Statistical Modeling of DTI Data

Annie Bathen, Fangfei Lan, Nicole LaPointe Jameson Jingyi Wang, Kyle Rehr, Sajala Shukla, and Elizabeth Wang

May 23, 2016

FOR EACH MODEL:

Write out model assumptions output performance metrics ----

# Goal Of the Project

The group was given **Diffusion Tensor Imaging** Data.

The **goal** of this project is to determine whether diffusion properties (through the provided metrics on cca, along with sex) are related to cognitive ability (measured through pasat score).

## Project Overview

1. Data Cleaning

* First visit only
* Discounting number of scans
* Aggregating CCA to a useable metrics (mean, median, minimum of all locations)
* Removal of RCST

1. Pasat Score Resolution

* Inputting missing data
* Significance of MS patients also getting a perfect score
* Exploratory analysis and tests of significance

1. Exploratory Analysis of Covariates

* Establish assumptions, shape, spread, skew, outliers and trends
* Initial statistical tests

1. Model Fitting

* Feature Selection
* Graphical Analysis
* Best Model and why

1. Final Model

* Performance in predicting pasat from metrics
* Compare with current results

## Establishing The Workbench

The packages in R the group used are:

library(refund)  
library(ggplot2)  
library(caret)

## Loading required package: lattice

library(lattice)  
library(MASS)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':  
##   
## select

## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

library(tidyr)

## Warning: package 'tidyr' was built under R version 3.2.5

library(reshape2)  
library(colorspace)  
library(glmnet)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following object is masked from 'package:tidyr':  
##   
## expand

## Loading required package: foreach

## Loaded glmnet 2.0-3

# Part One: Data Cleaning

data(DTI) #Read it in  
DTI\_CLEAN = subset(DTI,DTI$visit==1) #First Visit only  
DTI\_a = DTI\_CLEAN[,-3] # Removing Visit Time  
DTI\_b = DTI\_a[,-8] #Removing RCST  
  
#Preprocessed and Cleaned Data Set  
DTI.1 = DTI\_b

## CCA Metrics

#Mean  
cca.iMean = array(NA)  
for(i in 1:142){  
 cca.ivect=c(DTI.1$cca[i,])  
 cca.iMean[i]=mean(cca.ivect, na.rm = TRUE)  
}  
  
ccaMean = cca.iMean  
  
#Median  
cca.iMedian = array(NA)  
for(i in 1:142){  
 cca.ivect=c(DTI.1$cca[i,])  
 cca.iMedian[i]=median(cca.ivect, na.rm = TRUE)  
}  
  
ccaMed = cca.iMedian  
  
#Minimum  
cca.iMin = array(NA)  
for(i in 1:142){  
 cca.ivect=c(DTI.1$cca[i,])  
 cca.iMin[i]=min(cca.ivect, na.rm = TRUE)  
}  
  
ccaMin = cca.iMin  
  
#Final Data Set: Including CCA Metrics  
  
DTI.1 = cbind(DTI.1, ccaMean, ccaMed, ccaMin)  
  
#Final Data Set for Exploratory Analysis Missing Value Inputation  
#DTI.1  
  
#Data Set for model fitting (MS Patients only)  
DTI.2 = subset(DTI.1, as.numeric(DTI.1$case) == 1)  
#DTI.2

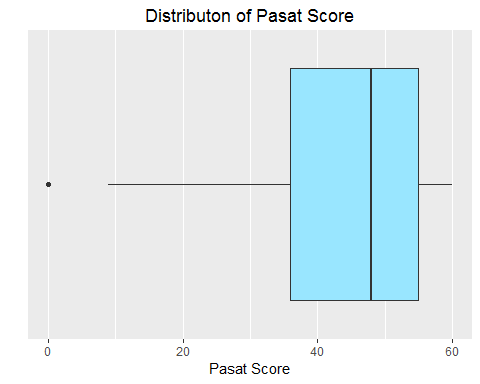
## Partitioning The Data

set.seed(2145) #To Make reproducible  
#Take a SRS Based on Outcome  
  
trainIndex = createDataPartition(DTI.2$pasat, p=.7, list= FALSE, times=1)  
  
