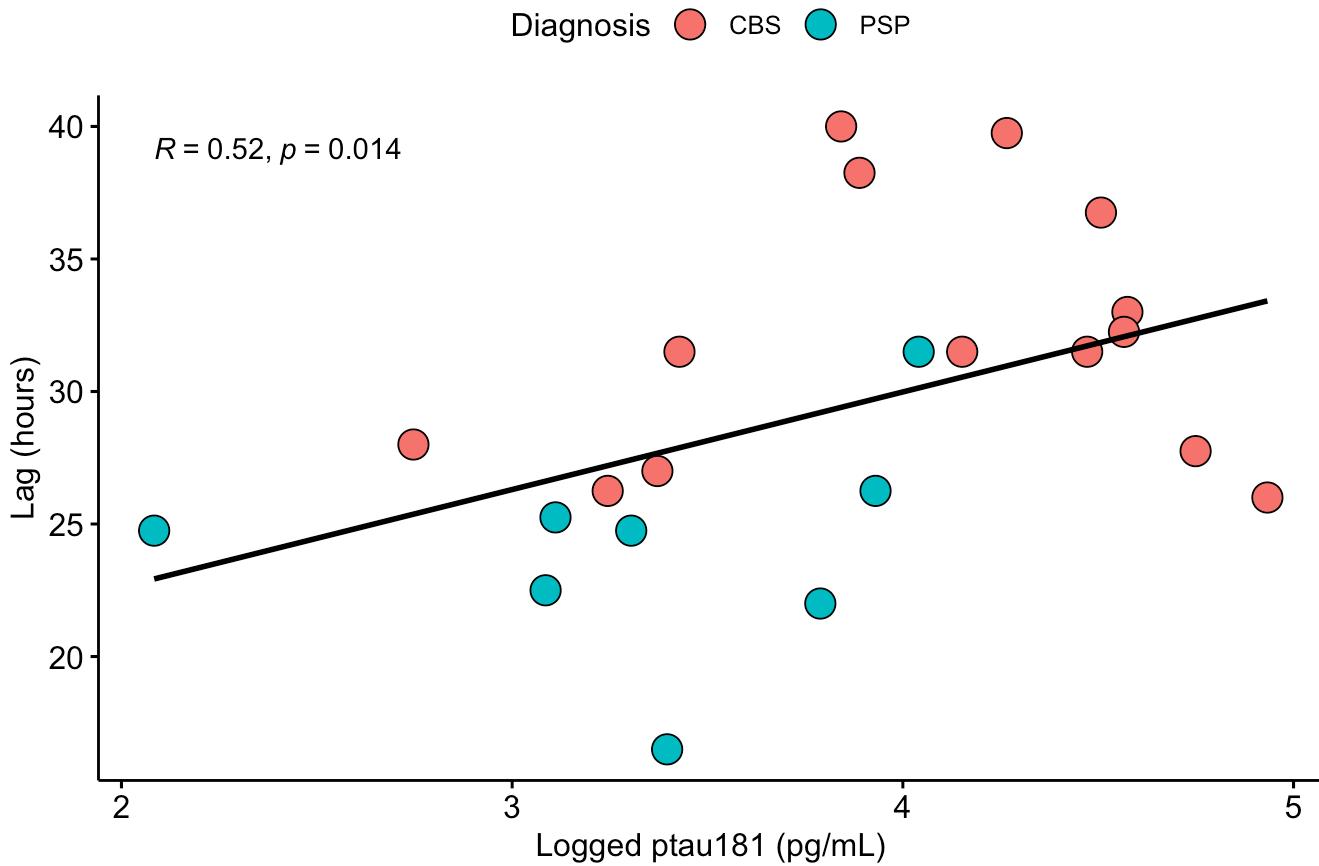


```
# Ptau181
ggscatter(RTposdf, x="logptau", y= "RTQUIC_survival_hours", fill="DX_APD",
           size=5, shape=21,
           palette = c(CBS= "#56B4E9", PSP = "#CC79A7"),
           add = "reg.line", cor.coef = TRUE, cor.method = "spearman",
           title="Correlation of lag with ptau181 levels (Spearman)",
           xlab="Logged ptau181 (pg/mL)", ylab = "Lag (hours)") +
           scale_fill_discrete(name = 'Diagnosis', labels = c("CBS", "PSP"))
```

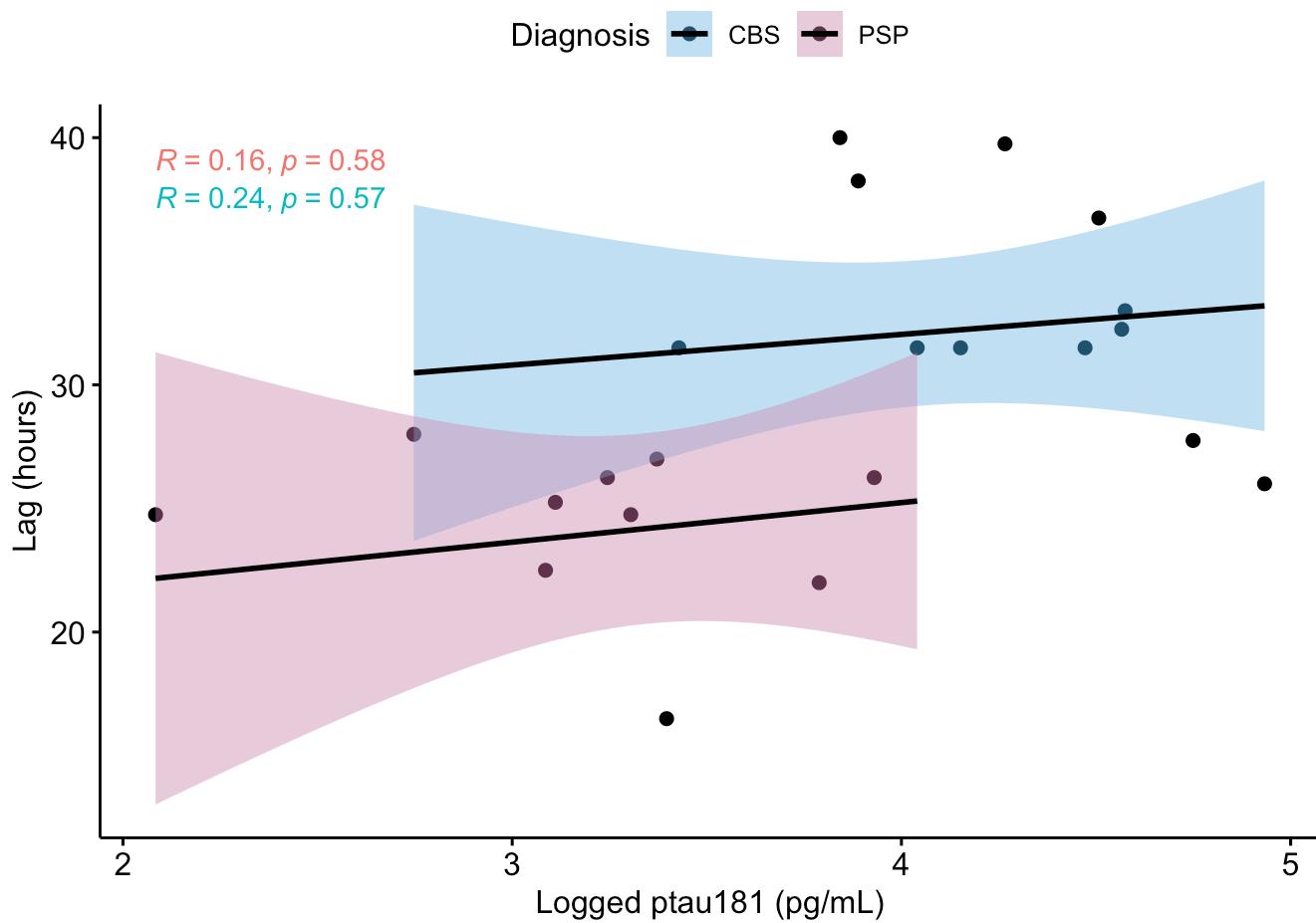
```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```
## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's colour values.
```

