COMP 4441 Final Project

Exploratory Analyses, Parametric Statistics, and Classifiers

Jonathan Ramos

An Alzheimer's Disease (AD) diagnosis can only be confirmed through post-mortem analyses of neurofibrillary tangles and plaque deposits, and so it is necessary to identify alternative prospective biomarkers that can aid in the classification of patients as demented or non-demented. It has been shown that over time, due to the atrophies associated with the disease progression of AD, patients with AD have a lighter brain mass than patients without AD. Although we cannot measure this directy, we can estimate it through MRI. From images gathered via MRI, we can estimate the patient's whole brain volume by masking an image with a brain atlas and counting the percent of pixels that are labeled as either grey or white matter. In this data set normalized whole brain volume (nWBV) as well as other potentially useful metrics were measured in 150 participants aged 60-96. participants were scanned 2 or more times with each visit separated by at least one year. Of the 150 participants, 72 were characterized as nondemented for the duration of the study, 64 were characterized as demented throughout the study and 14 were characterized as non-demented during the initial visit but were later characterized as demented at a later visit.

This Rmd file is split into three sections: 1. Exploratory Analyses 2. Parametric Statistics 3. Classifiers

Quick overview of variables:

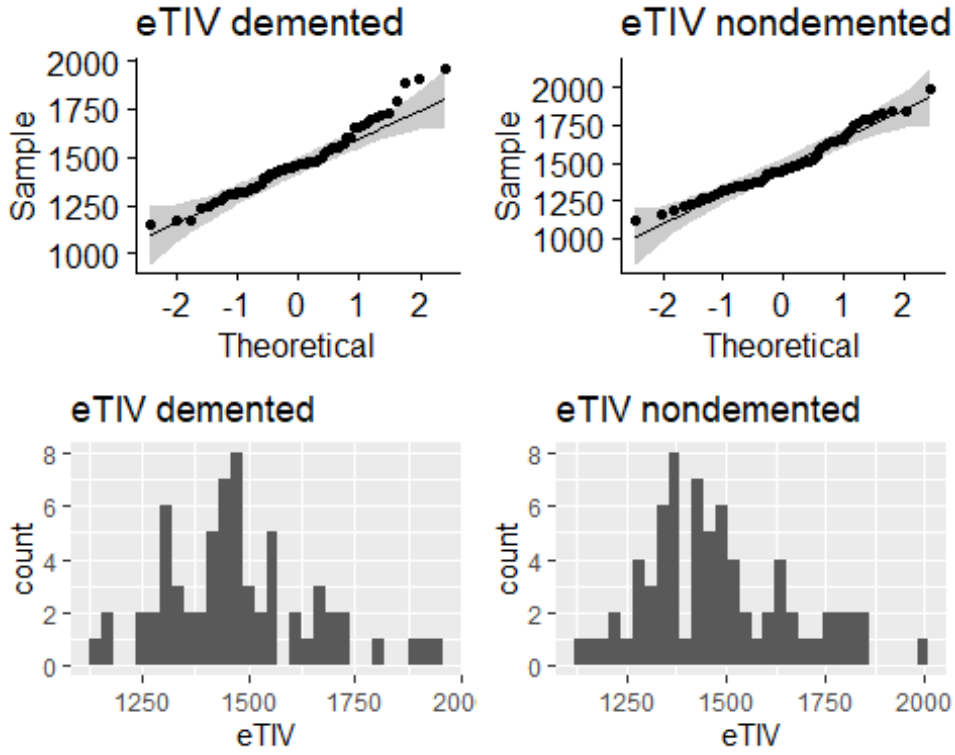
- Subject.ID
- MRI.ID
- Group
- Visit visit number (at least 2 per subject)
- MR.Delay days between last visit (at least one year between visits)
- M.F male / female
- Hand handedness (all participants were right-hanted)
- Age
- EDUC years of education
- SES Hollingshead Index of Social Position, ranging from 1 (highest) to 5 (lowest)
- MMSE Mini-mental state examination score, ranging from 0 (worst) to 30 (best)
- CDR Clinical Dementia Rating: 0 (none), 0.5 (very mild AD), 1 (mild AD), 2 (moderate AD)
- eTIV Estimate total intracrantial volume, mm3
- nWBV Normalized whole-brain volume as a percent of all voxels
- ASF Atlas scaling factor

1. Exploratory Analyses

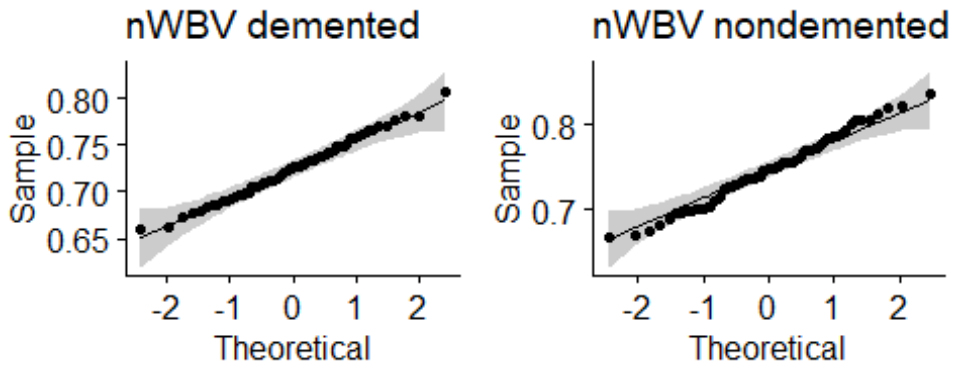
```
dat <- read.csv('oasis_longitudinal.csv')
dat.dem <- dat[dat$Group == 'Demented' & dat$Visit == 1,]
dat.ndem <- dat[dat$Group == 'Nondemented' & dat$Visit == 1,]
dat.conv <- dat[dat$Group == 'Converted' & dat$Visit == 1,]
```

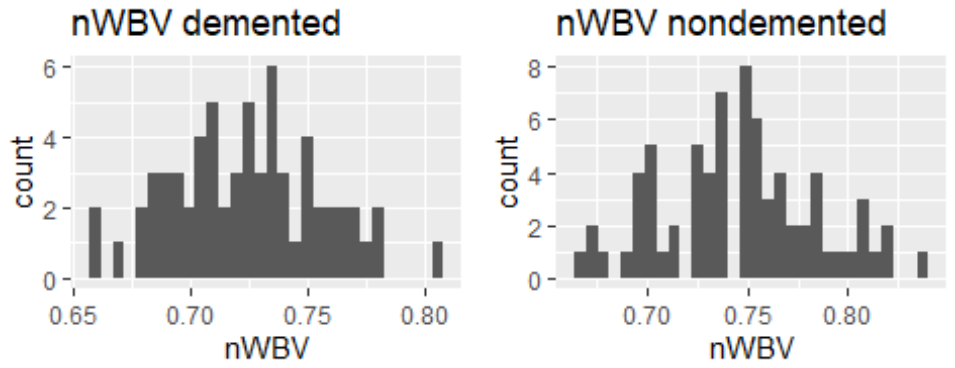
Let's just make some plots

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

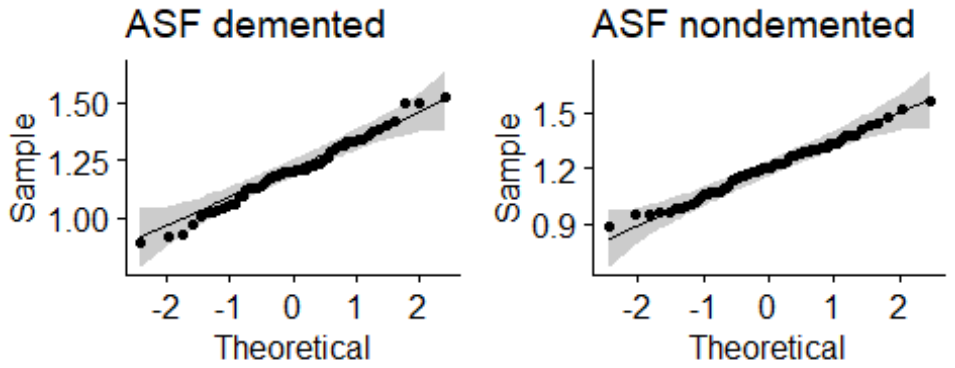


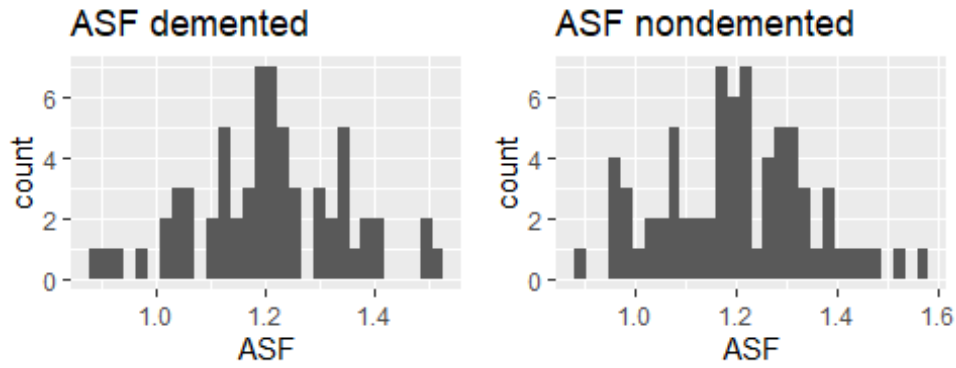

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





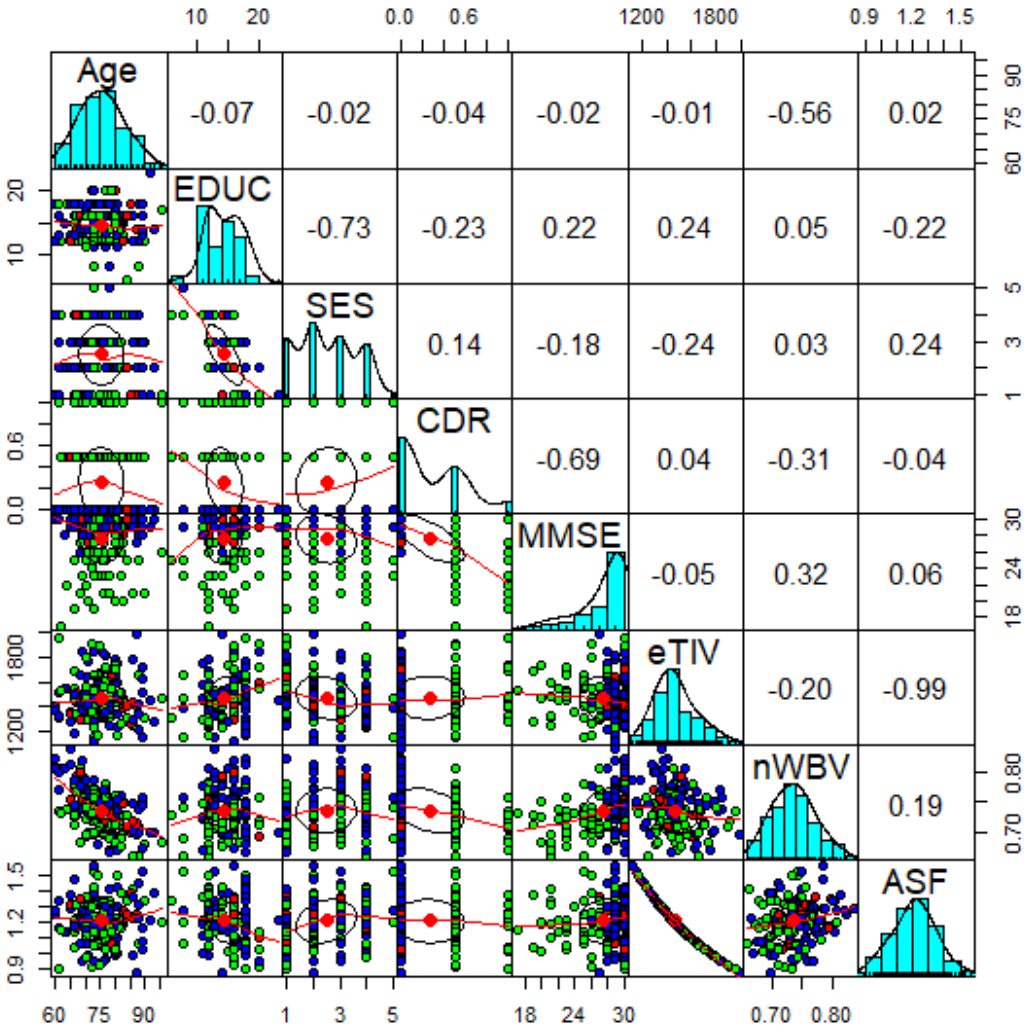

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





Ok great now let's just make a pair plot to get a sense for whether any of our more interesting variables are correlated or not.

```
# dat.init contains only data from the first visit of all participants
# I chose to only analyze the beginning of the set because some patients
visit more than others and maybe over represented in the set if I take all
timepoints from all patients.

dat.init <- dat[dat$Visit == 1,]
pairs.panels(dat.init[c('Age','EDUC','SES','CDR','MMSE','eTIV',
'nWBV','ASF')], pch=21,
             bg=c('red','green','blue')[factor(dat.init$Group)], gap=0)
```




```
# red is converted (from non demented at init, to demented at a subsequent
visit)
# green is demented
# blue is nondemented
```