DTITrain = DTI.2[trainIndex,]  
DTITest = DTI.2[-trainIndex,]  
  
#Training data: DTITrain

# Part Two: Pasat Score Resolution

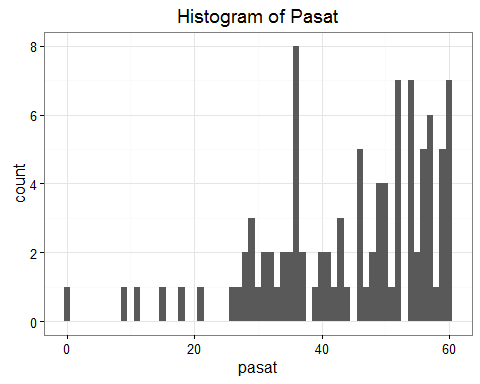
ggplot(DTI.1, aes(x = factor(0), y = pasat)) + geom\_boxplot(fill="#99e6ff") + xlab("") + coord\_flip() + scale\_x\_discrete(breaks = NULL) + labs(title="Distributon of Pasat Score", y="Pasat Score")

## Warning: Removed 42 rows containing non-finite values (stat\_boxplot).

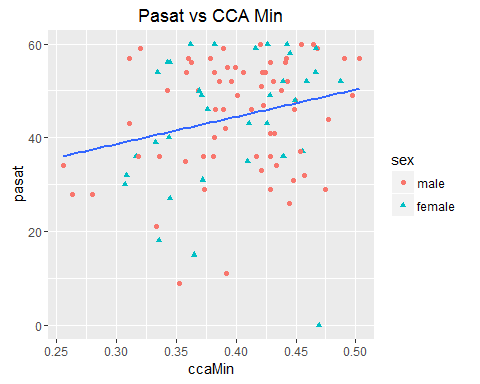


# pasat histogram  
ggplot(DTI.1, aes(pasat)) + geom\_histogram( binwidth=1 ) + theme\_bw() + ggtitle("Histogram of Pasat")

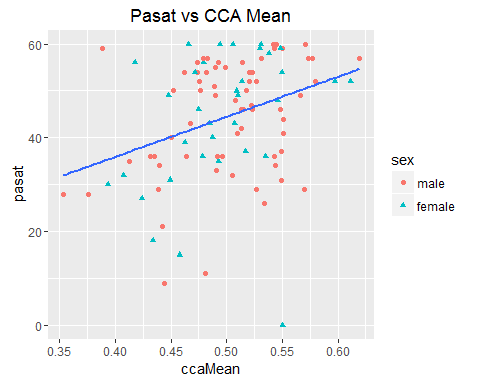
## Warning: Removed 42 rows containing non-finite values (stat\_bin).



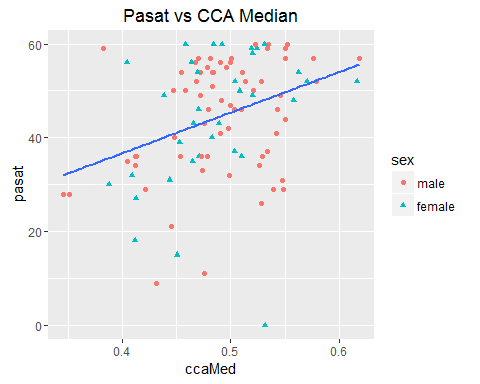
# pasat vs ccaMin  
ggplot(DTI.1[43:142,], aes(x = ccaMin, y = pasat)) +   
 geom\_point(aes(shape = sex, colour = sex)) +  
 geom\_smooth(method = "lm", se = FALSE) + ggtitle("Pasat vs CCA Min")



# pasat vs ccamean  
ggplot(DTI.1[43:142,], aes(x = ccaMean, y = pasat)) +   
 geom\_point(aes(shape = sex, colour = sex)) +  
 geom\_smooth(method = "lm", se = FALSE)+ ggtitle("Pasat vs CCA Mean")