Correlation of lag with ptau181 levels (Spearman)



```
ggscatter(RTposdf, x="logptau", y= "RTQUIC_survival_hours", fill="DX_APD",
  xlab="Logged ptau181 (pg/mL)", ylab = "Lag (hours)",
  add = "reg.line", conf.int = TRUE) +
  stat_cor(aes(color = DX_APD, label = paste("R^2~=", ~))), show.legend = FALSE) +
  scale_fill_manual(values=cbPalette_DX_APD, name="Diagnosis", labels=c("CBS", "PS
P"))
```



```
# T-tau
ggscatter(RTposdf, x="logttau", y= "RTQUIC_survival_hours", fill="DX_APD",
           size=5, shape=21,
           palette = c(CBS= "#56B4E9", PSP = "#CC79A7"),
           add = "reg.line", cor.coef = TRUE, cor.method = "spearman",
           title="Correlation of lag with t-tau levels (Spearman)",
           xlab="Logged t-tau (pg/mL)", ylab = "Lag (hours)") +
  scale_fill_discrete(name = 'Diagnosis',labels = c("CBS", "PSP"))
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

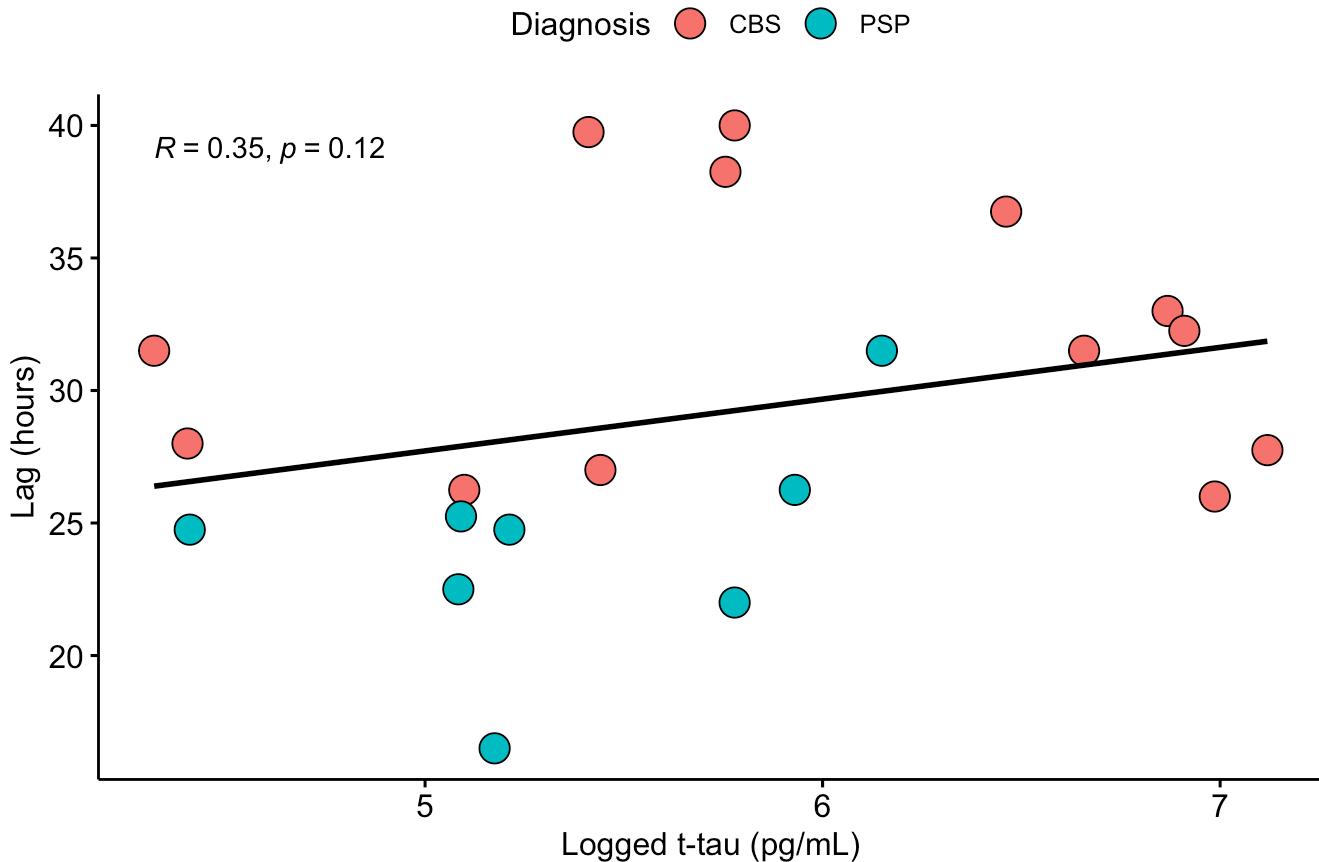
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_cor()`).
```

```
## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's colour values.
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

Correlation of lag with t-tau levels (Spearman)

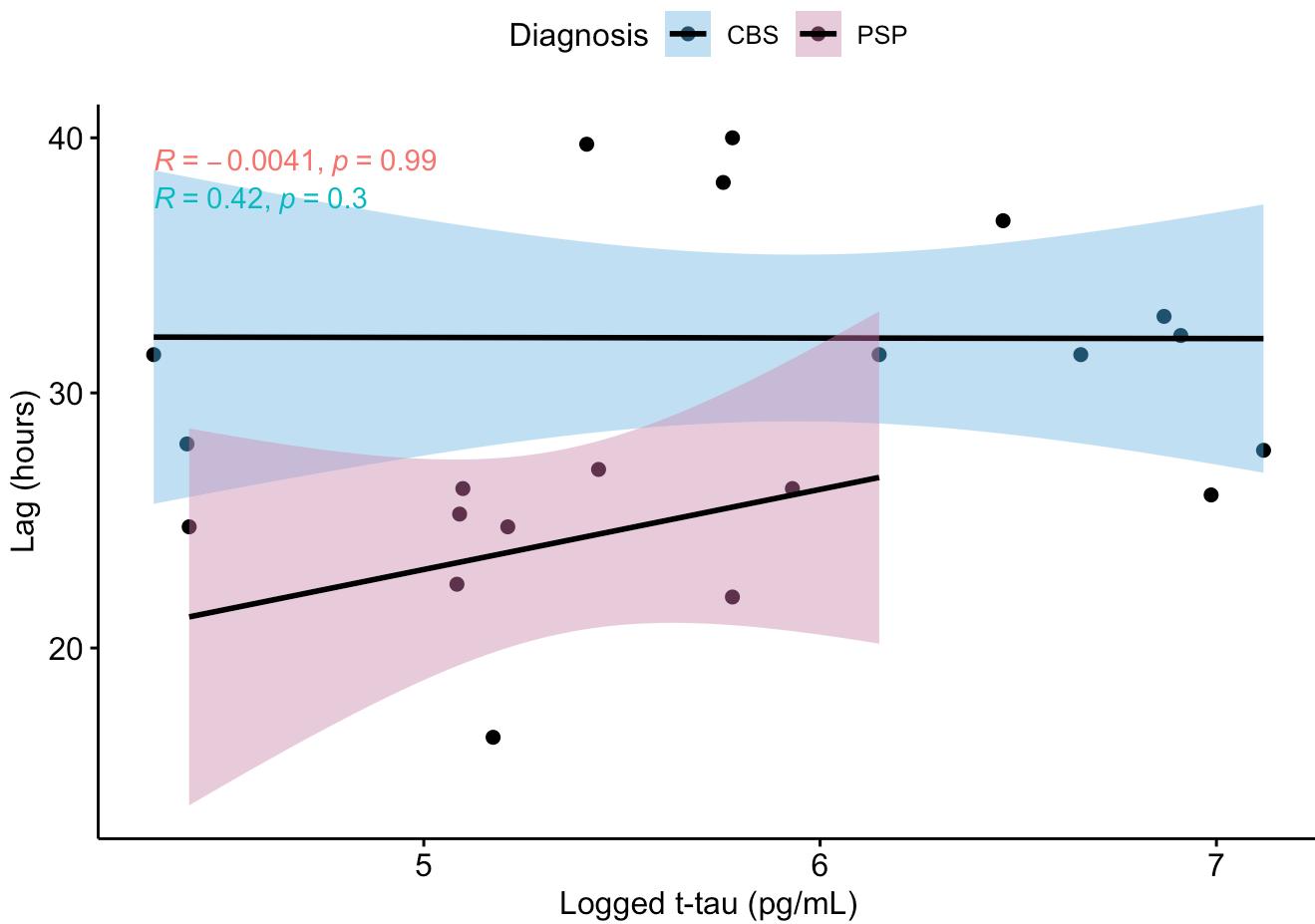