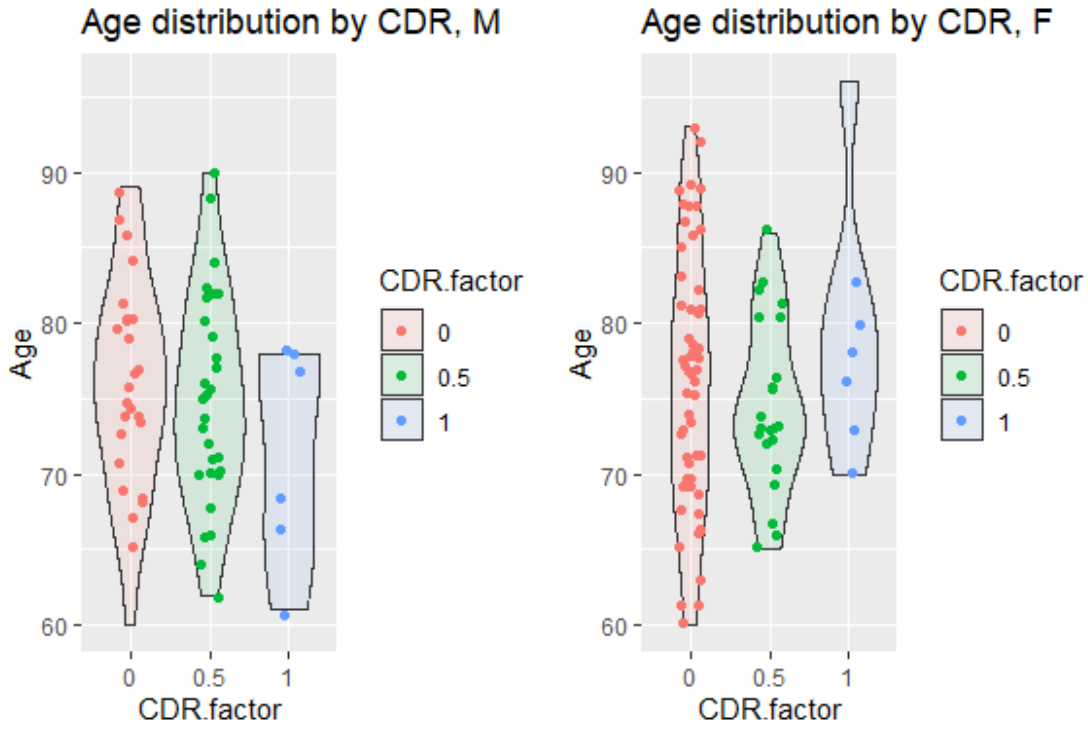
Since the label 'Demented' or 'Nondemented' is generated directly from CDR we will further examine CDR. A demented individual is anyone whose CDR is greater than 0, 0 being characterized as having no dementia, 0.5 very mild dementia, 1.0 mild dementia and 2.0 moderate dementia.

Let's see if CDR varies with any of our demographic variables:

```
dat.init$CDR.factor <- as.factor(dat.init$CDR)
v1 <- ggplot(dat.init[dat.init$M.F == 'M',], aes(x=CDR.factor, y=Age,
fill=CDR.factor)) +
  geom_violin(alpha=0.1) +
  geom_jitter(shape=21, position=position_jitter(0.15), aes(col=CDR.factor))
+
  ylim(60,96) +
  ggtitle('Age distribution by CDR, M')

v2 <- ggplot(dat.init[dat.init$M.F == 'F',], aes(x=CDR.factor, y=Age,
fill=CDR.factor)) +
  geom_violin(alpha=0.1) +
  geom_jitter(shape=21, position=position_jitter(0.15), aes(col=CDR.factor))
+
  ylim(60,96) +
  ggtitle('Age distribution by CDR, F')

v3 <- ggplot(dat.init, aes(x=CDR.factor, y=EDUC, fill=CDR.factor)) +
geom_violin(alpha=0.1) +
  geom_jitter(shape=21, position=position_jitter(0.15), aes(col=CDR.factor))
+
  ggtitle('EDUC distribution by CDR')

v4 <- ggplot(dat.init, aes(x=CDR.factor, y=SES, fill=CDR.factor)) +
geom_violin(alpha=0.1) +
  geom_jitter(shape=21, position=position_jitter(0.15), aes(col=CDR.factor))
+
  ggtitle('SES distribution by CDR')

v5 <- ggplot(dat.init, aes(x=CDR.factor, y=MMSE, fill=CDR.factor)) +
geom_violin(alpha=0.1) +
  geom_jitter(shape=21, position=position_jitter(0.15), aes(col=CDR.factor))
+
  ggtitle('MMSE distribution by CDR')

v6 <- ggplot(dat.init, aes(x=CDR.factor, y=eTIV, fill=CDR.factor)) +
geom_violin(alpha=0.1) +
  geom_jitter(shape=21, position=position_jitter(0.15), aes(col=CDR.factor))
+
  ggtitle('eTIV distribution by CDR')

v7 <- ggplot(dat.init, aes(x=CDR.factor, y=nWBV, fill=CDR.factor)) +
```

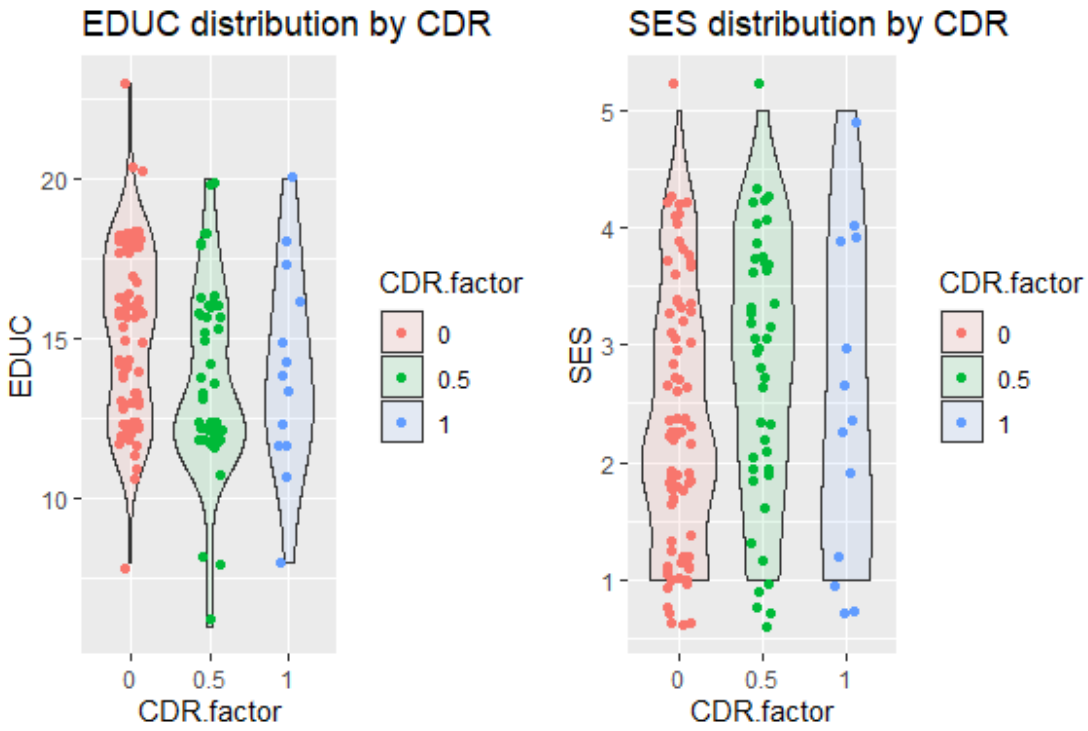
```
geom_violin(alpha=0.1) +
  geom_jitter(shape=21, position=position_jitter(0.15), aes(col=CDR.factor))
+
  ggtitle('nWBV distribution by CDR')

v8 <- ggplot(dat.init, aes(x=CDR.factor, y=ASF, fill=CDR.factor)) +
geom_violin(alpha=0.1) +
  geom_jitter(shape=21, position=position_jitter(0.15), aes(col=CDR.factor))
+
  ggtitle('ASF distribution by CDR')

grid.arrange(v1, v2, ncol=2)
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
## Removed 1 rows containing missing values (`geom_point()`).
```

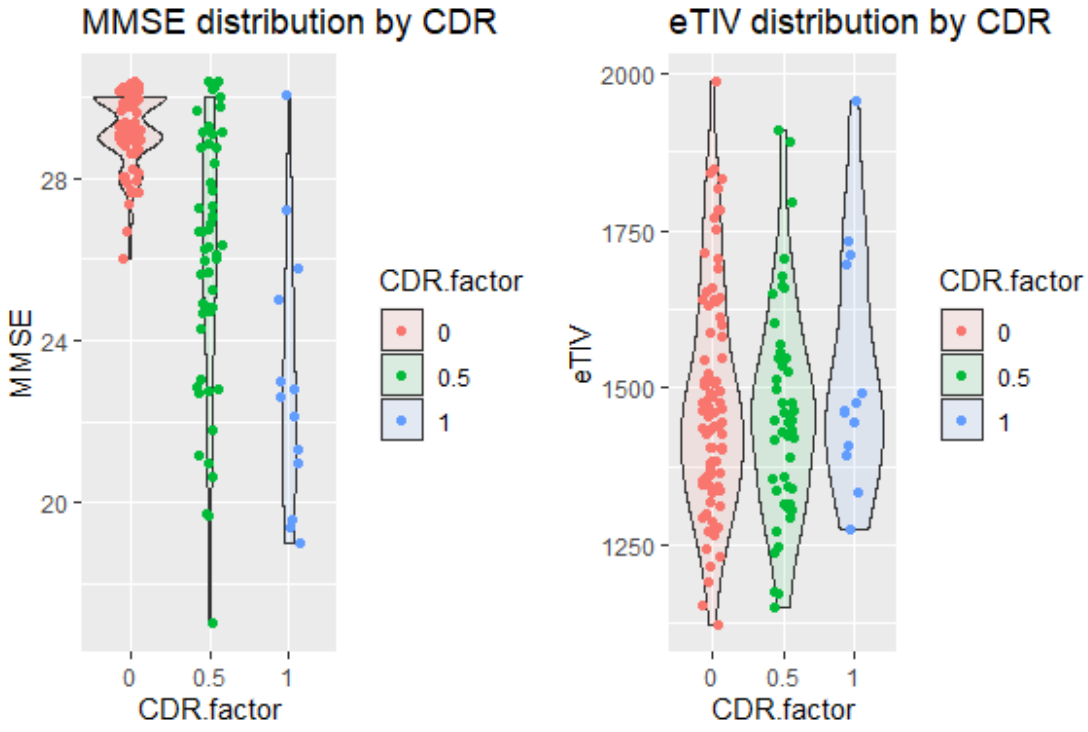



```
grid.arrange(v3, v4, ncol=2)
```

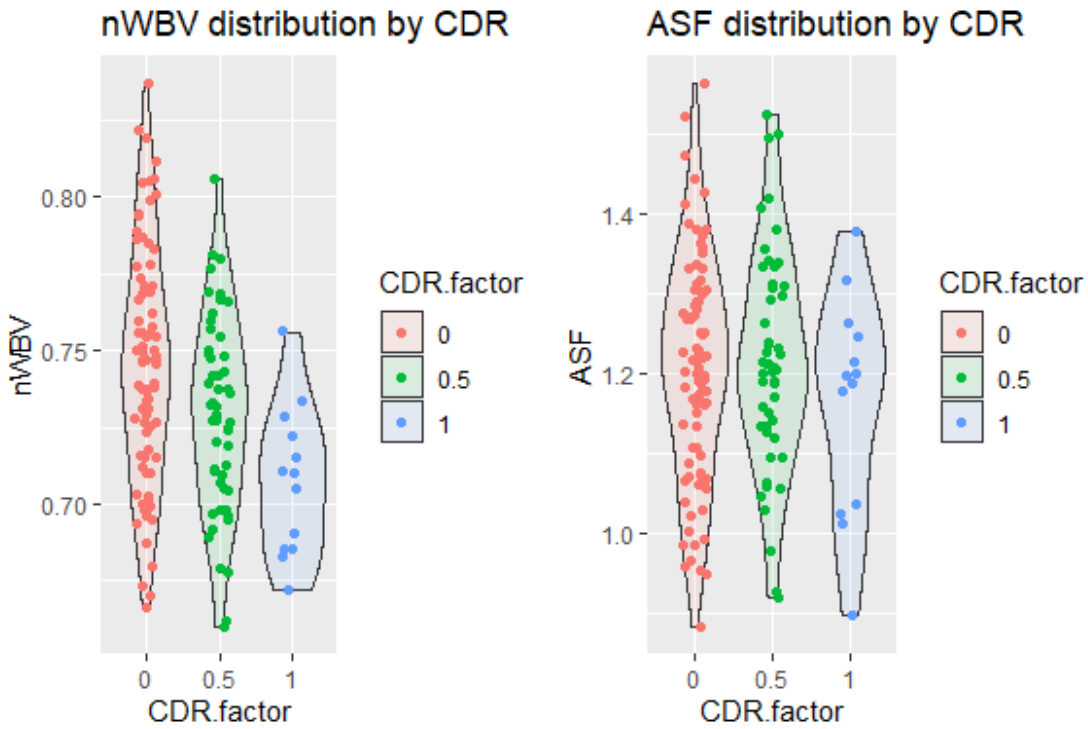
```
## Warning: Removed 8 rows containing non-finite values (`stat_ydensity()`).

## Warning: Removed 8 rows containing missing values (`geom_point()`).
```




```
grid.arrange(v5, v6, ncol=2)
```




```
grid.arrange(v7, v8, ncol=2)
```

