CODE ▾

Using Network Measures of Functional Brain Activity to Predict Personality

Jared P. Zimmerman, Jennifer Stiso, Katerina Placek

11/14/2017

Executive Overview

Here, we take advantage of a large, publicly-available dataset funded by the National Institutes of Health (NIH) to investigate a novel research question: whether functional brain connectivity can predict personality characteristics. Prior studies of human personality using functional magnetic resonance imaging (fMRI) data typically collapse data from many subjects to draw conclusions about the anatomical correlates of personality characteristics (e.g. Canil et al., 2001), but more recent study suggests that brain functional organization varies widely between individuals and can serve as a 'fingerprint' for identification of the individual within a larger group. Indeed, functional connectivity profiles of fMRI data defined per individual using network measures have been demonstrated to reliably discriminate individuals and relate to fluid intelligence and cognitive behavior (Finn et al., 2015). This suggests that intrinsic variation in functional connectivity profiles may reflect individual differences. Further recent work suggests that functional brain connectivity profiles may extend beyond description or discrimination of the individual to more advanced usage in prediction. Data-driven predictive modeling has been posed as a novel, generalizable method to generate predictions of behavioral measures from functional brain connectivity data in novel individuals. Cross-validation indicates robustness of observed brain-behavior relationship using this method (Shen et al., 2017). With this prior research in mind, we use similar data-driven methods in this study to develop predictive models of personality characteristics from network measures of functional brain connectivity data. We employ three models: 1) Random Forest 2) Boosting We compare these models in terms of predictive accuracy for self-reported personality characteristics using cross-validation.

Data Description

HIDE

```
#loading in data
dataFiles <- c('allData_ic149_fullCor_n810_noG.csv', 'allData_ic149_pCor_n810_noG.csv'
)
corNames <- c('full', 'partial')
data <- lapply(dataFiles, read.csv, stringsAsFactors = TRUE)
names(data) <- corNames
```

```
## $full
## [1]   810 11821
## 
## $partial
## [1]   810 11821
```

Our data has 810 observations, each a human subject from the HCP with neuroimaging data and behavioral data including the NEO five factor personality inventory. The dataset has 11821 variables. These include demographics such as age and gender, the NEO Five Factor personality scores, structural brain brain data including cortical thickness and regional brain volumes, and brain functional connectivity data including the pairwise connections between 149 brain regions in a functional brain network. These brain connectivity data account for most of the 11821 variables with `(149^2)/2 - 149/2` = 1.102610^{4} unique connetions in a 149 node brain network. Further information about these data follows.

Human Connectome Project

The Human Connectome Project (HCP) is a multi-center consortium funded by the NIH with the goal of mapping the macroscale structural and functional connectivity of the human brain (Van Essen et al. 2013). In total, the project collected neuroimaging data including structural magnetic resonance imaging (MRI), resting-state functional MRI, task-based functional MRI, and diffusion spectrum imaging, as well as subject measures including behavioral and cognitive assays, demographic data, and health data on over 1200 healthy young adults. These data have all been publically released in raw, processed, and highly-processed formats as resources for the research community. For this project, we utilize some of the highly-processed data from the Parcellations, Timeseries, and Netmats (PTN) release. Specifically, we use the netmats, or network matrices, data. For the PTN release, the HCP consortium used group independent component analysis (ICA) on processed rs-fMRI data to create a data-driven parcellation of the brain by identifying independent sources of variation within and across subjects. Following group ICA, a dual regression approach is used to fit the group-average parcellation template to each individual subject (Smith et al. 2013, Smith et al. 2014). This dual-regression step results in subject-specific maps and a corresponding timeseries for each ICA component. After dual-regression ICA, functional network connectivity matrices were constructed for each subject as $N * N$ matrices where N is the number of ICA components and each matrix element E_{ij} represents the functional connectivity between nodes i and j. The 'functional connectivity' between two network nodes (i.e. ICA components) is operationalized as some measure of the similarity in the timeseries of the two brain regions. A variety of measures have been proposed as measures of functional connectivity, however by far the most common measure is a simple Pearson's correlation of the timeseries. There is some debate, however, about whether Pearson's correlation is a good measure that actually represents "true" connections between brain areas, and it has been proposed that a more accurate measure of direct connections might be to use the partial correlation between regions (Smith et al. 2011). With partial correlation, the time-series of all other network nodes is regressed out of nodes i and j before estimating their correlation to give a more accurate measure of the information uniquely shared between the two regions. Because there is no obvious ground truth for functional brain networks, it is difficult to determine which measure gives a more accurate representation of the connectivity structure of the brain. Prediction offers one possible solution to this problem. Under the hypothesis that perception, affect, cognition, and behavior result from brain activity, the brain network method that is best able to predict these might be the most accurate representation of the connectivity structure of the brain. For this reason, we have chosen to compare the predictive power of brain network constructed with both full and partial correlations.