```
ggscatter(RTposdf, x="logttau", y= "RTQUIC_survival_hours", fill="DX_APD",
          xlab="Logged t-tau (pg/mL)", ylab = "Lag (hours)",
          add = "reg.line", conf.int = TRUE) +
  stat_cor(aes(color = DX_APD, label = paste("R^2~=", ~))), show.legend = FALSE) +
  scale_fill_manual(values=cbPalette_DX_APD, name="Diagnosis", labels=c("CBS", "PS
P"))
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_cor()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



```
# NfL
ggscatter(RTposdf, x="logNFL", y= "RTQUIC_survival_hours", fill="DX_APD",
           size=5, shape=21,
           palette = c(CBS= "#56B4E9", PSP = "#CC79A7"),
           add = "reg.line", cor.coef = TRUE, cor.method = "spearman",
           title="Correlation of lag with t-tau levels (Spearman)",
           xlab="Logged NfL (pg/mL)", ylab = "Lag (hours)") +
           scale_fill_discrete(name = 'Diagnosis', labels = c("CBS", "PSP"))
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

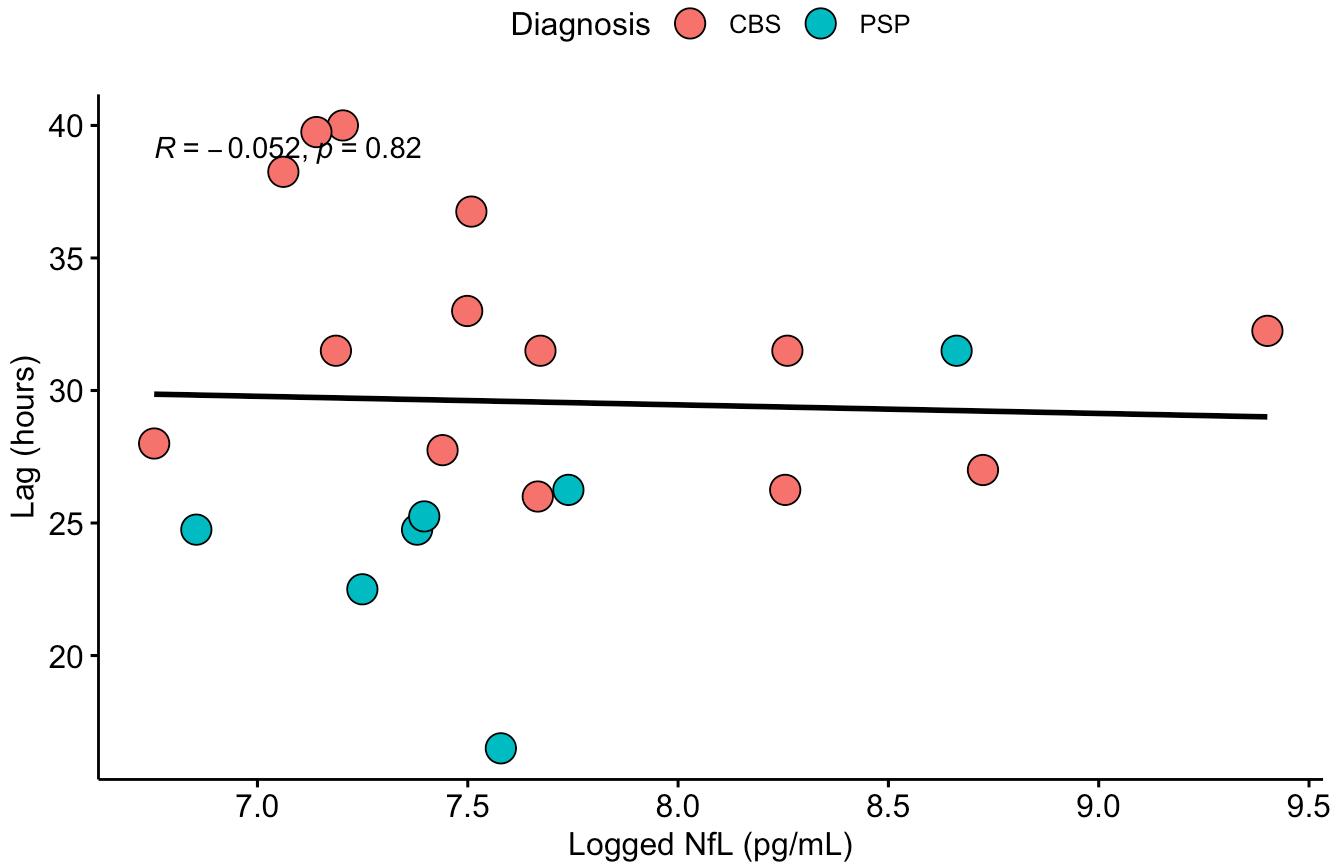
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_cor()`).
```

```
## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's colour values.
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

Correlation of lag with t-tau levels (Spearman)

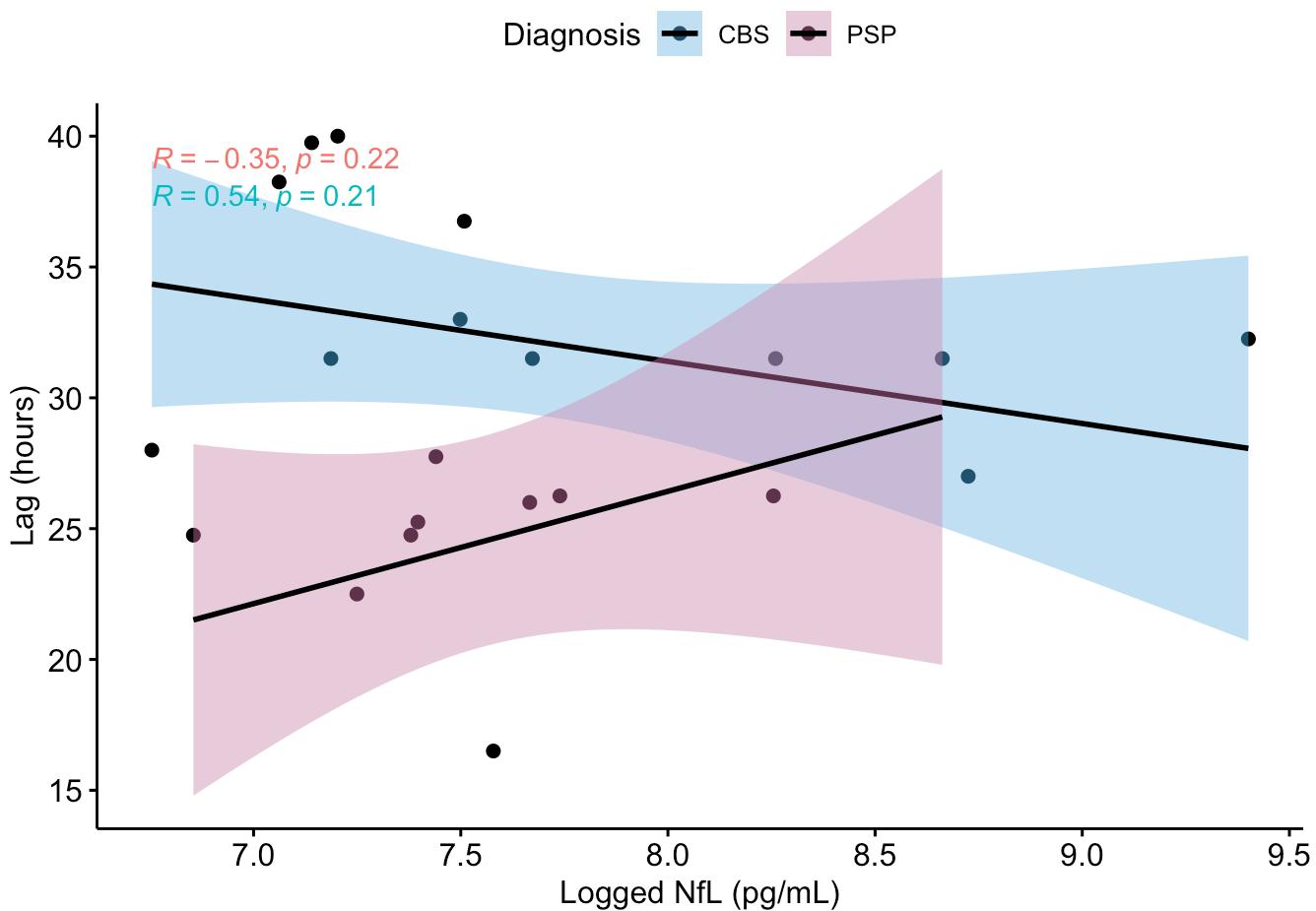