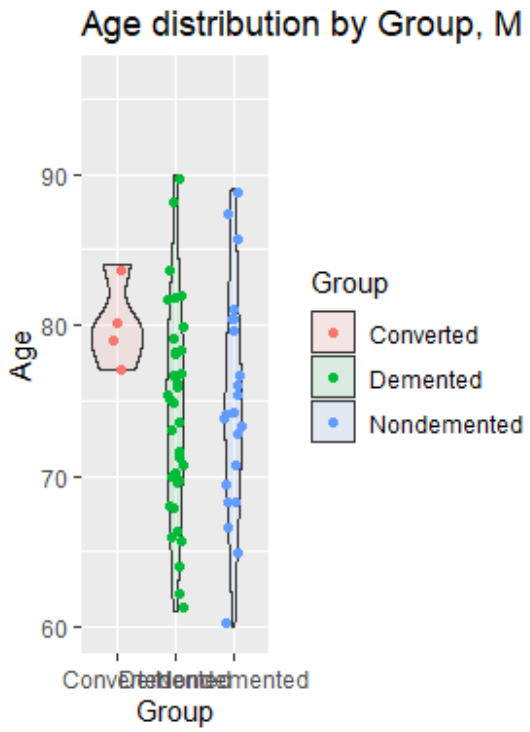


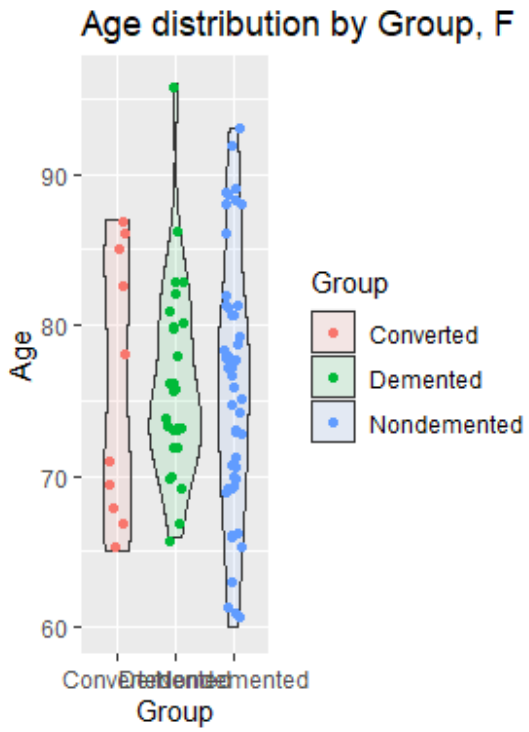
Patients belonging to the Demented group went through the study with a CDR of at least 0.5 for all visits whereas patients belonging to the Nondemented group went through with a CDR of 0 for all visits. Lastly, a third group arose called Converted where a patient began the study with a CDR of 0 but recieved a CDR of at least 0.5 at a later subsequent visit. Because of the way these labels are defined, there is an alternative way to try and understand the distributions above: we can group by the labels in the Group column rather than by CDR. Let's repeat the visualizations above but instead grouping by Group.

```
dat.init$CDR.factor <- as.factor(dat.init$CDR)
v1 <- ggplot(dat.init[dat.init$M.F == 'M',], aes(x=Group, y=Age, fill=Group))
+
  geom_violin(alpha=0.1) +
  geom_jitter(shape=21, position=position_jitter(0.15), aes(col=Group)) +
  ylim(60,96) +
  ggtitle('Age distribution by Group, M')

v2 <- ggplot(dat.init[dat.init$M.F == 'F',], aes(x=Group, y=Age, fill=Group))
+
  geom_violin(alpha=0.1) +
  geom_jitter(shape=21, position=position_jitter(0.15), aes(col=Group)) +
  ylim(60,96) +
  ggtitle('Age distribution by Group, F')

v3 <- ggplot(dat.init, aes(x=Group, y=EDUC, fill=Group)) +
geom_violin(alpha=0.1) +
  geom_jitter(shape=21, position=position_jitter(0.15), aes(col=Group)) +
  ggtitle('EDUC distribution by Group')
```

```r
v4 <- ggplot(dat.init, aes(x=Group, y=SES, fill=Group)) +
geom_violin(alpha=0.1) +
  geom_jitter(shape=21, position=position_jitter(0.15), aes(col=Group)) +
  ggtitle('SES distribution by Group')

v5 <- ggplot(dat.init, aes(x=Group, y=MMSE, fill=Group)) +
geom_violin(alpha=0.1) +
  geom_jitter(shape=21, position=position_jitter(0.15), aes(col=Group)) +
  ggtitle('MMSE distribution by Group')

v6 <- ggplot(dat.init, aes(x=Group, y=eTIV, fill=Group)) +
geom_violin(alpha=0.1) +
  geom_jitter(shape=21, position=position_jitter(0.15), aes(col=Group)) +
  ggtitle('eTIV distribution by Group')

v7 <- ggplot(dat.init, aes(x=Group, y=nWBV, fill=Group)) +
geom_violin(alpha=0.1) +
  geom_jitter(shape=21, position=position_jitter(0.15), aes(col=Group)) +
  ggtitle('nWBV distribution by Group')

v8 <- ggplot(dat.init, aes(x=Group, y=ASF, fill=Group)) +
geom_violin(alpha=0.1) +
  geom_jitter(shape=21, position=position_jitter(0.15), aes(col=Group)) +
  ggtitle('ASF distribution by Group')

grid.arrange(v1, v2, ncol=2)
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```

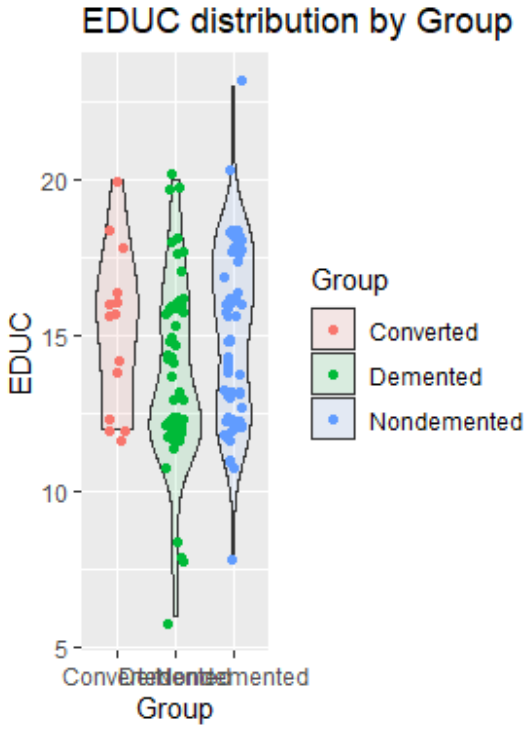


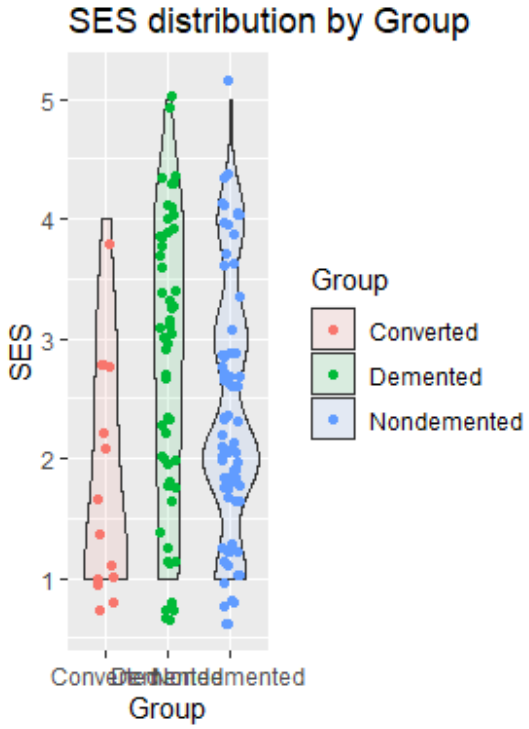



```
grid.arrange(v3, v4, ncol=2)
```

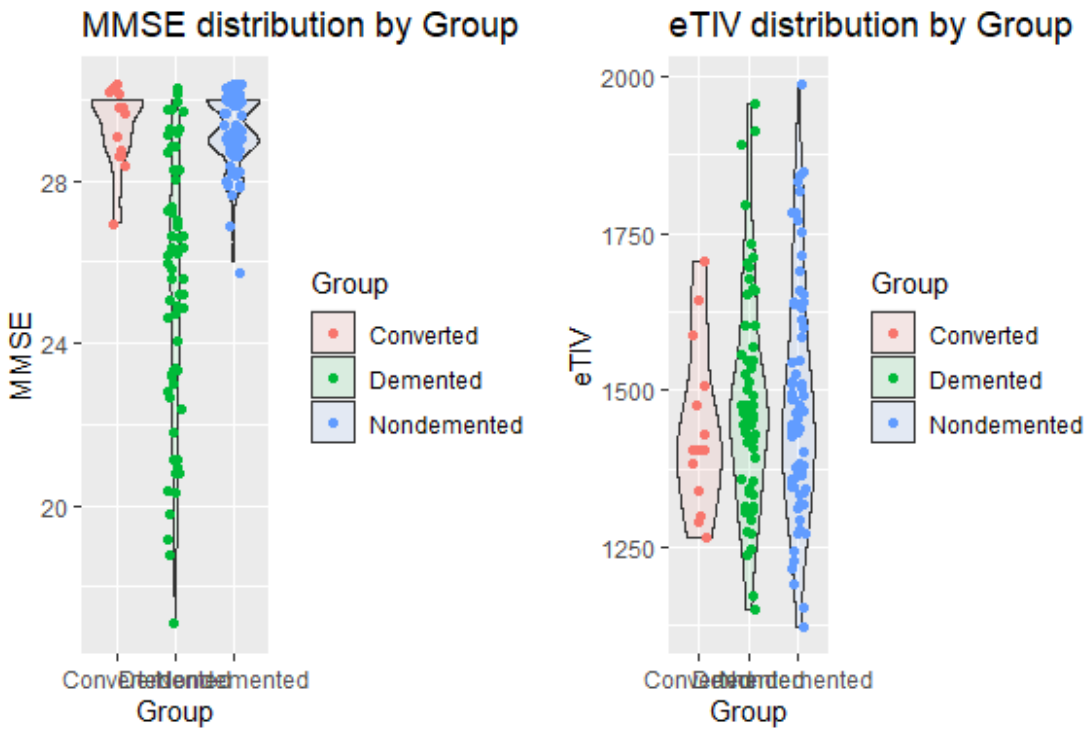
```
## Warning: Removed 8 rows containing non-finite values (`stat_ydensity()`).
```

```
## Warning: Removed 8 rows containing missing values (`geom_point()`).
```

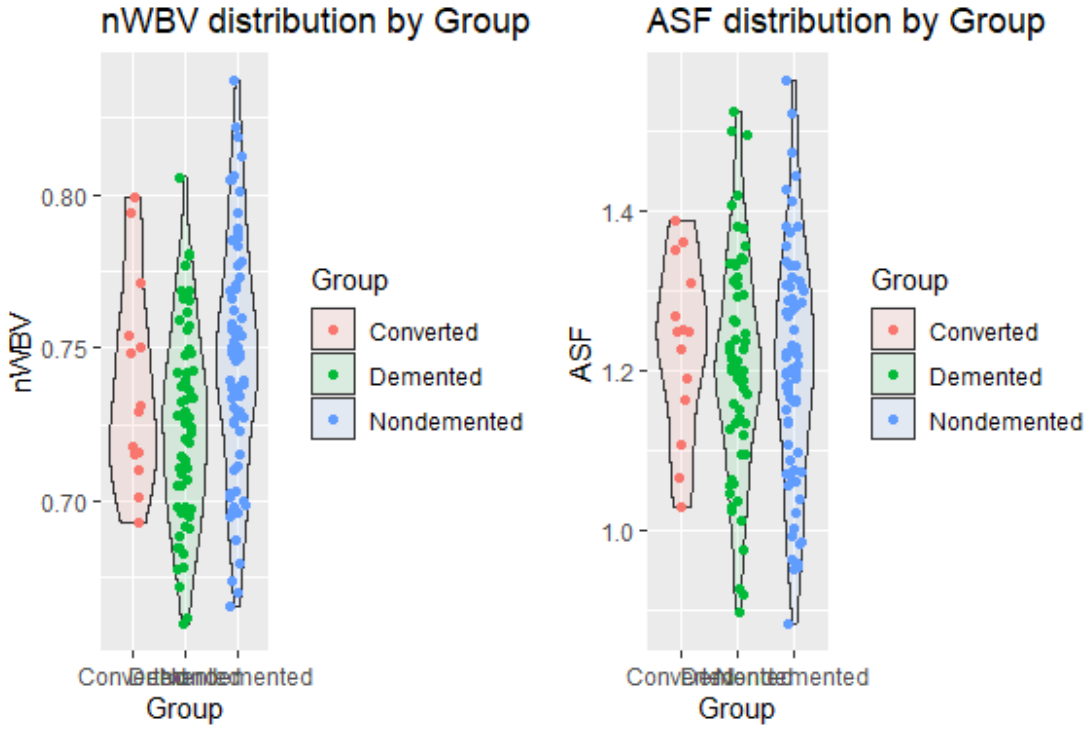