Additionally from the HCP we have structural brain data. These measures were derived by running the FreeSurfer structural analysis pipeline on the T1 structural MRI scans of HCP subjects. FreeSurfer performs automated brain segmentation and cortical surface parellation producing accurate and reproducible measurements of brain structure (Fischl, 2012).

The NEO Five Factor Inventory

The NEO Five Factor Inventory (NEO-FFI) is a short, 60-item personality inventory designed to measure a person’s ‘Big Five’ personality traits: openness to experience, conscientiousness, extraversion, agreeableness, and neuroticism (Costa & McCrae, 1992). The ‘Big Five’ personality traits derive from the eponymous Five Factor Model, which suggests that a person’s personality can be described along the aforementioned five dimensions. The name ‘NEO’ is a historic reflection of the test’s original goal of measuring personality characteristics of neuroticism, extraversion, and openness to experience, with agreeabless and conscientiousness more recent additions not reflected in the test name. The NEO-FFI is completed in a self-report format, and contains 12 items designed to assess each of the five factors. Each item consists of a statement about the participant, for example “I donâ□□t get much pleasure chatting with people.” Participants are required to rate their agreement with each given statement as ‘Strongly Disagree’, ‘Disagree,’ ‘Neutral’, ‘Agree’, or ‘Strongly Agree.’ Scores are derived for each of the Five Factors based on the participants answers. Briefly, points are allotted to each item based on the participant’s answer, and a Five Factor score is generated by summing the points for the 12 items associated with a factor and assigning a T value based on test norms. The NEO-FFI has been widely validated for use in the assessment of personality in healthy populations and for the assessment of clinically-defined personality disorders according to the Diagnostic and Statistical Manual of Mental Disorders - Version 5 (Trull, 2012). In addition to other various self-report measures, The NEO-FFI was collected on HCP participants and we include both the raw data (i.e. score on each of 60 items) and the Five Factor data (i.e. score for each of the five factors) in our dataset.

Network Measures

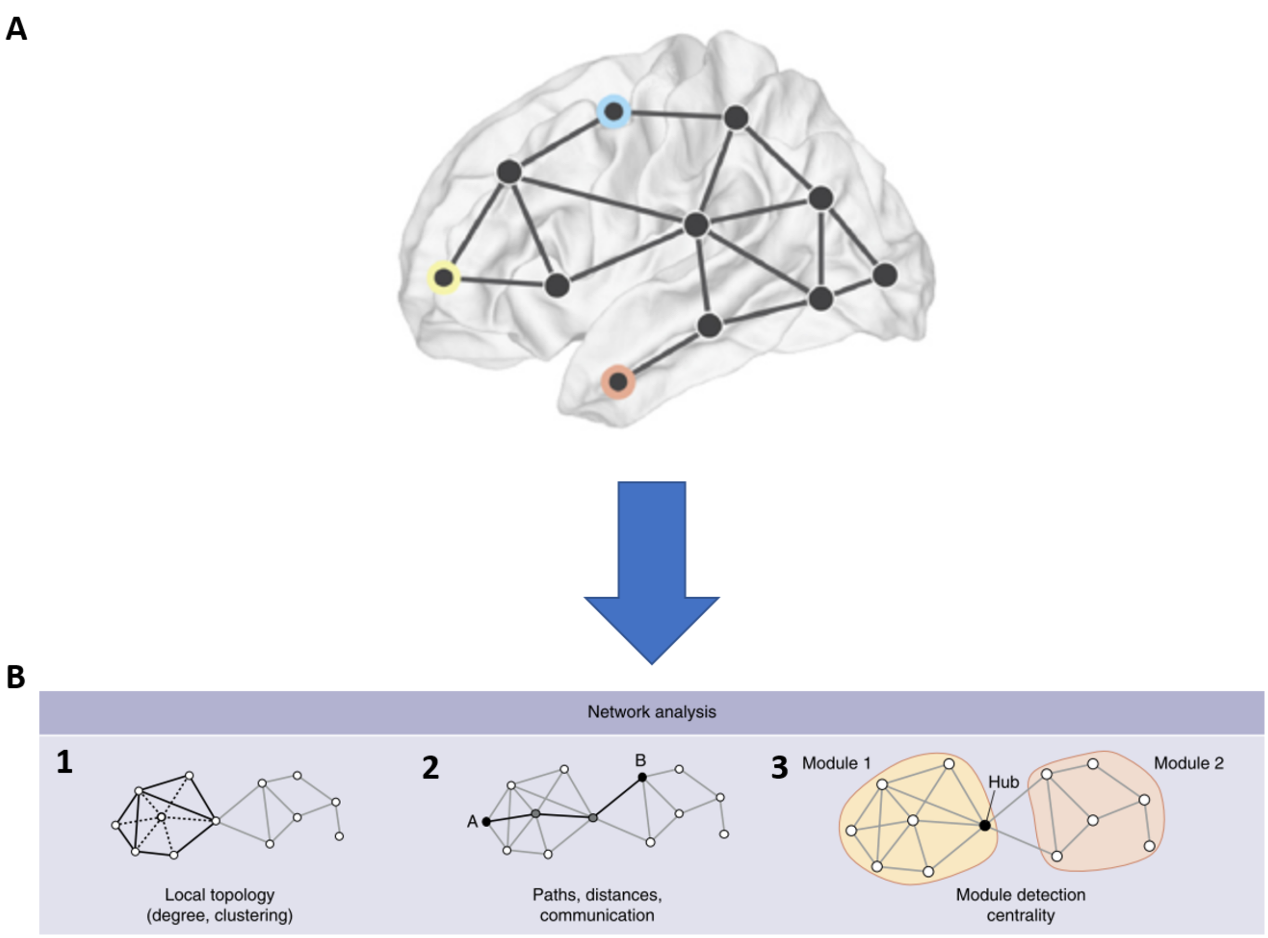
Recently, the field of neuroscience has begun to incorparte mathematical formalisms from graph theory to try and succintly quantify how regions of the brain interact as a cohesive network. In this approach, brain regions are defined as nodes, connected by edges (Fig 1A). The presence of an edge (usually a statistical interdependency, such as correlation of activity) is interpretted as an indication that two regions have similar activity, and are said to be functionally connected. We have chosen several graph statistics common to neuroscience to include in our analysis, with the hypothesis that they might better capture meaningful features of the brain than individual connections alone. While others have explored the relationship between edges, and personality measures, it is unclear how these more comprehensive summary statistics will explain variances in NEO-FFI scores.