```
ggscatter(RTposdf, x="logNFL", y= "RTQUIC_survival_hours", fill="DX_APD",
          xlab="Logged NfL (pg/mL)", ylab = "Lag (hours)",
          add = "reg.line", conf.int = TRUE) +
  stat_cor(aes(color = DX_APD, label = paste("R^2~=", ~))), show.legend = FALSE) +
  scale_fill_manual(values=cbPalette_DX_APD, name="Diagnosis", labels=c("CBS", "PS
P"))
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_cor()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



```
# ATI
ggscatter(RTposdf, x="ATI_2", y= "RTQUIC_survival_hours", fill="DX_APD",
           size=5, shape=21,
           palette = c(CBS= "#56B4E9", PSP = "#CC79A7"),
           add = "reg.line", cor.coef = TRUE, cor.method = "spearman",
           title="Correlation of lag with t-tau levels (Spearman)",
           xlab="ATI", ylab = "Lag (hours)") +
           scale_fill_discrete(name = 'Diagnosis',labels = c("CBS", "PSP"))
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

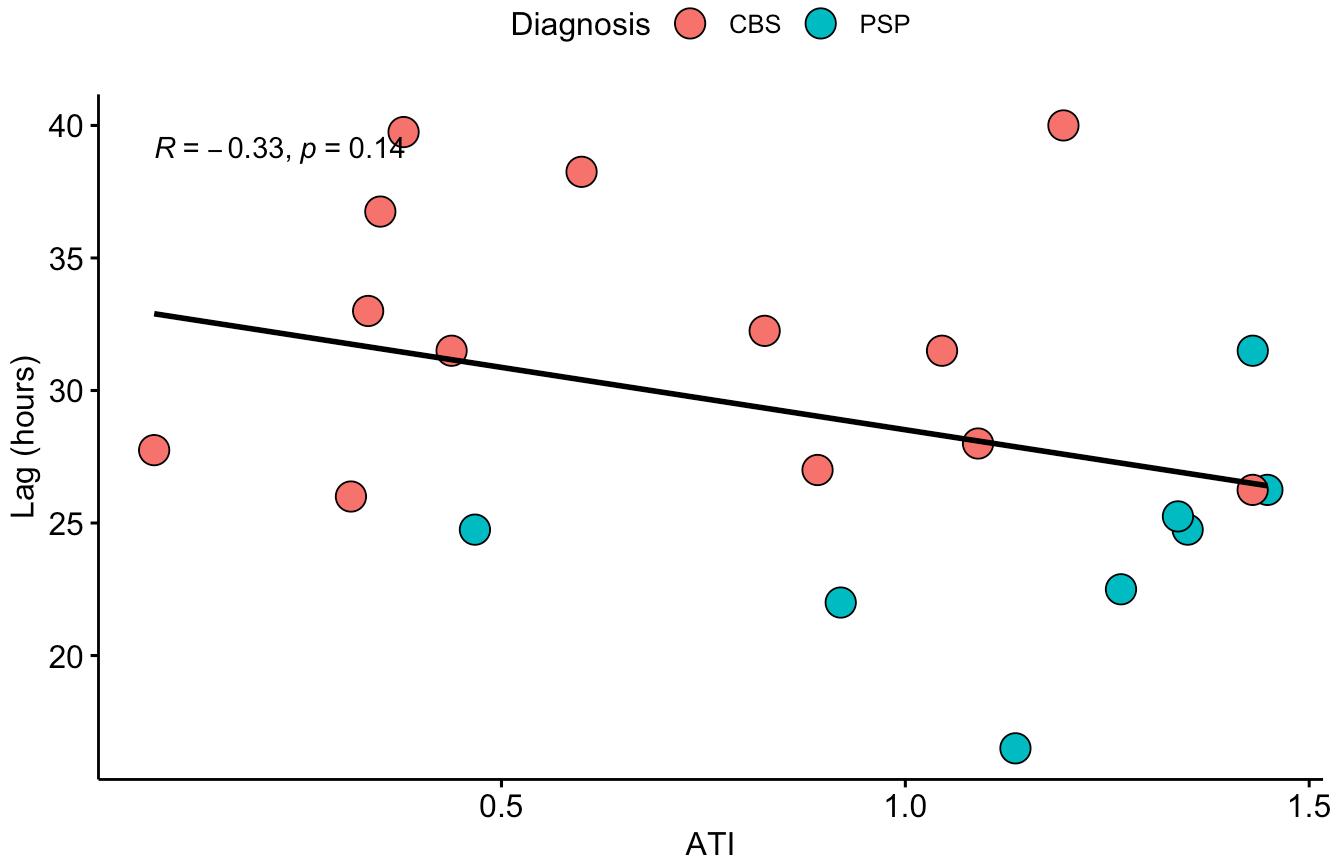
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_cor()`).
```

```
## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's colour values.
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

Correlation of lag with t-tau levels (Spearman)

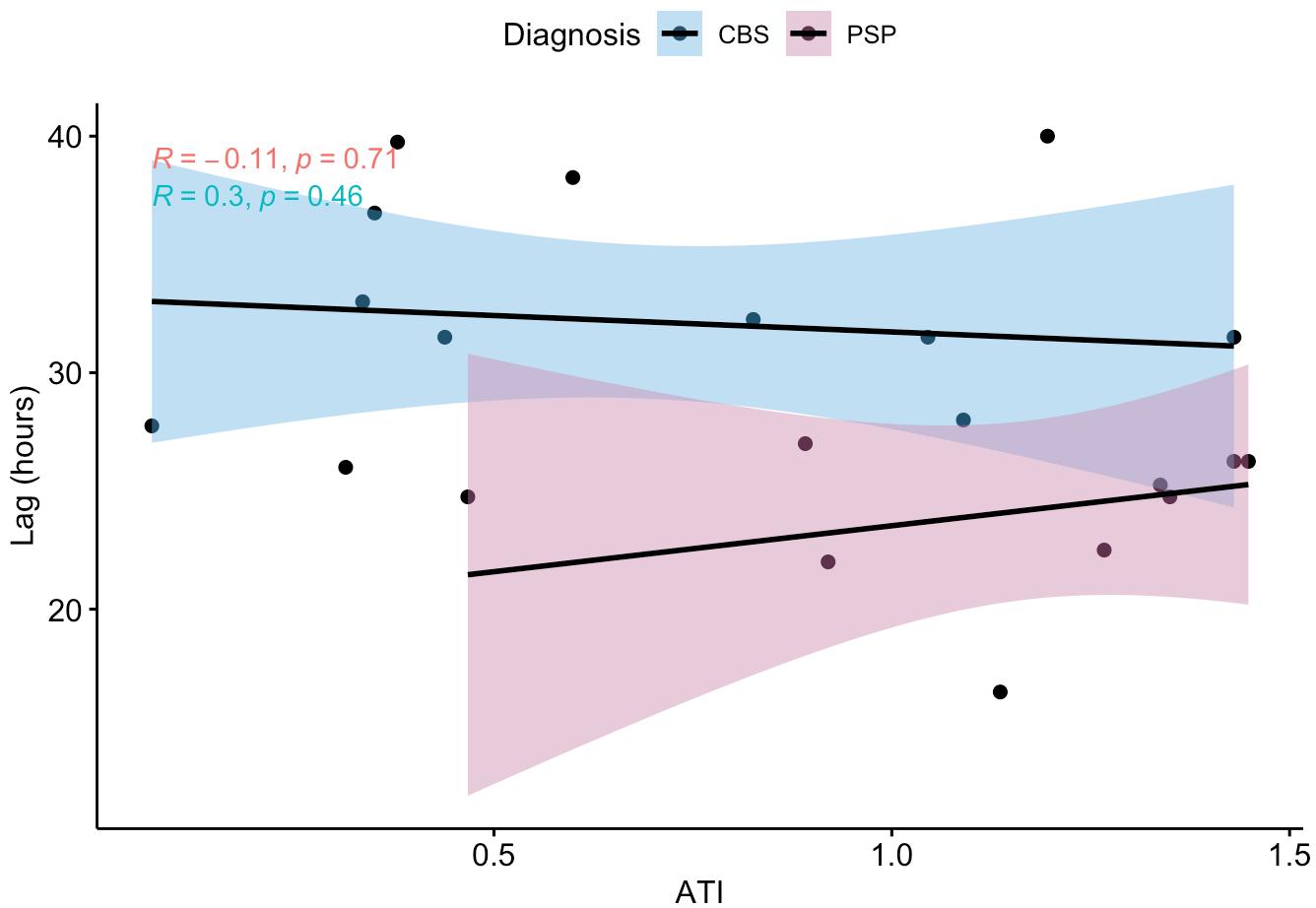


```
ggscatter(RTposdf, x="ATI_2", y= "RTQUIC_survival_hours", fill="DX_APD",
  xlab="ATI", ylab = "Lag (hours)",
  add = "reg.line", conf.int = TRUE) +
  stat_cor(aes(color = DX_APD, label = paste("R^2~=", ~))), show.legend = FALSE) +
  scale_fill_manual(values=cbPalette_DX_APD, name="Diagnosis", labels=c("CBS", "PS
P"))
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_cor()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



LAG STATISTICS: LOG-RANK TEST

```
summary(survfit(Surv(RTQUIC_survival_hours, RTQUIC_survived) ~ Early_onset, data = df),
times = 48)
```

```
## Call: survfit(formula = Surv(RTQUIC_survival_hours, RTQUIC_survived) ~
##   Early_onset, data = df)
##
##           Early_onset=Late-onset
##   time      n.risk      n.event      survival      std.err  lower 95% CI
##   48.0000    18.0000    14.0000    0.5625    0.0877    0.4144
## upper 95% CI
##   0.7635
##
##           Early_onset=Young-onset
##   time      n.risk      n.event      survival      std.err  lower 95% CI
##   48.0000    27.0000     8.0000    0.7710    0.0710    0.6440
## upper 95% CI
##   0.924
```

```
survdiff(Surv(RTQUIC_survival_hours, RTQUIC_survived) ~ Early_onset, data = df) #p=.05
```

```
## Call:
## survdiff(formula = Surv(RTQUIC_survival_hours, RTQUIC_survived) ~
##   Early_onset, data = df)
##
##           N Observed Expected (0-E)^2/E (0-E)^2/V
## Early_onset=Late-onset 32      14     9.54     2.08     3.73
## Early_onset=Young-onset 35       8    12.46     1.59     3.73
##
##  Chisq= 3.7 on 1 degrees of freedom, p= 0.05
```

```
summary(survfit(Surv(RTQUIC_survival_hours, RTQUIC_survived) ~ AD, data = df), times = 48)
```

```
## Call: survfit(formula = Surv(RTQUIC_survival_hours, RTQUIC_survived) ~
##   AD, data = df)
##
##           AD=AD Negative
##   time      n.risk      n.event      survival      std.err lower 95% CI
##   48.0000    37.0000    15.0000     0.7115     0.0628    0.5985
## upper 95% CI
##   0.8460
##
##           AD=AD Positive
##   time      n.risk      n.event      survival      std.err lower 95% CI
##   48.0000    8.0000     7.0000     0.533      0.129     0.332
## upper 95% CI
##   0.856
```

```
survdiff(Surv(RTQUIC_survival_hours, RTQUIC_survived) ~ AD, data = df)
```

```
## Call:
## survdiff(formula = Surv(RTQUIC_survival_hours, RTQUIC_survived) ~
##   AD, data = df)
##
##           N Observed Expected (0-E)^2/E (0-E)^2/V
## AD=AD Negative 52      15     17.02     0.239     1.07
## AD=AD Positive 15       7     4.98     0.817     1.07
##
##  Chisq= 1.1 on 1 degrees of freedom, p= 0.3
```

```
summary(survfit(Surv(RTQUIC_survival_hours, RTQUIC_survived) ~ DX_APD, data = df), times = 48)
```

```

## Call: survfit(formula = Surv(RTQUIC_survival_hours, RTQUIC_survived) ~
##   DX_APD, data = df)
##
##           DX_APD=CBS
##       time    n.risk    n.event    survival    std.err lower 95% CI
##   48.0000    25.0000    14.0000    0.6410    0.0768    0.5068
## upper 95% CI
##      0.8107
##
##           DX_APD=PSP
##       time    n.risk    n.event    survival    std.err lower 95% CI
##   48.0000    20.0000     8.0000    0.7143    0.0854    0.5651
## upper 95% CI
##      0.9028