```
grid.arrange(v5, v6, ncol=2)
```




```
grid.arrange(v7, v8, ncol=2)
```

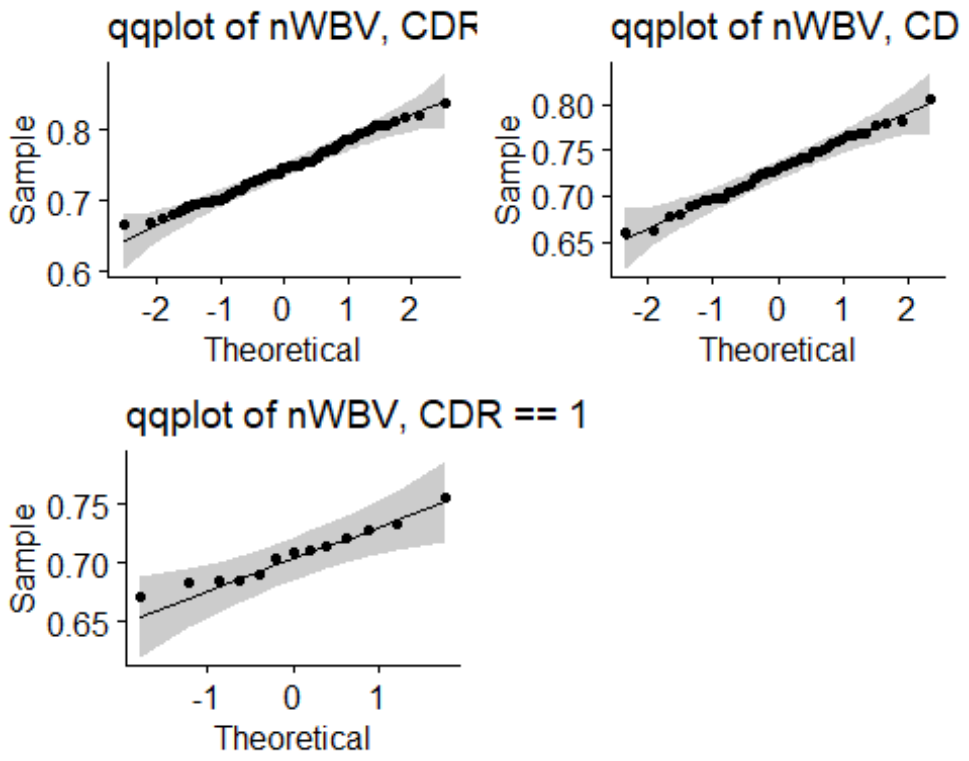


From the ggqqplots, histograms, and violin plots above, we can see that the data suggests that normalized whole brain volume might be different in Demented vs Non-demented participants Because the data are normally distributed, it is appropriate to run a t-test to test if the center of the distributions for nWBV for demented vs non demented are different for participants who maintained the same demented status throughout the duration of the study. From the t-test below, if we set alpha=0.05, we may reject the null hypothesis that the means nWBV for demented and non-demented participants are not different. We can therefore conclude that demented participants have a slightly lower mean nWBV than non-demented patients.

2. Parametric Statistics

Do patient's with different CDR have different nWBV?

```
g1 <- ggqqplot(dat.init[dat.init$CDR == 0, 'nWBV']) + ggtitle('qqplot of
nWBV, CDR == 0')
g2 <- ggqqplot(dat.init[dat.init$CDR == 0.5, 'nWBV']) + ggtitle('qqplot of
nWBV, CDR == 0.5')
g3 <- ggqqplot(dat.init[dat.init$CDR == 1, 'nWBV']) + ggtitle('qqplot of
nWBV, CDR == 1')
grid.arrange(g1, g2, g3, ncol=2)
```




```
shapiro.test(dat.init[dat.init$CDR == 0, 'nWBV'])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dat.init[dat.init$CDR == 0, "nWBV"]
## W = 0.98932, p-value = 0.7167
```

```
shapiro.test(dat.init[dat.init$CDR == 0.5, 'nWBV'])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dat.init[dat.init$CDR == 0.5, "nWBV"]
## W = 0.99253, p-value = 0.985
```

```
shapiro.test(dat.init[dat.init$CDR == 1, 'nWBV'])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dat.init[dat.init$CDR == 1, "nWBV"]
## W = 0.96389, p-value = 0.8124
```

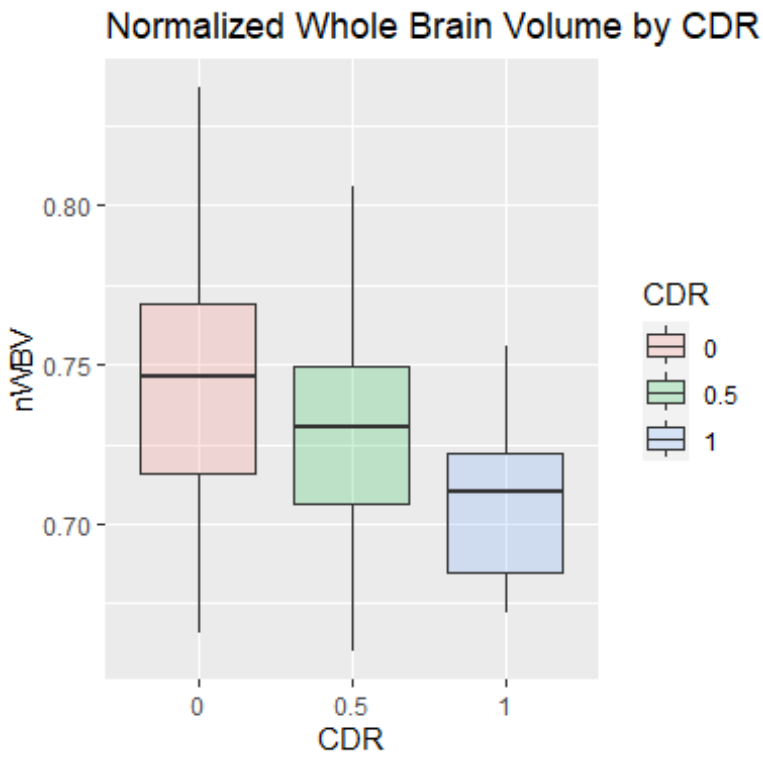
Great, based off of the qqplots and shapiro wilk tests for normality, it is appropriate to consider these groups as coming from a normal distribution. We can therefore perform an ANOVA to compare 3 ways if the means are different. From the violin plots above we can see that the variances are not all equal.

```
oneway.test(nWBV ~ CDR.factor,
            data=dat.init,
            var.equal = FALSE)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  nWBV and CDR.factor
## F = 11.53, num df = 2.000, denom df = 38.875, p-value = 0.0001171
```

From the one way ANOVA above, if we set alpha=0.05, we reject that null hypothesis that the means are all equal. We may now conclude that at least one of the means for nWBV is different across CDR. Now we should run some post hoc t-tests to see which CDR.factor groups are different from each other for nWBV. Since we are making 3 comparisons, to account for the likelihood of false positives during multiple testing, we will use the Bonferroni corrected p value which is 0.5 / 3 = 0.0167. As we can see from the t-tests below, all p-values are less than the Bonferroni corrected p of 0.0167 so we can therefore conclude that all CDR groups are different from each other in terms of nWBV.

```
dat.init$CDR <- as.factor(dat.init$CDR)
ggplot(dat.init, aes(x=CDR, y=nWBV, fill=CDR)) +
  geom_boxplot(alpha=0.2) +
  ggtitle("Normalized Whole Brain Volume by CDR")
```




```
print(0.05/3)
```

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

```
t.test(dat.init[dat.init$CDR == 0, 'nWBV'],dat.init[dat.init$CDR == 0.5,
'nWBV'])
```

```
##
##  Welch Two Sample t-test
##
## data:  dat.init[dat.init$CDR == 0, "nWBV"] and dat.init[dat.init$CDR ==
0.5, "nWBV"]
## t = 2.5728, df = 122.62, p-value = 0.01128
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.003566629 0.027366403
## sample estimates:
## mean of x mean of y
## 0.7446588 0.7291923
```

```
t.test(dat.init[dat.init$CDR == 0.5, 'nWBV'],dat.init[dat.init$CDR == 1,
'nWBV'])
```

```
##
##  Welch Two Sample t-test
##
## data:  dat.init[dat.init$CDR == 0.5, "nWBV"] and dat.init[dat.init$CDR ==
1, "nWBV"]
```

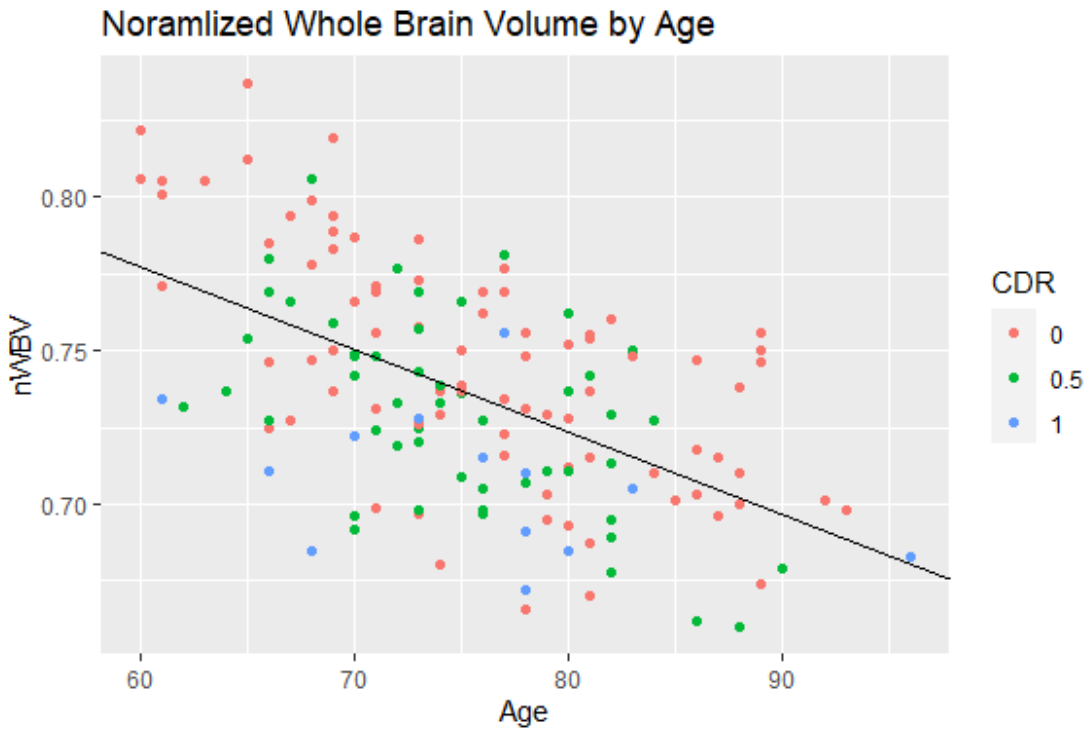
```
## t = 2.7265, df = 23.624, p-value = 0.01186
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.005267378 0.038194161
## sample estimates:
## mean of x mean of y
## 0.7291923 0.7074615
```

```
t.test(dat.init[dat.init$CDR == 1, 'nWBV'],dat.init[dat.init$CDR == 0,
'nWBV'])
```

```
##
##  Welch Two Sample t-test
##
## data:  dat.init[dat.init$CDR == 1, "nWBV"] and dat.init[dat.init$CDR == 0,
"nWBV"]
## t = -4.754, df = 22.448, p-value = 9.115e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.05340538 -0.02098919
## sample estimates:
## mean of x mean of y
## 0.7074615 0.7446588
```

During our exploratory analyses above, the pair plot suggested that nWBV may be corrolated with Age.

```
nWBV.Age.lm <- lm(nWBV ~ Age, data = dat.init)
m <- nWBV.Age.lm$coefficients[2]
b <- nWBV.Age.lm$coefficients[1]

ggplot(dat.init, aes(x=Age, y=nWBV, color=CDR)) +
  geom_point() +
  geom_abline(slope = m, intercept = b) +
  ggtitle("Noramlized Whole Brain Volume by Age")
```

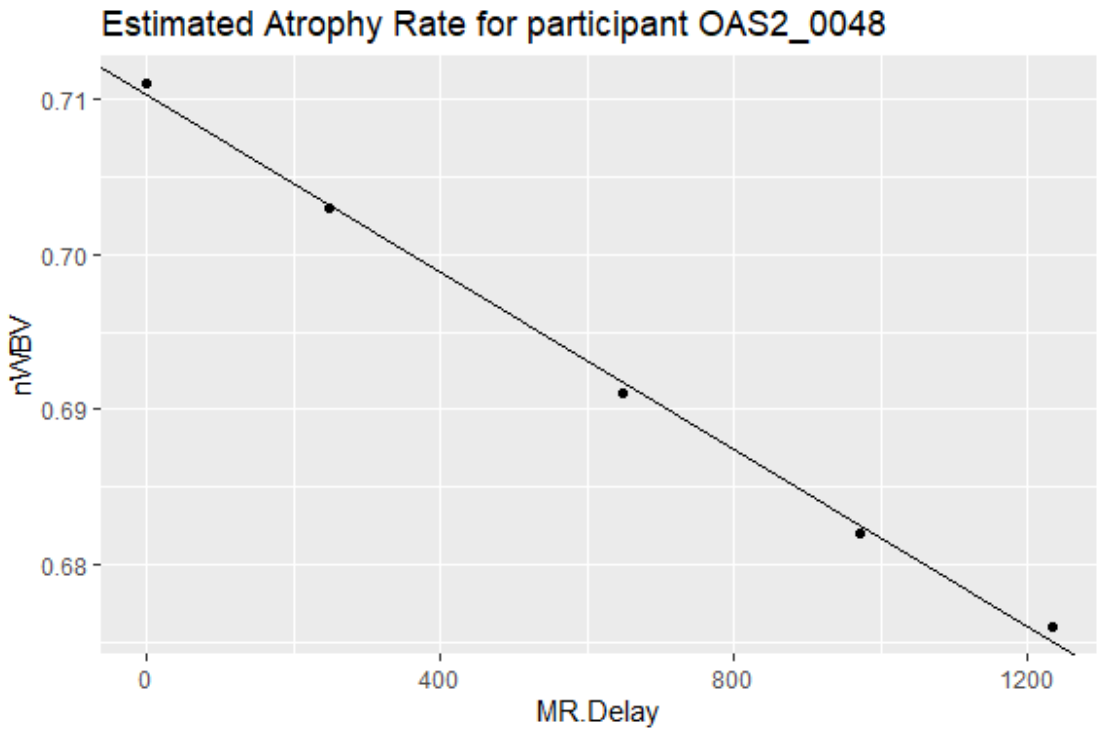