# pasat vs ccamedian  
ggplot(DTI.1[43:142,], aes(x = ccaMed, y = pasat)) +   
 geom\_point(aes(shape = sex, colour = sex)) +  
 geom\_smooth(method = "lm", se = FALSE) + ggtitle("Pasat vs CCA Median")



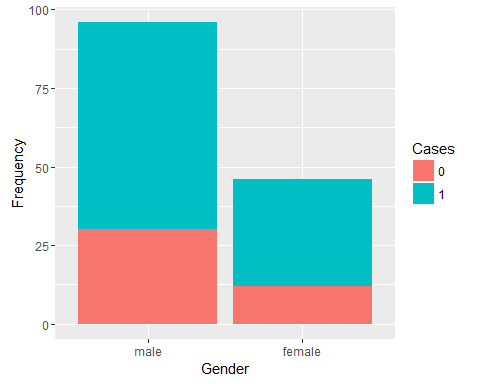
# Part Three: Exploratory Analysis

## Gender

summary(DTI.1$sex)

## male female   
## 96 46

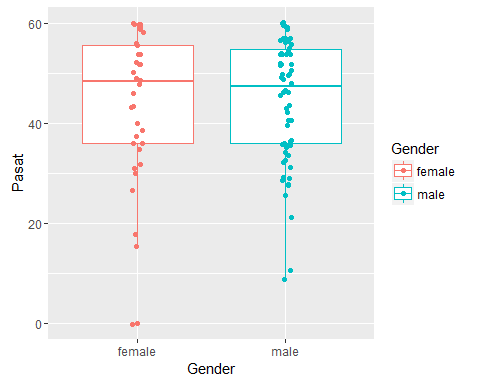
femaleData = subset(DTI.1, DTI.1$sex=="female")  
maleData = subset(DTI.1, DTI.1$sex=="male")  
  
# Frequency of male and female grouped by cases  
Cases = factor(DTI.1$case)  
caseByGender = ggplot(DTI.1, aes(x=DTI.1$sex, fill=Cases))  
caseByGender + geom\_bar() + xlab("Gender") + ylab("Frequency")



# Boxplot of Pasat score grouped by gender  
DTI.1$sex = factor(DTI.1$sex, c("female","male"))  
Gender = DTI.1$sex  
pasatVal = ggplot(DTI.1, aes(x=sex, y=pasat)) + geom\_boxplot(aes(colour=Gender)) + xlab("Gender") + ylab("Pasat")  
pasatVal + geom\_jitter(width = 0.1, aes(color=sex))

## Warning: Removed 42 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 42 rows containing missing values (geom\_point).



12/46 female patients do not have MS (meaning 34/46 patients do have MS)

30/96 male patients do not have MS (meaning 66/96 patients do have MS)

proportion of patients with MS differs between men and women).

Also found a p-value using a chisquard test: p-value = 0.664

### Testing

# Proportion Test  
table(DTI.1$case, DTI.1$sex)

##   
## female male  
## 0 12 30  
## 1 34 66

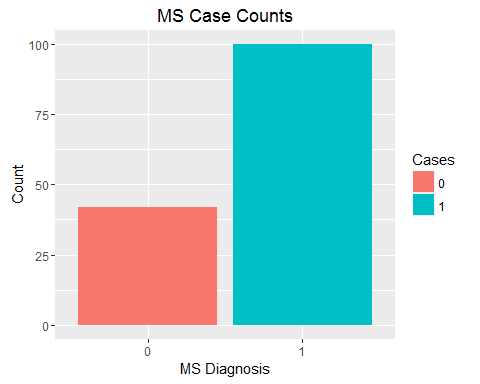
prop.test(x = c(34, 66), n = c(46, 96), correct = FALSE)

##   
## 2-sample test for equality of proportions without continuity  
## correction  
##   
## data: c(34, 66) out of c(46, 96)  
## X-squared = 0.398, df = 1, p-value = 0.5281  
## alternative hypothesis: two.sided  
## 95 percent confidence interval:  
## -0.1055291 0.2087900  
## sample estimates:  
## prop 1 prop 2   
## 0.7391304 0.6875000

* For a test of independence of proportions, we produced a **P-Value of .5281**, suggesting that we cannot reject the null hypothesis of MS Diagnosis being independent of sex.
* To reaffirm the test, using two independent proportions under conditions of np and nq being >= 10, a z-test produced a **P-value of .7357**, suggesting that there is not enough evidence to reject the Null Hypothesis that the proportion of patients with MS differs between men and women.