```

```
survdiff(Surv(RTQUIC_survival_hours, RTQUIC_survived) ~ DX_APD, data = df)
```

```

## Call:
## survdiff(formula = Surv(RTQUIC_survival_hours, RTQUIC_survived) ~
##   DX_APD, data = df)
##
##           N Observed Expected (0-E)^2/E (0-E)^2/V
## DX_APD=CBS 39      14     13.42    0.0254    0.066
## DX_APD=PSP 28       8      8.58    0.0396    0.066
##
## Chisq= 0.1 on 1 degrees of freedom, p= 0.8

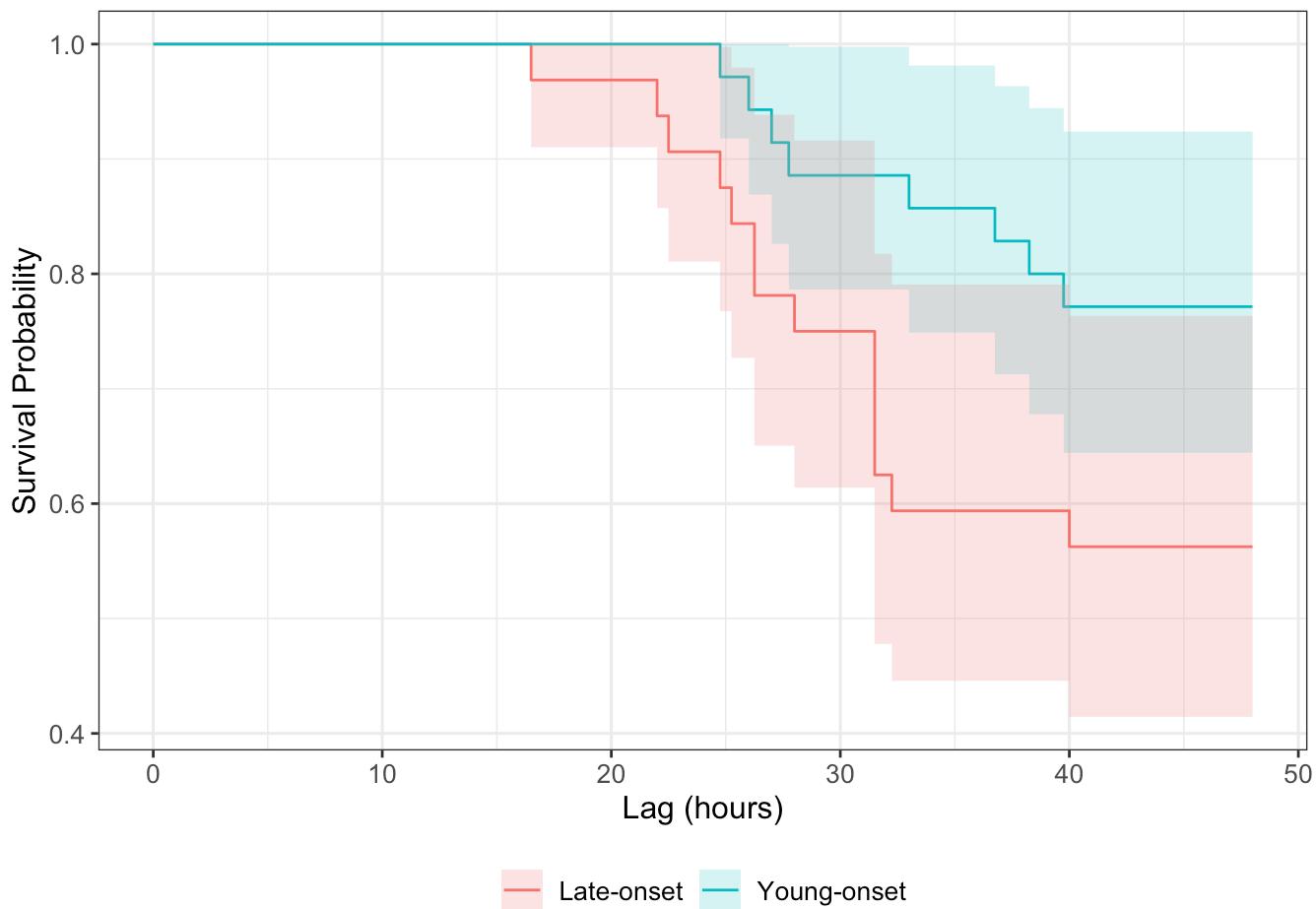
```