```
summary(nWBV.Age.lm)

##
## Call:
## lm(formula = nWBV ~ Age, data = dat.init)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.071152 -0.022550 -0.000974  0.025509  0.072759
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.9395002  0.0249893  37.596  < 2e-16 ***
## Age         -0.0026963  0.0003296  -8.181 1.18e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03036 on 148 degrees of freedom
## Multiple R-squared:  0.3114, Adjusted R-squared:  0.3067
## F-statistic: 66.93 on 1 and 148 DF,  p-value: 1.183e-13

cor.test(dat.init$nWBV, dat.init$Age, method="pearson")

##
##  Pearson's product-moment correlation
##
## data:  dat.init$nWBV and dat.init$Age
## t = -8.1809, df = 148, p-value = 1.183e-13
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6593248 -0.4368312
## sample estimates:
##        cor
## -0.5580268
```

Do participants in the demented group experience a greater loss in nWBV than nondemented participants? Since this was a longitudinal study we can assess the unique rate of atrophy for each patient and then compare across groups. It seems straightforward to take the difference between the first and last visit and compare distributions of differences; however some participants visited more than others and the number of days between visits is not consistent. This means that differences in the difference between the first and last visit may just be a result of the difference in time between the first and last visit. To account for this I will fit a line to each patient for nWBV and compare the slopes of these lines. This means that we can estimate the rate of atrophy with the slope of the linear regression line for each patient.

```
dat.lm <- as.data.frame(dat %>% group_by(Subject.ID) %>% do(tidy(lm(nWBV ~
MR.Delay, dat=.))))
dat.slopes <- dat.lm[dat.lm$term == 'MR.Delay',]

# let's just run a quick test on 1 participant to confirm that we got what we
expect

# using lm to get estimate slope the usual way,
df.test <- dat[dat$Subject.ID == 'OAS2_0048',]
coef <- lm(nWBV ~ MR.Delay, data = df.test)$coefficients
ggplot(data = df.test,aes(x=MR.Delay, y=nWBV)) +
  geom_point() +
  geom_abline(slope = coef[2], intercept = coef[1]) +
  ggtitle("Estimated Atrophy Rate for participant OAS2_0048")
```

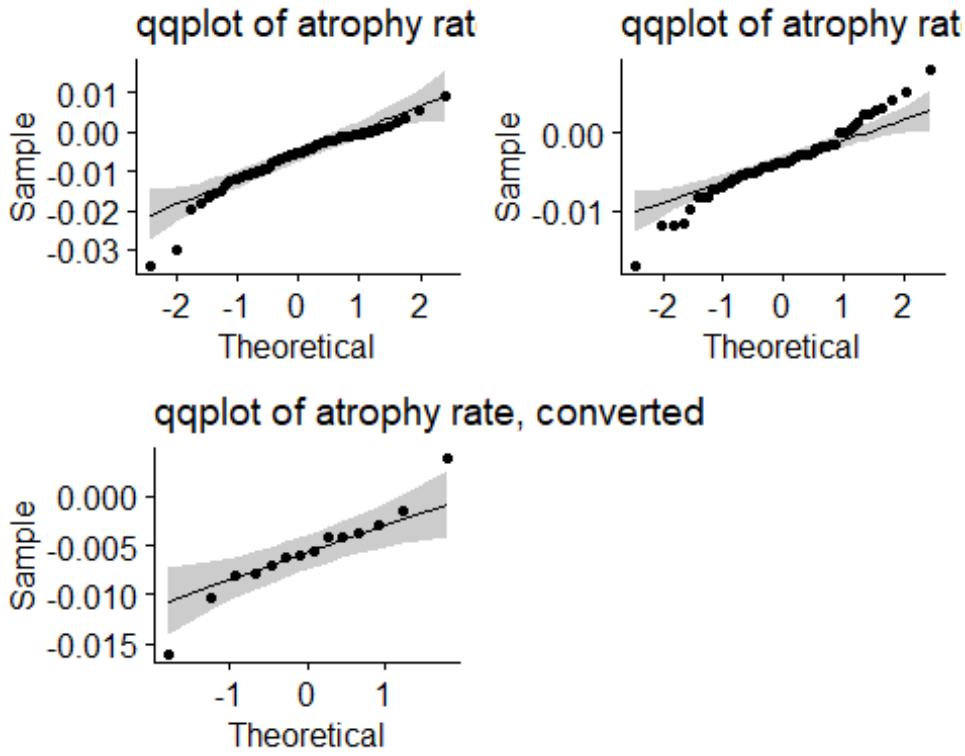



```
# we expect true if both methods yield the same result
(coef[2] == dat.slopes[dat.slopes$Subject.ID == 'OAS2_00048',]$estimate)

## logical(0)

# now let's just add in a column of grpi[ labels to our slope data frame
dat.group <- distinct(dat[,c('Subject.ID','Group')])
dat.slopes <- merge(dat.slopes, dat.group, by='Subject.ID')
dat.slopes$estimate <- dat.slopes$estimate *365

# splitting up slopes by group
# Demented means that CDR remained at least 0.5 for the duration of the study
# Nondemented means that CDR remained 0 for the duration of the study (no AD)
# Converted means that CDR was 0 initially then increased to at least 0.5 by
the end of the study.

slope.dem <- dat.slopes[dat.slopes$Group == 'Demented',]
slope.ndem <- dat.slopes[dat.slopes$Group == 'Nondemented',]
slope.conv <- dat.slopes[dat.slopes$Group == 'Converted',]
```

Now I just want to check if the slopes for atrophy rate are normally distributed. From the qqplots below we can see that the converted and demented slopes are roughly normally distributed. The nondemented slopes deviate from normality more, but the n size is high enough that these deviations from normality likely won't matter.

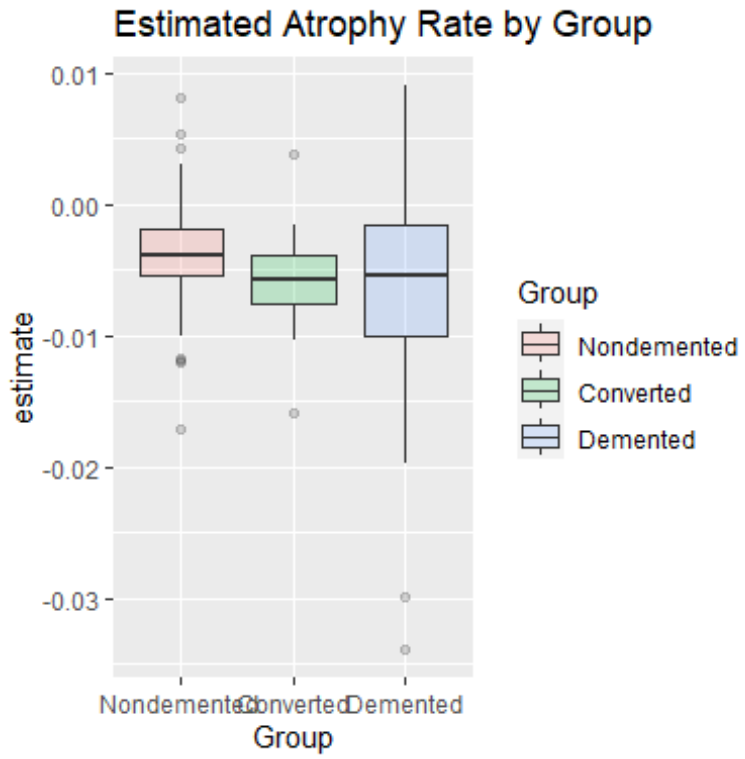
```
g1 <- ggqqplot(slope.dem$estimate) + ggtitle('qqplot of atrophy rate,
demented')
g2 <- ggqqplot(slope.ndem$estimate) + ggtitle('qqplot of atrophy rate,
```

```
nondemented')
g3 <- ggqqplot(slope.conv$estimate) + ggtitle('qqplot of atrophy rate,
converted')

grid.arrange(g1, g2, g3, ncol=2)
```




```
oneway.test(estimate ~ Group,
            var.equal = FALSE,
            data=dat.slopes)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  estimate and Group
## F = 4.1303, num df = 2.000, denom df = 36.864, p-value = 0.02407

# a refresher of Bonferroni's corrected p value for 3 comparisons
print(0.05/3)

## [1] 0.01666667

t.test(slope.dem$estimate, slope.ndem$estimate)

##
##  Welch Two Sample t-test
##
## data:  slope.dem$estimate and slope.ndem$estimate
## t = -2.6662, df = 96.172, p-value = 0.009001
```

```
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.0047962373 -0.0007024829
## sample estimates:
##    mean of x    mean of y
## -0.006408423 -0.003659063

t.test(slope.ndem$estimate, slope.conv$estimate)

##
##  Welch Two Sample t-test
##
## data:  slope.ndem$estimate and slope.conv$estimate
## t = 1.5855, df = 17.432, p-value = 0.1308
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.0006714251  0.0047634559
## sample estimates:
##    mean of x    mean of y
## -0.003659063 -0.005705078

t.test(slope.conv$estimate, slope.dem$estimate)

##
##  Welch Two Sample t-test
##
## data:  slope.conv$estimate and slope.dem$estimate
## t = 0.46709, df = 30.362, p-value = 0.6438
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.002370381  0.003777070
## sample estimates:
##    mean of x    mean of y
## -0.005705078 -0.006408423

dat.slopes$Group <- factor(dat.slopes$Group, levels=c("Nondemented",
"Converted", "Demented"))
ggplot(dat.slopes, aes(x=Group, y=estimate, fill=Group)) +
  geom_boxplot(alpha=0.2) +
  ggtitle("Estimated Atrophy Rate by Group")
```




```
# let's make a dataframe to build a model on that incorporates our newly
calculated
# estimated atrophy rate (which was estimated via linear regression)
# since the estimated atrophy rates were calculated based on the particpants
most recent visit,
# i will train and test the model based only on those data.

# here's the data only for a participant's most recent visit.
dat.recent <- dat %>% group_by(Subject.ID) %>% top_n(1, Visit)

# merging our recent data frame and slope estimates data frame
dat.recent.est <- merge(dat.slopes, dat.recent, by='Subject.ID')

# dropping unnecessary columns left over from the regression summary
to_drop = c("term","std.error", "statistic", "p.value")
dat.recent.est <- dat.recent.est[, !(names(dat.recent.est) %in% to_drop)]
head(dat.recent.est)

##   Subject.ID     estimate     Group.x        MRI.ID     Group.y Visit
MR.Delay
## 1  OAS2_0001 -0.011980306 Nondemented OAS2_0001_MR2 Nondemented     2
457
## 2  OAS2_0002 -0.006111841    Demented OAS2_0002_MR3    Demented     3
1895
## 3  OAS2_0004  0.005427509 Nondemented OAS2_0004_MR2 Nondemented     2
538
```

```
## 4  OAS2_0005 -0.001462183 Nondemented OAS2_0005_MR3 Nondemented     3
1603
## 5  OAS2_0007 -0.010625263    Demented OAS2_0007_MR4    Demented     4
1281
## 6  OAS2_0008  0.002459569 Nondemented OAS2_0008_MR2 Nondemented     2
742
##   M.F Hand Age EDUC SES MMSE CDR eTIV  nWBV   ASF
## 1   M    R  88   14   2   30 0.0 2004 0.681 0.876
## 2   M    R  80   12  NA   22 0.5 1698 0.701 1.034
## 3   F    R  90   18   3   27 0.0 1200 0.718 1.462
## 4   M    R  85   12   4   30 0.0 1699 0.705 1.033
## 5   M    R  75   16  NA   27 1.0 1372 0.710 1.279
## 6   F    R  95   14   2   29 0.0 1257 0.703 1.396
```

```
# dealing with missing data
dat.missing <- dat.recent.est[!complete.cases(dat.recent.est),]
dat.complete <- dat.recent.est[complete.cases(dat.recent.est),]