## Case

Cases = factor(DTI.1$case)  
  
#Distribution Plot  
qplot(factor(case), data=DTI.1, geom="bar", fill=Cases) + labs(title = "MS Case Counts", x="MS Diagnosis", y="Count")



#Pearson Chisq test for count differences  
chisq.test(table(DTI.1$case))

##   
## Chi-squared test for given probabilities  
##   
## data: table(DTI.1$case)  
## X-squared = 23.69, df = 1, p-value = 1.132e-06

#Produces a X-sq value of 23.69, with a p-value <.05, this we reject the null hypothesis of an equal distribution of MA Case counts

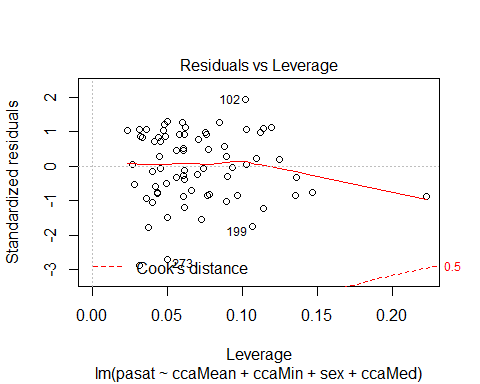
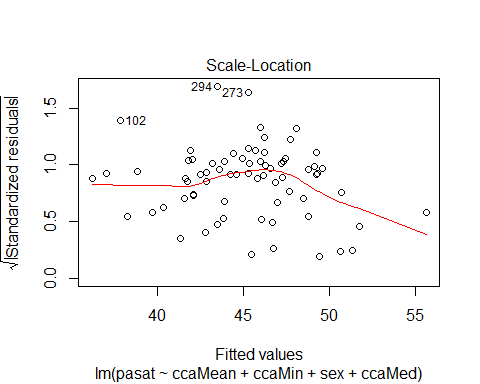
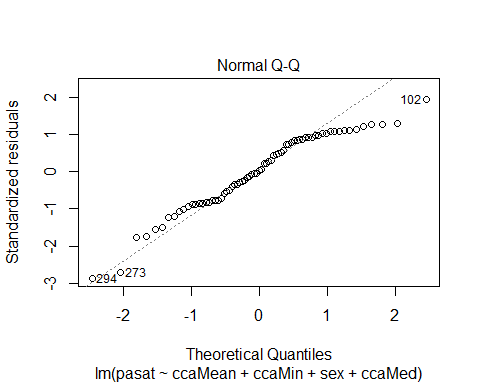
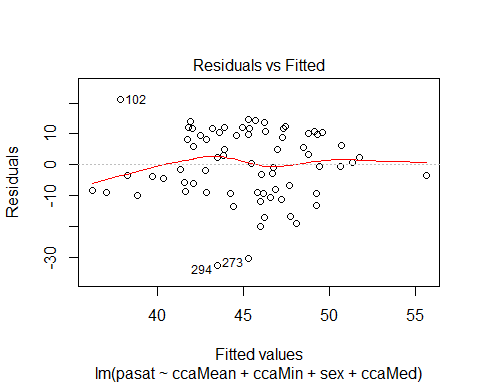
# Part Four: Model Fitting

## Model 1: Basic Linear Model (Multiple Linear Regression)

#Baseline  
lm.fit <- lm(pasat ~ ccaMean+ccaMin+ sex+ ccaMed, data=DTITrain)  
  
summary(lm.fit)