LAG STATISTICS: KAPLAN-MEIER CURVE FOR WHOLE DATASET

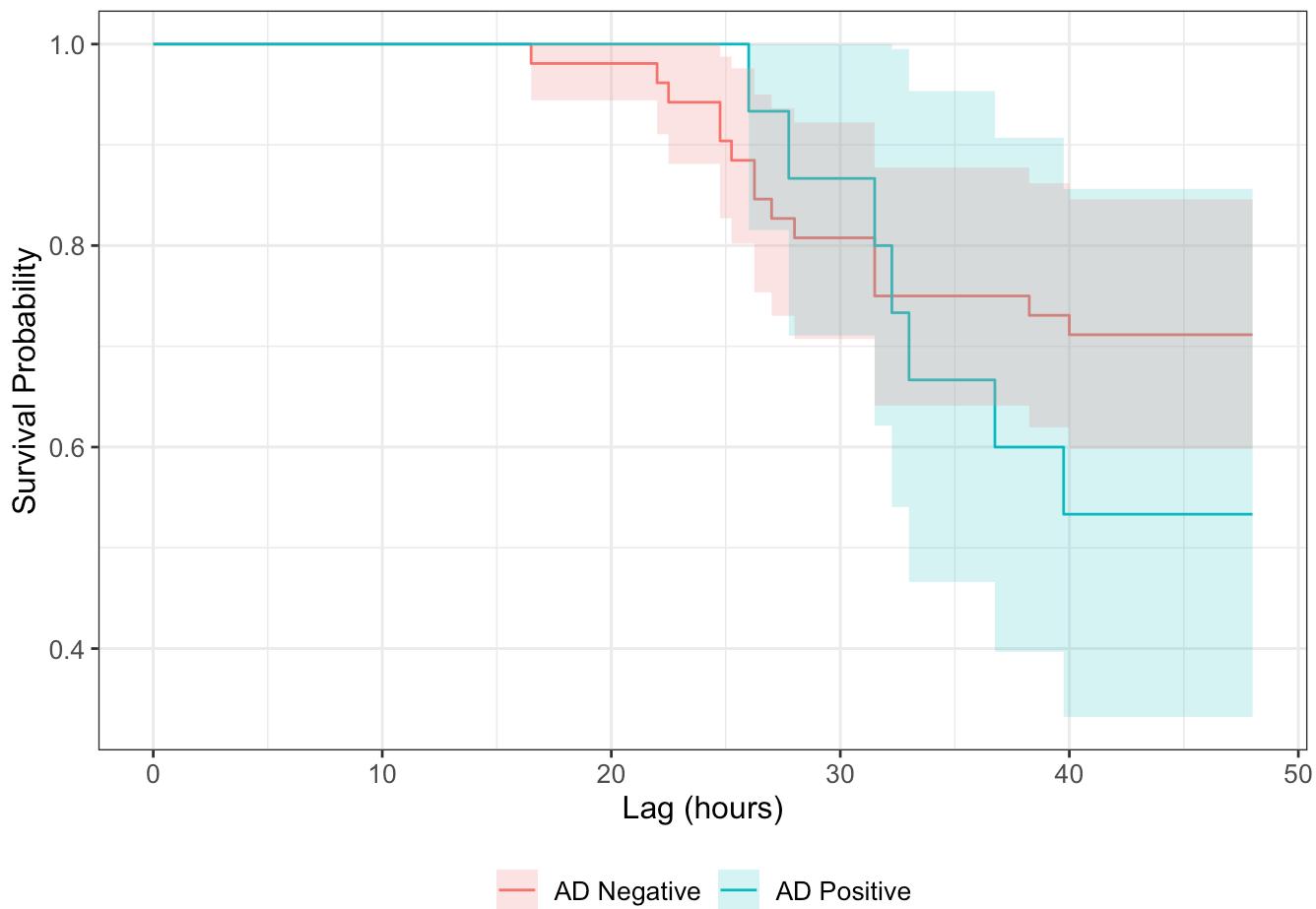
```

# Uses survfit2() in ggsurvfit() package:
survfit2(Surv(RTQUIC_survival_hours, RTQUIC_survived) ~ Early_onset, data = df) %>%
  ggsurvfit() + add_confidence_interval() + labs(x="Lag (hours)", "Survival probability") #bold() is required otherwise the Y axis will not be bold in spite of element_text specification below

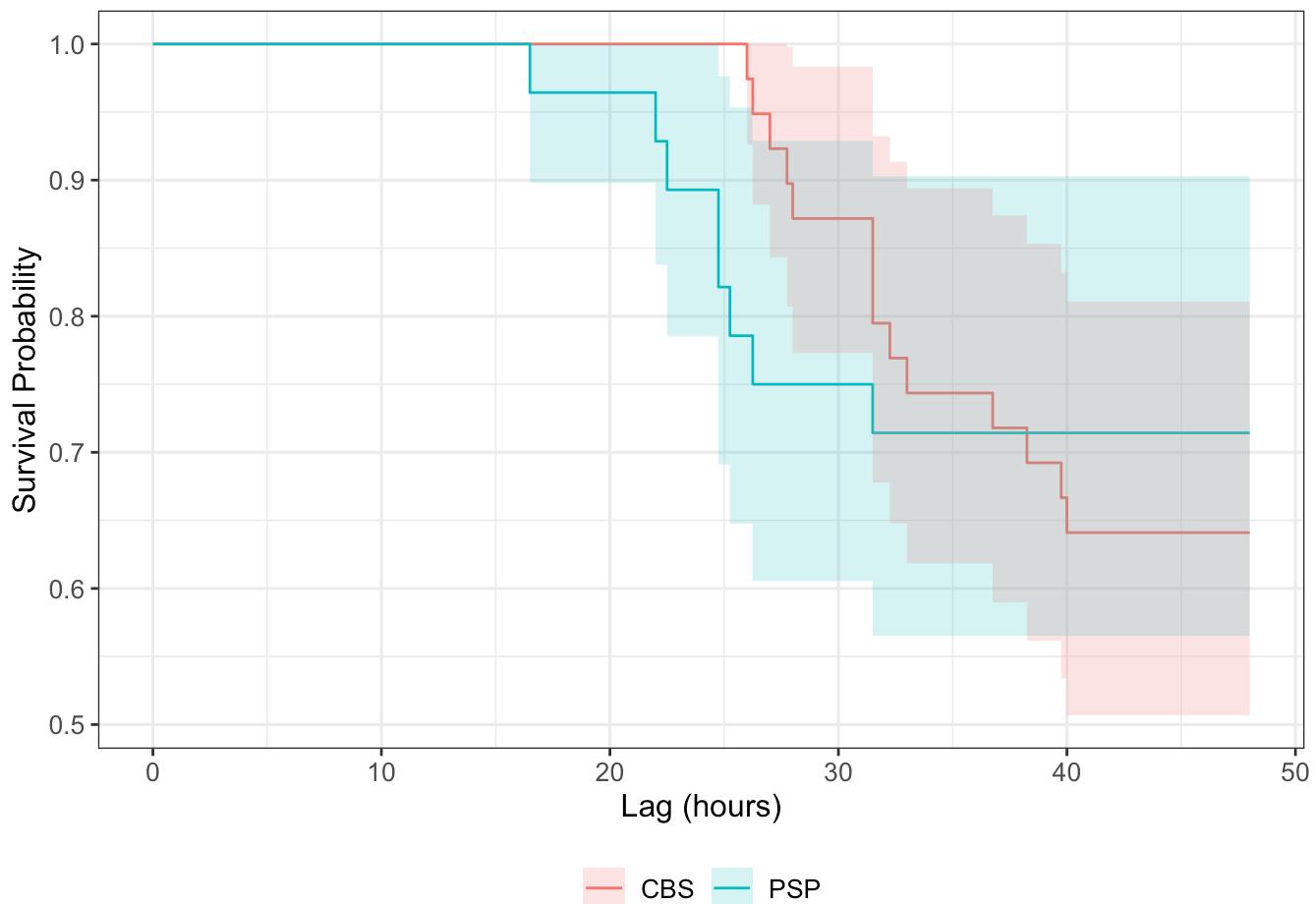
```



```
survfit2(Surv(RTQUIC_survival_hours, RTQUIC_survived) ~ AD, data = df) %>%
  ggsurvfit() + add_confidence_interval() + labs(x="Lag (hours)", "Survival probability") #bold() is required otherwise the Y axis will not be bold in spite of element_text specification below
```



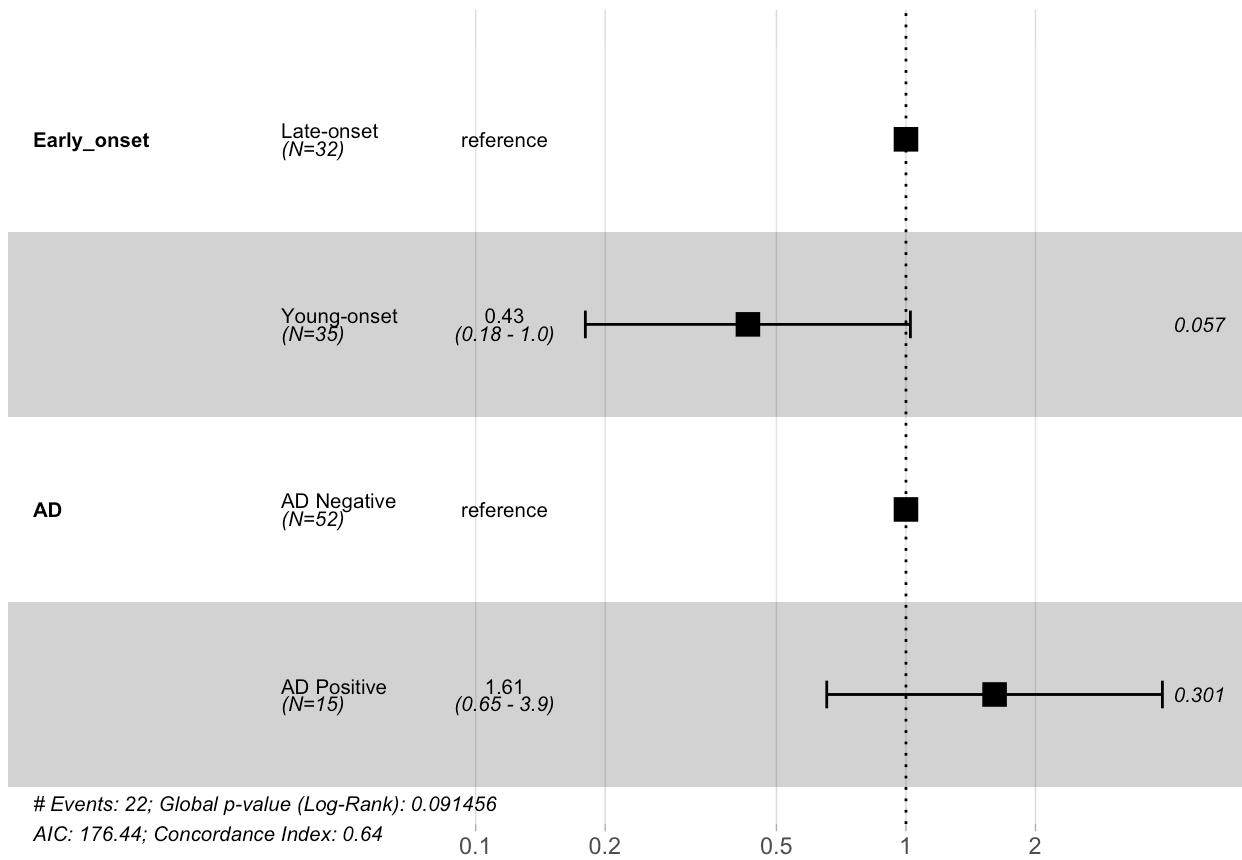
```
survfit2(Surv(RTQUIC_survival_hours, RTQUIC_survived) ~ DX_APD, data = df) %>%
  ggsurvfit() + add_confidence_interval() + labs(x="Lag (hours)", "Survival probability") #bold() is required otherwise the Y axis will not be bold in spite of element_text specification below
```



LAG STATISTICS: ANALYSES ON ONLY THE RT-QUIC SUBJECTS

```
# Fit a Cox proportional hazards model. Redundant with the fisher analyses already shown above
fit.coxph <- coxph(Surv(RTQUIC_survival_hours, RTQUIC_survived) ~ Early_onset + AD, data = df)
ggforest(fit.coxph, data = df) # Redundant with previous analyses using chisquare and fisher approach in the manuscript - but does suggest potential difference between AD+ and AD-.
```