# let's take a look
(dat.missing)
```

```
##     Subject.ID      estimate  Group.x        MRI.ID  Group.y Visit
MR.Delay M.F
## 2    OAS2_0002 -6.111841e-03 Demented OAS2_0002_MR3 Demented     3
1895   M
## 5    OAS2_0007 -1.062526e-02 Demented OAS2_0007_MR4 Demented     4
1281   M
## 53   OAS2_0063 -1.191837e-02 Demented OAS2_0063_MR2 Demented     2
490   F
## 82   OAS2_0099 -5.427509e-03 Demented OAS2_0099_MR2 Demented     2
807   F
## 95   OAS2_0114 -1.152632e-02 Demented OAS2_0114_MR2 Demented     2
570   F
## 131  OAS2_0160 -6.612319e-04 Demented OAS2_0160_MR2 Demented     2
552   M
## 145  OAS2_0181 -5.181405e-05 Demented OAS2_0181_MR3 Demented     3
1107   F
## 146  OAS2_0182 -9.407216e-04 Demented OAS2_0182_MR2 Demented     2
776   M
##     Hand Age EDUC SES MMSE CDR eTIV  nWBV   ASF
## 2      R  80   12  NA   22 0.5 1698 0.701 1.034
## 5      R  75   16  NA   27 1.0 1372 0.710 1.279
## 53     R  81   12  NA   27 0.5 1453 0.721 1.208
## 82     R  83   12  NA   23 0.5 1484 0.750 1.183
## 95     R  78   12  NA   27 1.0 1309 0.709 1.341
## 131    R  78   12  NA   29 1.0 1569 0.704 1.119
## 145    R  77   12  NA   NA 1.0 1159 0.733 1.515
## 146    R  75   12  NA   20 0.5 1654 0.696 1.061
```

```r
# since SES is correlated with EDUC we will fill the missing SES value with
the median based on EDUC level
# finding median SES of those with EDUC equal 12 or 16
SES.educ12 <- median(dat.complete[dat.complete$EDUC == 12,"SES"])
SES.educ16 <- median(dat.complete[dat.complete$EDUC == 16,"SES"])

# since MMSE is corrolated with CDR, we will fill the missing MMSE value with
the median based on CDR score
# finding median MMSE of those with CDR equal to 1.0
MMSE.CDR.1 <- median(dat.complete[dat.complete$CDR == 1.0,"MMSE"])

# now filling in val
dat.missing[dat.missing$EDUC ==12, "SES"] <- SES.educ12
dat.missing[dat.missing$EDUC ==16, "SES"] <- SES.educ16
dat.missing[dat.missing$Subject.ID == "OAS2_0181","MMSE"] <- MMSE.CDR.1

(dat.missing)
```

```
##     Subject.ID      estimate  Group.x        MRI.ID  Group.y Visit
MR.Delay M.F
## 2    OAS2_0002 -6.111841e-03 Demented OAS2_0002_MR3 Demented     3
1895   M
## 5    OAS2_0007 -1.062526e-02 Demented OAS2_0007_MR4 Demented     4
1281   M
## 53   OAS2_0063 -1.191837e-02 Demented OAS2_0063_MR2 Demented     2
490   F
## 82   OAS2_0099 -5.427509e-03 Demented OAS2_0099_MR2 Demented     2
807   F
## 95   OAS2_0114 -1.152632e-02 Demented OAS2_0114_MR2 Demented     2
570   F
## 131  OAS2_0160 -6.612319e-04 Demented OAS2_0160_MR2 Demented     2
552   M
## 145  OAS2_0181 -5.181405e-05 Demented OAS2_0181_MR3 Demented     3
1107   F
## 146  OAS2_0182 -9.407216e-04 Demented OAS2_0182_MR2 Demented     2
776   M
##     Hand Age EDUC SES MMSE CDR eTIV  nWBV   ASF
## 2      R  80   12   3   22 0.5 1698 0.701 1.034
## 5      R  75   16   2   27 1.0 1372 0.710 1.279
## 53     R  81   12   3   27 0.5 1453 0.721 1.208
## 82     R  83   12   3   23 0.5 1484 0.750 1.183
## 95     R  78   12   3   27 1.0 1309 0.709 1.341
## 131    R  78   12   3   29 1.0 1569 0.704 1.119
## 145    R  77   12   3   20 1.0 1159 0.733 1.515
## 146    R  75   12   3   20 0.5 1654 0.696 1.061
```

```r
# stitching our dataframes back together
dat.recent.est.cleaned <- bind_rows(dat.complete, dat.missing)

# renaming our "estimate" column to "atrophy.rate"
```

```r
dat.recent.est.cleaned <- dat.recent.est.cleaned %>% rename(atrophy.rate =
estimate)

# if there are no rows with nan in our final cleaned data frame this should
return an empty data frame
# let's check
dat.recent.est.cleaned[!complete.cases(dat.recent.est.cleaned),]
```

```
##  [1] Subject.ID   atrophy.rate Group.x      MRI.ID       Group.y
##  [6] Visit        MR.Delay     M.F          Hand         Age
## [11] EDUC         SES          MMSE         CDR          eTIV
## [16] nWBV         ASF
## <0 rows> (or 0-length row.names)
```

```r
head(dat.recent.est.cleaned)
```

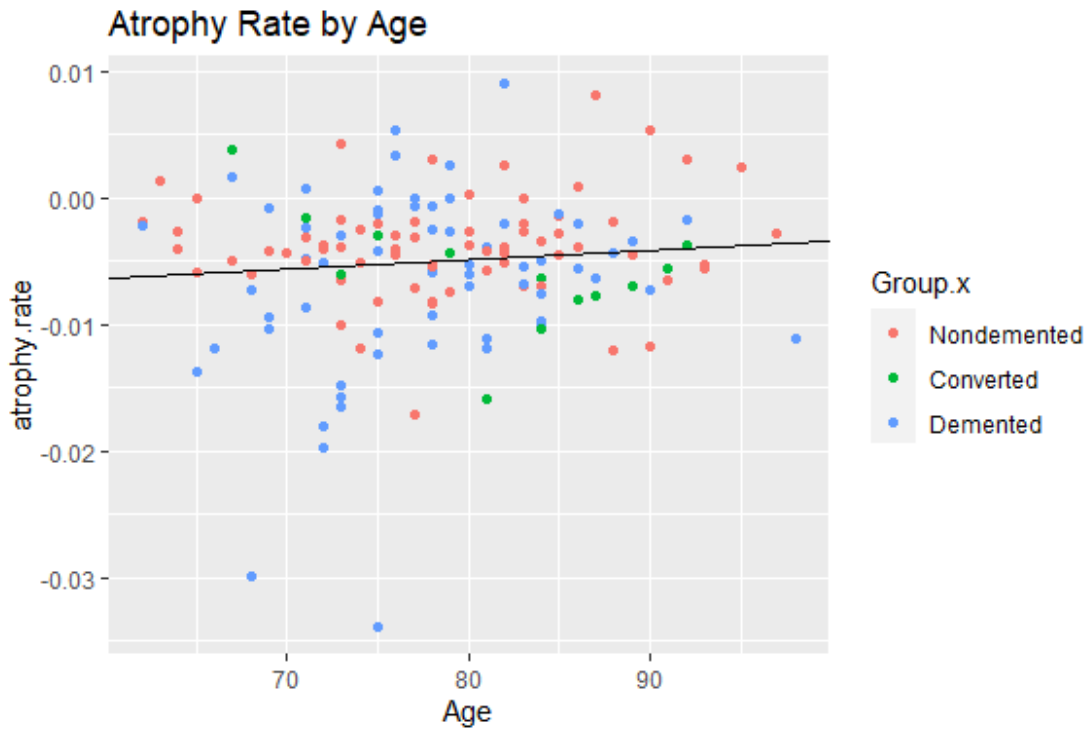
```
##   Subject.ID atrophy.rate     Group.x        MRI.ID     Group.y Visit
MR.Delay
## 1  OAS2_0001 -0.011980306 Nondemented OAS2_0001_MR2 Nondemented     2
457
## 2  OAS2_0004  0.005427509 Nondemented OAS2_0004_MR2 Nondemented     2
538
## 3  OAS2_0005 -0.001462183 Nondemented OAS2_0005_MR3 Nondemented     3
1603
## 4  OAS2_0008  0.002459569 Nondemented OAS2_0008_MR2 Nondemented     2
742
## 5  OAS2_0009 -0.009505208    Demented OAS2_0009_MR2    Demented     2
576
## 6  OAS2_0010 -0.007265808    Demented OAS2_0010_MR2    Demented     2
854
##   M.F Hand Age EDUC SES MMSE CDR eTIV  nWBV   ASF
## 1   M    R  88   14   2   30 0.0 2004 0.681 0.876
## 2   F    R  90   18   3   27 0.0 1200 0.718 1.462
## 3   M    R  85   12   4   30 0.0 1699 0.705 1.033
## 4   F    R  95   14   2   29 0.0 1257 0.703 1.396
## 5   M    R  69   12   2   24 0.5 1480 0.791 1.186
## 6   F    R  68   12   3   29 0.5 1482 0.752 1.184
```

```r
atrophyrate.Age.lm <- lm(atrophy.rate~Age, data = dat.recent.est.cleaned)
summary(atrophyrate.Age.lm)
```

```
##
## Call:
## lm(formula = atrophy.rate ~ Age, data = dat.recent.est.cleaned)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0286487 -0.0023691  0.0007336  0.0032667  0.0137982
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -1.075e-02  4.898e-03  -2.195   0.0297 *
## Age          7.306e-05  6.218e-05   1.175   0.2419
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005821 on 148 degrees of freedom
## Multiple R-squared:  0.009242,   Adjusted R-squared:  0.002548
## F-statistic: 1.381 on 1 and 148 DF,  p-value: 0.2419
```

```
b <- atrophyrate.Age.lm$coefficients[1]
m <- atrophyrate.Age.lm$coefficients[2]

ggplot(dat.recent.est.cleaned, aes(x=Age, y=atrophy.rate, color=Group.x)) +
  geom_point() +
  geom_abline(slope = m, intercept = b) +
  ggtitle("Atrophy Rate by Age")
```




```
cor.test(dat.recent.est.cleaned$atrophy.rate, dat.recent.est.cleaned$Age,
method="pearson")

##
##  Pearson's product-moment correlation
##
## data:  dat.recent.est.cleaned$atrophy.rate and dat.recent.est.cleaned$Age
## t = 1.175, df = 148, p-value = 0.2419
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.06512967  0.25250651
```

```
## sample estimates:
##       cor
## 0.0961353
```

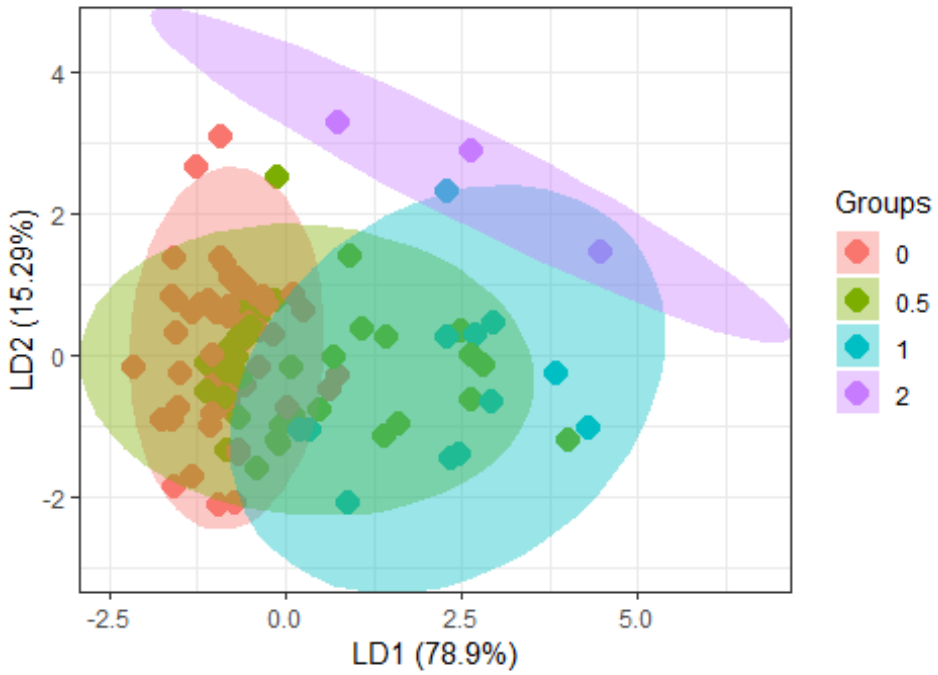
3. Training Classifiers

Here I plan to use LDA and Random Forest, then compare models.

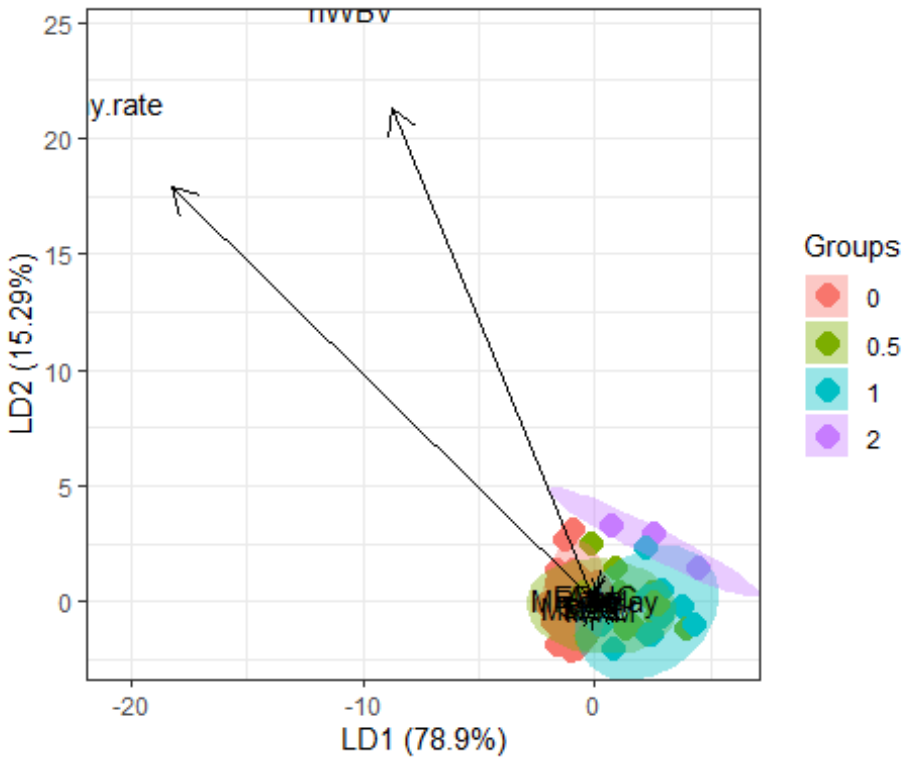
```
set.seed(123456)