##   
## Call:  
## lm(formula = pasat ~ ccaMean + ccaMin + sex + ccaMed, data = DTITrain)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -32.493 -8.739 0.500 9.956 21.097   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 16.610 14.247 1.166 0.248  
## ccaMean -23.288 147.512 -0.158 0.875  
## ccaMin -43.125 49.145 -0.877 0.383  
## sexfemale 3.112 2.892 1.076 0.286  
## ccaMed 115.450 136.246 0.847 0.400  
##   
## Residual standard error: 11.52 on 66 degrees of freedom  
## Multiple R-squared: 0.0961, Adjusted R-squared: 0.04132   
## F-statistic: 1.754 on 4 and 66 DF, p-value: 0.1487

plot(lm.fit)



* None of our predictors are significant, suggesting either a bad fit, departures from assumptions, or multicollinearity.
* As we can see from the QQ Norm plot, there are huge departures from an assumption of normality. Residuals also are clustered and not "clouded" around zero. A Linear fit as is would not be suggested for this data. Strong outliers are also present, and R^2 value is also extremely low.

## Model 2: Functional Generalized Additive Model

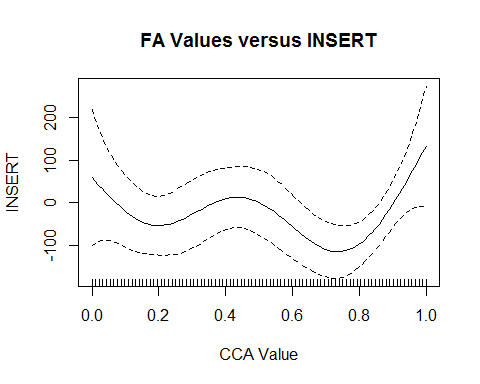
## FGAM for Q1, with all default options  
  
Y = as.vector(DTI$case)  
X = as.numeric(DTI$sex) - 1 # Female = 1, Male = 0  
Z = DTI$cca  
fit\_1 <- fgam(Y ~ X + lf(Z), family = binomial)

* The model fitted is in fact logit(case = 1) = gamma\_0 + gamma\_1 \* X + Int(Z(S)\*beta(s))

The invloved tests are: 1. Z score test for gamma\_0 (intercept) 2. Z score test for gamma\_1 (X) 3. Generalized Chi-sq test for smooth terms b\_k, k from 3 to K\_b (b\_k assumed to ~ N(0, sigma\_k^2), which then can be deemed as random effects)

## Test of Approximate Significance of Smooth Terms of Best Model:

fit\_3 <- fgam(Y ~ X + lf(Z, k=30, presmooth="fpca.sc",presmooth.opts=list(nbasis=8, pve=.975)), family = binomial)  
  
plot(fit\_3, main="FA Values versus INSERT", ylab="INSERT", xlab="CCA Value")

 - Therefore, if the test of Approximate significance of smooth terms (in summary table) is significant.

* We cannot reject that not all the b\_K are zero, which means beta(s) is nontrivial.

## Model 3: LASSO on Cognition Ability

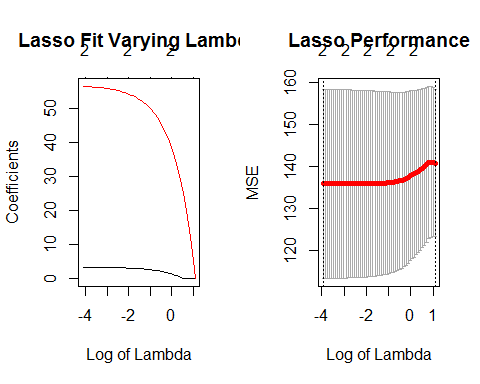
* Least Absolute Shrinkage and Selection Operatorperforms both variable selection and regularization to enhance the prediction accuracy.
* "Best of both worlds" of other penalized regression, becuase it minimizes usual sum of squared errors, with a bound on the sum of the aboluste value of coefficiants
* WHY USE LASSO

gend = as.numeric(DTITrain$sex)  
LTrain = cbind(gend,DTITrain$ccaMed)  
LTrain = as.matrix(LTrain)  
  
#Fit  
fit.lasso = glmnet(x=LTrain, y=DTITrain$pasat, family="gaussian", alpha=1)

A lasso model using just gender and ccaMed has a 23.22% explanation of the null deviance with lambda set to .08136.