Hazard ratio



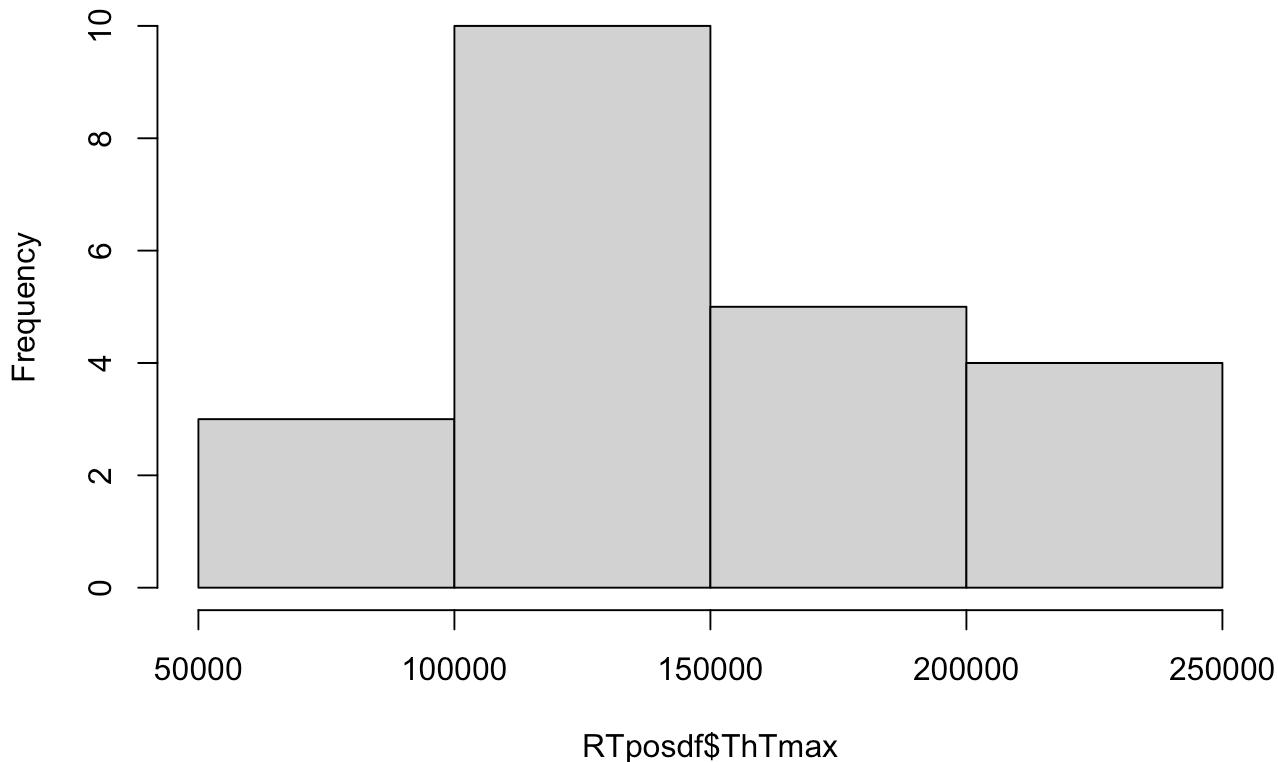
10.2. RTQUIC PARAMETERS SUPP: THT MAX

THT max is the max fluorescent signal reached after 48 hours of monitoring of the assay.

THT STATISTICS: DISTRIBUTION

```
hist(RTposdf$ThTmax)
```

Histogram of RTposdf\$ThTmax



```
shapiro.test(df[RTposdf$DX_APD == "CBS", ]$ThTmax) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: df[RTposdf$DX_APD == "CBS", ]$ThTmax  
## W = 0.97461, p-value = 0.8929
```

```
shapiro.test(df[RTposdf$DX_APD == "PSP", ]$ThTmax) #normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: df[RTposdf$DX_APD == "PSP", ]$ThTmax  
## W = 0.96251, p-value = 0.8254
```

```
shapiro.test(df[RTposdf$Early_onset == "Young-onset", ]$ThTmax) #normal
```

```
## 
## Shapiro-Wilk normality test
## 
## data: df[RTposdf$Early_onset == "Young-onset", ]$ThTmax
## W = 0.98373, p-value = 0.9944
```

```
shapiro.test(df[RTposdf$Early_onset == "Late-onset", ]$ThTmax) #normal
```

```
## 
## Shapiro-Wilk normality test
## 
## data: df[RTposdf$Early_onset == "Late-onset", ]$ThTmax
## W = 0.91162, p-value = 0.2923
```

```
shapiro.test(df[RTposdf$AD == "AD Positive", ]$ThTmax) #normal
```

```
## 
## Shapiro-Wilk normality test
## 
## data: df[RTposdf$AD == "AD Positive", ]$ThTmax
## W = 0.95514, p-value = 0.71
```

```
shapiro.test(df[RTposdf$AD == "AD Negative", ]$ThTmax) #normal
```

```
## 
## Shapiro-Wilk normality test
## 
## data: df[RTposdf$AD == "AD Negative", ]$ThTmax
## W = 0.90142, p-value = 0.1923
```

```
leveneTest(ThTmax ~ DX_APD, data = RTposdf) #homoscedasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.8632 0.3639
##       20
```

```
leveneTest(ThTmax ~ Early_onset, data = RTposdf) #homoscedasticity
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.1422 0.7101
##       20
```

```
leveneTest(ThTmax ~ AD, data = RTposdf) #homoscedasticity
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  2.0719 0.1655
##       20
```

THT STATISTICS: SUMMARY

```
RTposdf %>% group_by(DX_APD) %>% summarize(count=n(), format(round(mean(ThTmax, na.rm=T),2),2), sd=sd(ThTmax, na.rm=T))
```

```
## # A tibble: 2 × 4
##   DX_APD count `format(round(mean(ThTmax, na.rm = T), 2), 2)`      sd
##   <chr>    <int> <chr>                                         <dbl>
## 1 CBS        14  151528.1                                     49767.
## 2 PSP        8   151257.6                                     38716.
```

```
RTposdf %>% group_by(AD) %>% summarize(count=n(), format(round(mean(ThTmax, na.rm=T),2),2), sd=sd(ThTmax, na.rm=T))
```

```
## # A tibble: 2 × 4
##   AD          count `format(round(mean(ThTmax, na.rm = T), 2), 2)`      sd
##   <chr>        <int> <chr>                                         <dbl>
## 1 AD Negative  15  151575.8                                     51823.
## 2 AD Positive   7   151116.9                                     29129.
```

```
RTposdf %>% group_by(Early_onset) %>% summarize(count=n(), format(round(mean(ThTmax, na.rm=T),2),2), sd=sd(ThTmax, na.rm=T))
```

```
## # A tibble: 2 × 4
##   Early_onset count `format(round(mean(ThTmax, na.rm = T), 2), 2)`      sd
##   <fct>        <int> <chr>                                         <dbl>
## 1 Late-onset    14  150343.7                                     47846.
## 2 Young-onset    8   153330.4                                     42902.
```

```
RTposdf %>% summarize(count=n(), format(round(mean(ThTmax, na.rm=T),2),2), sd=sd(ThTmax, na.rm=T))
```

```
##   count format(round(mean(ThTmax, na.rm = T), 2), 2)      sd
## 1    22                           151429.8 45087.45
```

THT STATISTICS: TTEST

```
t.test(RTposdf$ThTmax ~ RTposdf$DX_APD, var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data: RTposdf$ThTmax by RTposdf$DX_APD
## t = 0.013211, df = 20, p-value = 0.9896
## alternative hypothesis: true difference in means between group CBS and group PSP is not equal to 0
## 95 percent confidence interval:
## -42442.23 42983.26
## sample estimates:
## mean in group CBS mean in group PSP
##           151528.1          151257.6
```

```
t.test(RTposdf$ThTmax ~ RTposdf$Early_onset, var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data: RTposdf$ThTmax by RTposdf$Early_onset
## t = -0.14594, df = 20, p-value = 0.8854
## alternative hypothesis: true difference in means between group Late-onset and group Young-onset is not equal to 0
## 95 percent confidence interval:
## -45676.87 39703.55
## sample estimates:
## mean in group Late-onset mean in group Young-onset
##           150343.7          153330.4
```

```
t.test(RTposdf$ThTmax ~ RTposdf$AD, var.equal=TRUE)
```

```
## 
## Two Sample t-test
##
## data: RTposdf$ThTmax by RTposdf$AD
## t = 0.021702, df = 20, p-value = 0.9829
## alternative hypothesis: true difference in means between group AD Negative and group
## AD Positive is not equal to 0
## 95 percent confidence interval:
## -43654.26 44572.15
## sample estimates:
## mean in group AD Negative mean in group AD Positive
## 151575.8 151116.9
```