# dropping a few more columns that we aren't interested in
to_drop = c("Subject.ID","MRI.ID", "Group.x","Group.y", "Hand", "Visit",
"ASF")
dat.clas.CDR <- dat.recent.est.cleaned[, !(names(dat.recent.est.cleaned) %in%
to_drop)]

# changing CDR to our factor (this is our predicted value)
# we can then reconstruct Group labels from this
dat.clas.CDR["CDR"] <-  as.factor(dat.clas.CDR$CDR)

split_i <- sample(2, nrow(dat.clas.CDR),
                  replace = TRUE,
                  prob = c(0.7,0.3))

training <- dat.clas.CDR[split_i == 1,]
testing <- dat.clas.CDR[split_i == 2,]

linear <- lda(CDR ~ ., training)
(linear)
```

```
## Call:
## lda(CDR ~ ., data = training)
##
## Prior probabilities of groups:
##          0        0.5          1          2
## 0.52380952 0.33333333 0.11428571 0.02857143
##
## Group means:
##     atrophy.rate MR.Delay      M.FM      Age     EDUC      SES     MMSE
## 0   -0.003790510 1202.255 0.2909091 78.01818 15.10909 2.290909 29.09091
## 0.5 -0.004824156 1054.114 0.4571429 78.54286 13.37143 2.771429 26.20000
## 1   -0.009647416  749.250 0.3333333 74.50000 13.16667 3.000000 21.66667
## 2   -0.005939464  643.000 0.3333333 85.00000 17.00000 1.666667 20.33333
##         eTIV      nWBV
## 0   1477.655 0.7364000
## 0.5 1469.857 0.7184000
## 1   1430.917 0.6951667
## 2   1538.000 0.7066667
##
```

```
## Coefficients of linear discriminants:
##                          LD1           LD2           LD3
## atrophy.rate -1.828425e+01 17.8819890257 43.6201761543
## MR.Delay     -3.311450e-04 -0.0003959909  0.0003686478
## M.FM          2.655196e-01 -0.2597611678  1.4536117223
## Age          -2.032231e-02  0.1147924758  0.0926122135
## EDUC          2.673984e-02  0.2339426438 -0.3070855526
## SES          -1.515959e-01 -0.1115010590 -0.3451672202
## MMSE         -3.437260e-01 -0.1696269391 -0.0727073851
## eTIV         -8.399545e-04  0.0018225990 -0.0001371923
## nWBV         -8.706303e+00 21.3335073535 12.3444291339
##
## Proportion of trace:
##    LD1    LD2    LD3
## 0.7890 0.1529 0.0581

ggord(linear, training$CDR, txt=NULL, arrow=NULL)
```




```
ggord(linear, training$CDR)
```




```
# how well did the training go?
p1 <- predict(linear, training)$class
tab <- table(Predicted = p1, Actual = training$CDR)
(tab)
```

```
##           Actual
## Predicted  0 0.5  1  2
##       0   51  20  0  0
##       0.5  4  10  3  0
##       1    0   5  8  0
##       2    0   0  1  3
```

```
# how well did the testing go?
lin.p2 <- predict(linear, testing)$class

confusionMatrix(data=lin.p2, reference = testing$CDR)
```

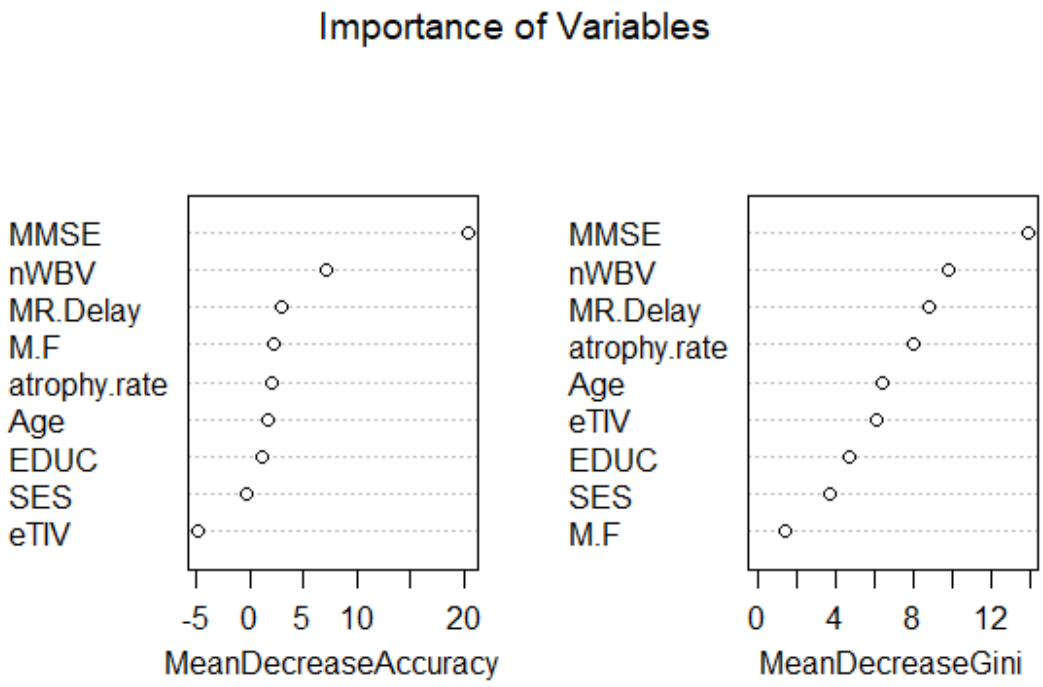
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0 0.5  1  2
##        0   18   8  1  0
##        0.5  0   9  2  0
##        1    0   1  2  0
##        2    0   0  4  0
##
## Overall Statistics
```

```
##
##                Accuracy : 0.6444
##                  95% CI : (0.4878, 0.7813)
##     No Information Rate : 0.4
##     P-Value [Acc > NIR] : 0.0007966
##
##                   Kappa : 0.4521
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 0 Class: 0.5 Class: 1 Class: 2
## Sensitivity            1.0000     0.5000  0.22222       NA
## Specificity            0.6667     0.9259  0.97222  0.91111
## Pos Pred Value         0.6667     0.8182  0.66667       NA
## Neg Pred Value         1.0000     0.7353  0.83333       NA
## Prevalence             0.4000     0.4000  0.20000  0.00000
## Detection Rate         0.4000     0.2000  0.04444  0.00000
## Detection Prevalence   0.6000     0.2444  0.06667  0.08889
## Balanced Accuracy      0.8333     0.7130  0.59722       NA
```

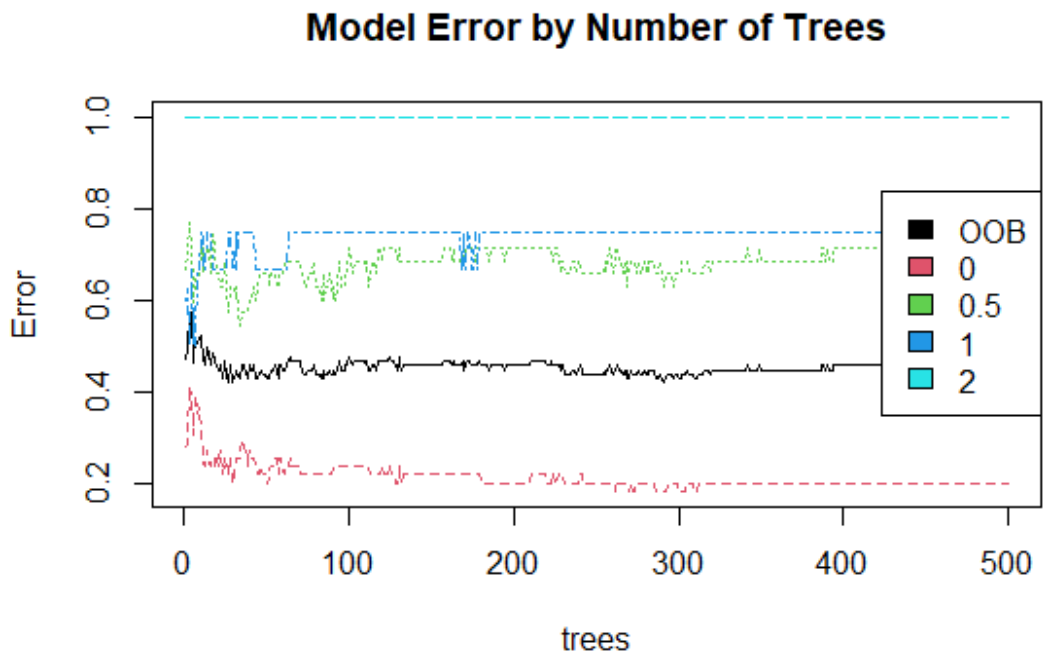
```
set.seed(123456)
split_i <- sample(2, nrow(dat.clas.CDR),
                  replace = TRUE,
                  prob = c(0.7,0.3))

training <- dat.clas.CDR[split_i == 1,]
testing <- dat.clas.CDR[split_i == 2,]

random.forest <- randomForest(formula = CDR ~ .,
                              data = training,
                              importance=TRUE)

# how well did the training go?
print(random.forest)
```

```
##
## Call:
##  randomForest(formula = CDR ~ ., data = training, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 45.71%
## Confusion matrix:
##      0 0.5 1 2 class.error
## 0   44  11 0 0   0.2000000
## 0.5 18  10 7 0   0.7142857
```

```
## 1    0   9 3 0   0.7500000
## 2    0   2 1 0   1.0000000

# how well did the testing go?
rf.p2 <- predict(random.forest, testing)
tab2 <- table(Predicted = rf.p2, Actual = testing$CDR)

varImpPlot(random.forest, main = "Importance of Variables")
```




```
plot(random.forest, main = "Model Error by Number of Trees")
legend(x = "right",
       legend = colnames(random.forest$err.rate),
       fill = 1:ncol(random.forest$err.rate))
```




```
confusionMatrix(data=rf.p2, reference = testing$CDR)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0 0.5  1  2
##        0   14   5  2  0
##        0.5  4  13  5  0
##        1    0   0  2  0
##        2    0   0  0  0
##
## Overall Statistics
##
##                Accuracy : 0.6444
##                  95% CI : (0.4878, 0.7813)
##     No Information Rate : 0.4
##     P-Value [Acc > NIR] : 0.0007966
##
##                   Kappa : 0.4161
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 0 Class: 0.5 Class: 1 Class: 2
## Sensitivity            0.7778     0.7222  0.22222       NA
## Specificity            0.7407     0.6667  1.00000        1
## Pos Pred Value         0.6667     0.5909  1.00000       NA
```

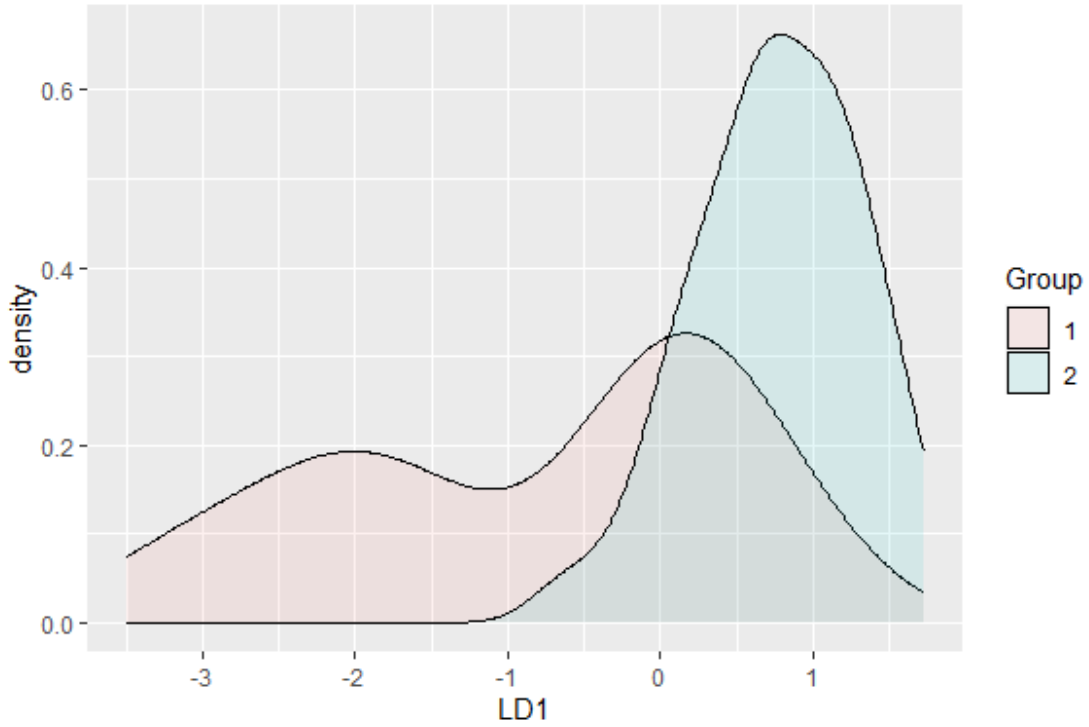
```
## Neg Pred Value         0.8333     0.7826  0.83721       NA
## Prevalence             0.4000     0.4000  0.20000        0
## Detection Rate         0.3111     0.2889  0.04444        0
## Detection Prevalence   0.4667     0.4889  0.04444        0
## Balanced Accuracy      0.7593     0.6944  0.61111       NA
```