### LASSO Cross Validation: 10-Fold Cross Validation

for (i in 0:10) {  
 assign(paste("fit", i, sep=""), cv.glmnet(LTrain, DTITrain$pasat, type.measure="mse", alpha=i/10,family="gaussian"))  
}  
  
par(mfrow=c(1,2))  
plot(fit.lasso, xvar="lambda", xlab= "Log of Lambda", ylab="Coefficients", main="Lasso Fit Varying Lambda")  
  
plot(fit10, main="Lasso Performance", xlab="Log of Lambda", ylab="MSE")



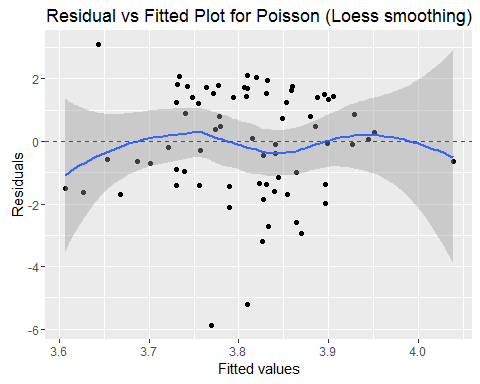
## Model 4: Poisson on Cognition

poissonModel = glm(DTITrain$pasat ~ ccaMean + sex, family="poisson", data=DTITrain)  
#Res Dev =219.50  
poissonModel2 = glm(DTITrain$pasat ~ ccaMed + sex, family="poisson", data=DTITrain)  
#Res Dev= 217.96  
poissonModel3 = glm(DTITrain$pasat ~ ccaMed+ ccaMean+ ccaMin + sex, family="poisson", data=DTITrain)  
#Res Dev= 215.23

All These models are fitting the intercept with High Null Deviance. Not a good fit!

Where where assumptions broken?

ggplot(poissonModel3, aes(.fitted, .resid))+geom\_point() +stat\_smooth(method="loess") + geom\_hline(yintercept=0, col="red", linetype="dashed") + xlab("Fitted values")+ylab("Residuals") + ggtitle("Residual vs Fitted Plot for Poisson (Loess smoothing)")



## Fitting A Quasi-Poisson

* Poisson regression accounting for overdispersion

quasipoissonModel = glm(DTITrain$pasat ~ ccaMean + ccaMed +ccaMin + sex, data=DTITrain, family=quasipoisson)  
  
summary(quasipoissonModel)

##   
## Call:  
## glm(formula = DTITrain$pasat ~ ccaMean + ccaMed + ccaMin + sex,   
## family = quasipoisson, data = DTITrain)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -5.875 -1.375 0.049 1.420 3.108   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.16382 0.32583 9.710 2.43e-14 \*\*\*  
## ccaMean -0.38968 3.29841 -0.118 0.906   
## ccaMed 2.44371 3.01938 0.809 0.421   
## ccaMin -0.94705 1.09133 -0.868 0.389   
## sexfemale 0.06916 0.06381 1.084 0.282   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for quasipoisson family taken to be 2.983242)  
##   
## Null deviance: 235.89 on 70 degrees of freedom  
## Residual deviance: 215.23 on 66 degrees of freedom  
## AIC: NA  
##   
## Number of Fisher Scoring iterations: 4

#Res Dev is 215.23 of 66 df

Hypothesis Testing On "Best" Model: - Running an F-Test on the full versus reduced model

ccaPoissonModel = glm(DTITrain$pasat ~ ccaMean, data=DTITrain, family=quasipoisson)  
  
deviancecca = 223.91 #Residual deviance of ccaPoissonModel  
devianceFull = 215.23 #Residual deviance of the full model  
paraDropped = 3 #parameter dropped  
overDispersion = 2.983242 #Over dispersion parameter from the full model  
  
# Test statistics  
F.stat = ((deviancecca-devianceFull)/paraDropped)/overDispersion

The F-statistic is **.96986**, which is close to 1.

The p-value is close to 1, so we do not reject null hypothesis. Thus, the parameters dropped are not significant to the patients' pasat scores.