THT STATISTICS: BARPLOTS

```
ggplot(RTposdf, aes(x=DX_APD, y=ThTmax, color=DX_APD)) + #No need for color or fill

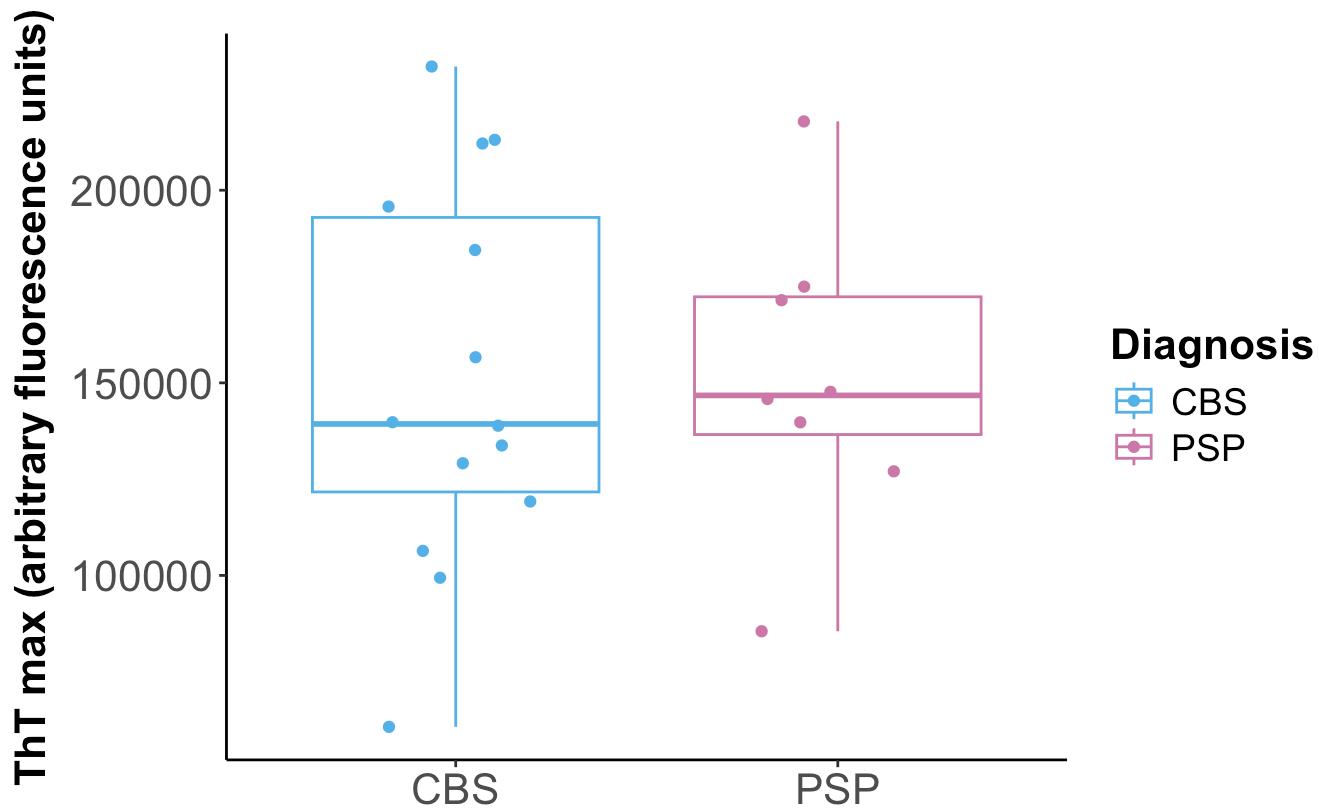
#Add actual datapoints
geom_boxplot() +
  geom_jitter(aes(color=DX_APD), position=position_jitterdodge()) +

# Fix legends and set up the appearance of the points
scale_color_manual(values=cbPalette_DX_APD, name="Diagnosis", breaks=c("CBS", "PS
P"), labels=c("CBS", "PSP")) + #expression allows you to add greek letters

# Fix labs
  labs(title=expression("ThT max by diagnosis"),
       subtitle="",
       x="", y=expression(bold("ThT max (arbitrary fluorescence units)"))) + #bold() is
required otherwise the Y axis will not be bold in spite of element_text specification be
low

#Aesthetic only #Needs to be after Annotate() otherwise messes with the size of the
asterisk
  theme_classic() +
  theme(plot.title = element_text(size=16, hjust=0.5, face="bold")) +
  theme(axis.text=element_text(size=16), axis.title=element_text(size=16, face="bold"))
+
  theme(legend.title = element_text(face="bold", size=16), legend.text= element_text(s
ize=14))
```

ThT max by diagnosis



```
ggplot(RTposdf, aes(x=AD, y=ThTmax, color=AD)) + #No need for color or fill

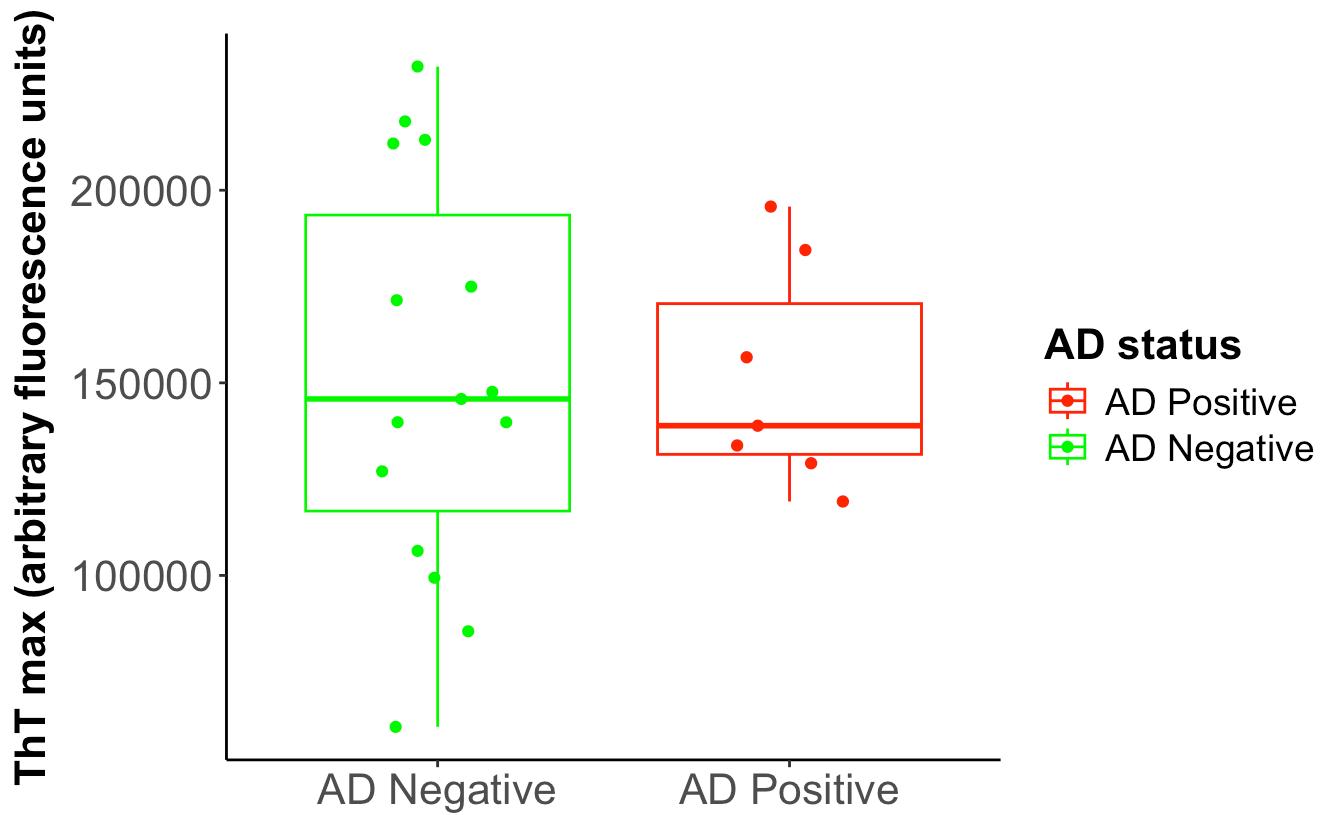
  #Add actual datapoints
  geom_boxplot() +
  geom_jitter(aes(color=AD), position=position_jitterdodge()) +

  # Fix legends and set up the appearance of the points
  scale_color_manual(values=c("red", "green"), name="AD status", breaks=c("AD Positive", "AD Negative"), labels=c("AD Positive", "AD Negative")) + #expression allows you to add greek letters

  # Fix labs
  labs(title=expression("ThT max by AD status"),
       subtitle="",
       x="", y=expression(bold("ThT max (arbitrary fluorescence units)"))) + #bold() is required otherwise the Y axis will not be bold in spite of element_text specification below

  #Aesthetic only #Needs to be after Annotate() otherwise messes with the size of the asterisk
  theme_classic() +
  theme(plot.title = element_text(size=16, hjust=0.5, face="bold")) +
  theme(axis.text=element_text(size=16), axis.title=element_text(size=16, face="bold"))
+
  theme(legend.title = element_text(face="bold", size=16), legend.text= element_text(size=14))
```

ThT max by AD status



```
ggplot(RTposdf, aes(x=Early_onset, y=ThTmax, color=Early_onset))+ #No need for color or fill

#Add actual datapoints
geom_boxplot() +
  geom_jitter(aes(color=Early_onset), position=position_jitterdodge()) +

# Fix legends and set up the appearance of the points
scale_color_manual(values=c("hotpink1", "royalblue1"), name="Onset", breaks=c("Young-onset", "Late-onset"), labels=c("Young-onset", "Late-onset")) + #expression allows you to add greek letters

# Fix labs
labs(title=expression("ThT max by type of onset"),
     subtitle="",
     x="", y=expression(bold("ThT max (arbitrary fluorescence units)")))+ #bold() is required otherwise the Y axis will not be bold in spite of element_text specification below

#Aesthetic only #Needs to be after Annotate() otherwise messes with the size of the asterisk
theme_classic() +
  theme(plot.title = element_text(size=16, hjust=0.5, face="bold")) +
  theme(axis.text=element_text(size=16), axis.title=element_text(size=16, face="bold"))
+
  theme(legend.title = element_text(face="bold", size=16), legend.text= element_text(size=14))
```

ThT max by type of onset