Does the model improve if we just try to predict whether participants were demented at all? Treat all participants who maintained a CDR of 0 as Nondemented Treat all others as Demented (CDR of at least 0.5 by the end of the study)

```
set.seed(123456)

# dropping a few more columns that we aren't interested in
to_drop = c("Subject.ID","MRI.ID",  "Group.y","Group.x", "Hand", "Visit",
"ASF")

dat.clas.grp <- dat.recent.est.cleaned[, !(names(dat.recent.est.cleaned) %in%
to_drop)]

# Classifying based on Group Label is the same as treating all
# participants who maintained a CDR of 0 as nondemented
# and all others as demented
# this means that we can achieve the same model by treating all non zero CDR
scores as a single category.

dat.clas.grp[dat.clas.grp$CDR != 0, "CDR"] <- 1
dat.clas.grp[dat.clas.grp$CDR == 0, "CDR"] <-2
dat.clas.grp$CDR <- as.factor(dat.clas.grp$CDR)

split_i <- sample(2, nrow(dat.clas.grp),
                  replace = TRUE,
                  prob = c(0.70,0.30))

training <- dat.clas.grp[split_i == 1,]
testing <- dat.clas.grp[split_i == 2,]

linear <- lda(CDR ~ ., training)
(linear)
```

```
## Call:
## lda(CDR ~ ., data = training)
##
## Prior probabilities of groups:
##         1         2
## 0.4761905 0.5238095
##
## Group means:
##   atrophy.rate MR.Delay      M.FM      Age     EDUC      SES     MMSE
eTIV
## 1 -0.006048657  956.280 0.4200000 77.96000 13.54000 2.760000 24.76000
```

```
1464.600
## 2 -0.003790510 1202.255 0.2909091 78.01818 15.10909 2.290909 29.09091
1477.655
##      nWBV
## 1 0.71212
## 2 0.73640
##
## Coefficients of linear discriminants:
##                          LD1
## atrophy.rate  4.378288e+00
## MR.Delay      8.532363e-05
## M.FM         -7.283073e-01
## Age           7.711326e-03
## EDUC          1.173940e-01
## SES           2.151687e-01
## MMSE          2.765775e-01
## eTIV          1.061292e-03
## nWBV          7.050769e+00

lin.p1 <- predict(linear, training)$class

lin.p2 <- predict(linear, testing)$class


LD1_proj <- predict(linear, training)$x
Group <- training$CDR
df.LD1 <- data.frame(LD1_proj, Group = as.factor(Group))
ggplot(data = df.LD1) +
  geom_density(aes(LD1, fill = Group), alpha = 0.1)
```

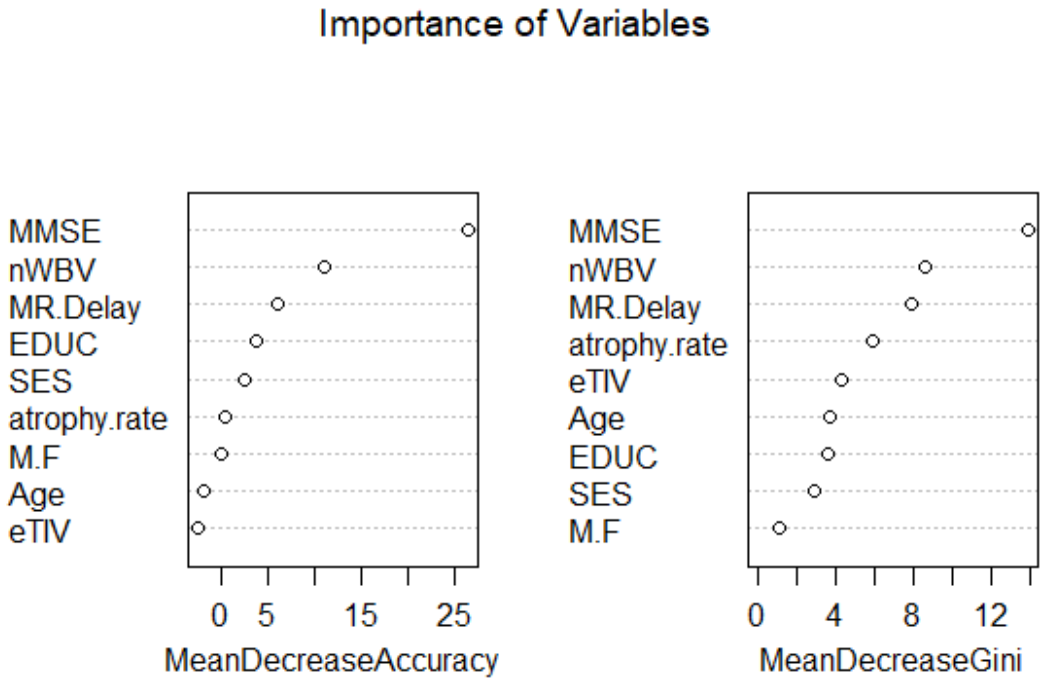



```
confusionMatrix(data=lin.p1, reference = training$CDR)
```

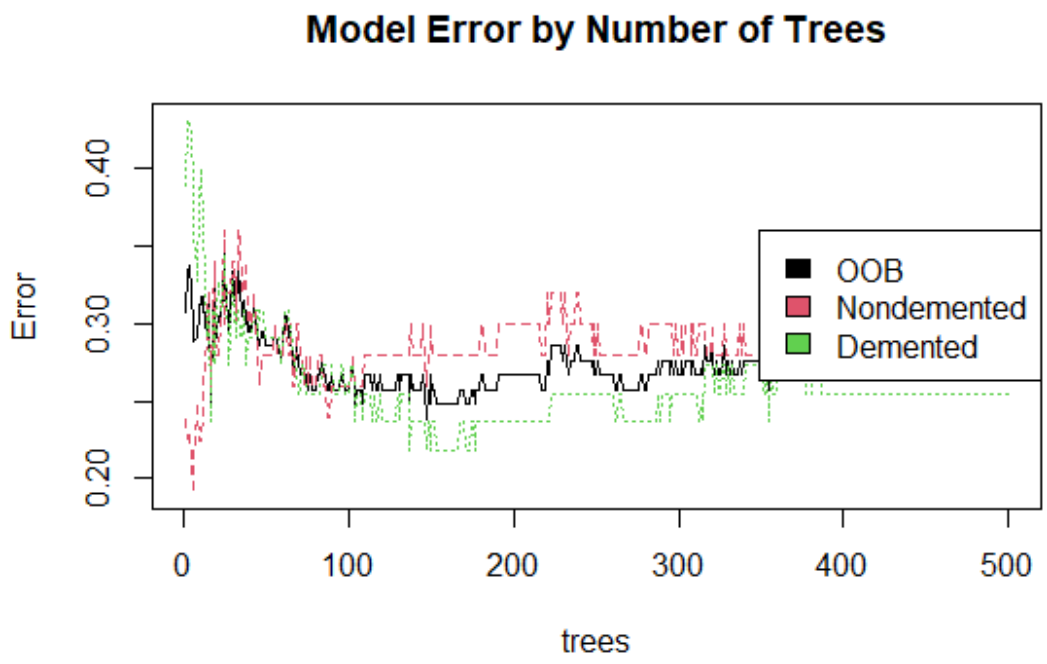
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 30  3
##          2 20 52
##
##                Accuracy : 0.781
##                  95% CI : (0.6897, 0.8558)
##     No Information Rate : 0.5238
##     P-Value [Acc > NIR] : 4.563e-08
##
##                   Kappa : 0.554
##
##  Mcnemar's Test P-Value : 0.0008492
##
##             Sensitivity : 0.6000
##             Specificity : 0.9455
##          Pos Pred Value : 0.9091
##          Neg Pred Value : 0.7222
##              Prevalence : 0.4762
##          Detection Rate : 0.2857
##    Detection Prevalence : 0.3143
##       Balanced Accuracy : 0.7727
##
```

```
##        'Positive' Class : 1
##

confusionMatrix(data=lin.p2, reference = testing$CDR)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 20  0
##          2  7 18
##
##                Accuracy : 0.8444
##                  95% CI : (0.7054, 0.9351)
##     No Information Rate : 0.6
##     P-Value [Acc > NIR] : 0.0003707
##
##                   Kappa : 0.6957
##
##  Mcnemar's Test P-Value : 0.0233422
##
##             Sensitivity : 0.7407
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.7200
##              Prevalence : 0.6000
##          Detection Rate : 0.4444
##    Detection Prevalence : 0.4444
##       Balanced Accuracy : 0.8704
##
##        'Positive' Class : 1
##

set.seed(123456)
random.forest <- randomForest(formula = CDR ~ ., data = training,
importance=TRUE)
print(random.forest)

##
## Call:
##  randomForest(formula = CDR ~ ., data = training, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 26.67%
## Confusion matrix:
##    1  2 class.error
## 1 36 14   0.2800000
## 2 14 41   0.2545455
```

```
rf.p2 <- predict(random.forest, testing, type = 'response')

varImpPlot(random.forest, main = "Importance of Variables")
```




```
plot(random.forest, main = "Model Error by Number of Trees")
legend(x = "right",
       legend = c("OOB","Nondemented", "Demented"),
       fill = 1:ncol(random.forest$err.rate))
```



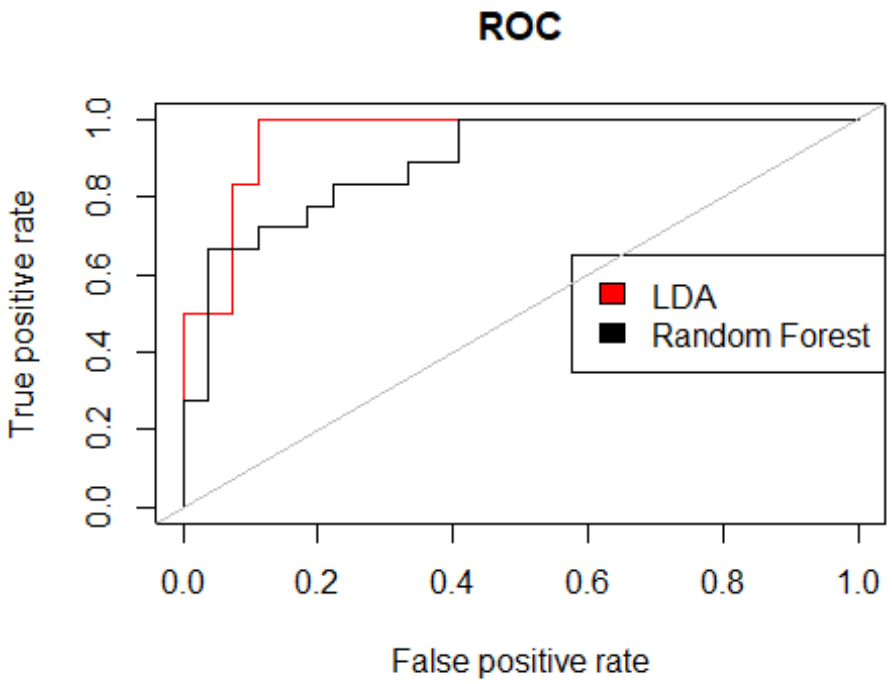

```
confusionMatrix(data=rf.p2, reference = testing$CDR)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 22  5
##          2  5 13
##
##                Accuracy : 0.7778
##                  95% CI : (0.6291, 0.888)
##     No Information Rate : 0.6
##     P-Value [Acc > NIR] : 0.009392
##
##                   Kappa : 0.537
##
##  Mcnemar's Test P-Value : 1.000000
##
##             Sensitivity : 0.8148
##             Specificity : 0.7222
##          Pos Pred Value : 0.8148
##          Neg Pred Value : 0.7222
##              Prevalence : 0.6000
##          Detection Rate : 0.4889
##    Detection Prevalence : 0.6000
##       Balanced Accuracy : 0.7685
##
```

```
##        'Positive' Class : 1
##
```

Model Evaluation

We prefer the model with the larger area under the receiver operating characteristics curve. For AD diagnosing, higher sensitivity is preferred over accuracy or specificity. Based on the confusion matrices above as well as the AUC plotted below, we prefer the LDA model.

```
# LDA AUROC
lda.pred <- predict(linear, testing)
lda.prediction <- prediction(lda.pred$posterior[,2], testing$CDR)
lda.auc <- performance(lda.prediction, measure = "auc")@y.values[[1]]
print(lda.auc)

## [1] 0.9567901

lda.perf.plot <- performance(lda.prediction,"tpr","fpr")
plot(lda.perf.plot, col="red", main="ROC")

# RF AUROC
rf.pred <- predict(random.forest, type="prob", testing)[,2]
rf.prediction <- prediction(rf.pred, testing$CDR)
rf.auc <- performance(rf.prediction, measure = "auc")@y.values[[1]]
print(rf.auc)

## [1] 0.8930041

rf.perf.plot <- performance(rf.prediction, "tpr","fpr")
plot(rf.perf.plot, add=TRUE)

legend(x = "right",
       legend = c("LDA","Random Forest"), fill=c("red", "black"))
abline(coef = c(0, 1), col="grey")
```


ROC
True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
LDA
Random Forest