# Model 5: Function-on-scalar Regression

* Insert BG
* We will only be comparing with CCA results on Pasat score

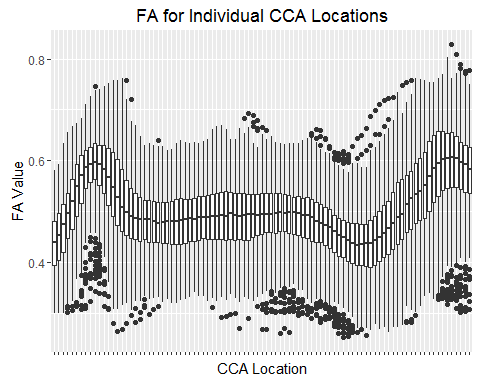
### Functional Plots of FA Values Across Location

In seeing how much CCA FA values range within each location:

ggplot(data=melt(as.data.frame(DTI$cca)), aes(variable, value)) + geom\_boxplot(aes())+  
labs(title="FA for Individual CCA Locations",x="CCA Location",y="FA Value")+  
theme(axis.text.x = element\_blank())

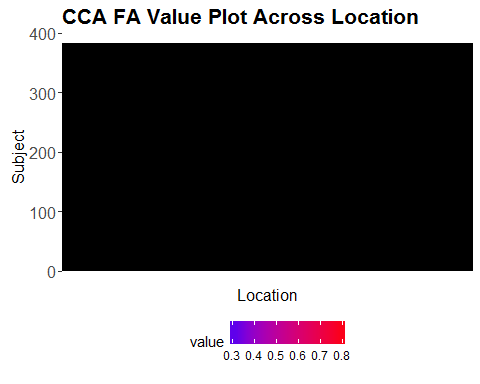
## No id variables; using all as measure variables

## Warning: Removed 36 rows containing non-finite values (stat\_boxplot).



Which can be better presented by a functional graph:

cca.obj = DTI$cca  
  
rownames(cca.obj) = seq(nrow(cca.obj))  
colnames(cca.obj) = seq(ncol(cca.obj))  
names(dimnames(cca.obj)) = c("subj", "grid")  
  
cca.df = melt(cca.obj)  
  
###  
  
ggplot(cca.df,aes(x=grid,y=subj, fill= value)) + scale\_fill\_continuous(low = "#0000ff", high = "#ff0000", na.value="grey50", guide= "colourbar") + geom\_tile(colour='black') +   
 ggtitle("CCA FA Value Plot Across Location")+ xlab("Location") + ylab("Subject")+  
 theme(axis.text=element\_text(size=12),  
 axis.title=element\_text(size=12),  
 plot.title = element\_text(hjust = 0, size=16, face="bold"),  
 panel.background=element\_blank(),  
 panel.grid.minor=element\_blank(),  
 legend.position="bottom") + scale\_x\_discrete(breaks=1:nrow(cca.df))



### Comparison with actual models: Function on Scalar Regression

We will be comparing this to a Bayesian function-on-scalar regression model which does INSERT

\*Explain differences in error estimation

"GLS" doesn't do anything Bayesian - just fits an unpenalized GLS estimator for the specified model

library(refund)  
  
#Ordinary Least Sqaures  
OLS = bayes\_fosr(cca ~ pasat, data = DTI.1, Kt = 10, est.method = "OLS")

## Using OLS to estimate model parameters

#Generalized Least Squares  
GLS = bayes\_fosr(cca ~ pasat, data = DTI.1, Kt = 10, est.method = "GLS")

## Using OLS to estimate residual covariance   
## GLS

## Comparison  
models = c("OLS", "GLS")  
intercepts = sapply(models, function(u) get(u)$beta.hat[1,])  
slopes = sapply(models, function(u) get(u)$beta.hat[2,])  
plot.dat = melt(intercepts); colnames(plot.dat) = c("grid", "method", "value")  
ggplot(plot.dat, aes(x = grid, y = value, group = method, color = method)) +  
geom\_path() + theme\_bw() + labs(title="Comparative Bayes FOSR Plots For Error Estimation", x="Location", y="Estimate Value")