These measures can be divided into two categories: (1) node statistics, which describe properties of a single node, and it's connections to other nodes, and (2) network statistics, which describe the structure of all connections in a graph. We will first discuss network statistics. The first network statistic calculated is the $global\ efficiency$ (Fig B2). This measure is proportional to the average distance between a single node, to any other nodes by travelling along edges. The next measure is a $community\ partition$ (Fig B3). This breaks a nextwork into communities, where the nodes in a community will have more connections within that community, than between between communities. We will now discuss node statistics. The first, and most intuitive measure is $strength$ (Fig B1). The strength is simply the sum of all edges connected to a given node. The second measure is $betweenness\ centrality$. Betweenness centrality is the fraction of all shortest paths in a network that pass through a given node. Intuitively, nodes with high betweenness centrality are thought to be important for communication between many, distant groups. The last measure is $within\ module\ degree$, which quantifies the proportion of a nodes connections that are within a community.

Firgure 1: Schematic of Network Measures

HIDE

```
knitr::include_graphics("/Users/jenis/Documents/R/Final/full_network_ex.png")
```



HIDE

```
#Bassett & Sporns 2017
```

Data Cleaning

Variable Selection

First, we exclude some variables and modify some variables included in the HCP data release that are not of interest for the current project. In particular, the FreeSurfer brain anatomy data includes a number of measures of no interest. Some examples include `FS_BrainSeg_Vol` which is a measure of total volume of the brian segmentation. This is redundant with the measure `FS_IntraCranial_Vol` which is a measure of total intracranial volume and is more commonly used as a measure of general head size. Additionally, other measures we exclude here are general measures of the entire brain or measures of brain tissue classes instead of specific brain regions, such as `FS_Tot_WM_Vol` which is the measure of total white-matter volume. Also some of the FreeSurfer measures are not measurements of the brain, but rather measurements of the quality of the brain segmentation algorithms, such as `FS_Total_Defect_Holes` .

We exclude these variables and others from our dataset:

HIDE

```
data2 <- lapply(data, function(x) {

  x %>%
    dplyr::select(-X, -Subject, -FS_BrainSeg_Vol, -FS_BrainSeg_Vol_No_Vent, -FS_BrainS
eg_Vol_No_Vent_Surf,
                  -FS_LCort_GM_Vol, -FS_RCort_GM_Vol, -FS_TotCort_GM_Vol, -FS_SubCort_
GM_Vol, -FS_Total_GM_Vol, -FS_SupraTentorial_Vol, -FS_R_WM_Vol, -FS_Tot_WM_Vol, -FS_Ma
sk_Vol,  -FS_BrainSegVol_eTIV_Ratio, -FS_MaskVol_eTIV_Ratio, -FS_LH_Defect_Holes,  -FS
_RH_Defect_Holes, -FS_Total_Defect_Holes, -FS_L_WM_Hypointens_Vol, -FS_R_WM_Hypointens
_Vol, -FS_WM_Hypointens_Vol)
})
```

Scaling

Of the brain structural measures we retain, there are two types of measures. For subcortical brain regions we have measures of regional volume, and for cortical brain regions we have measures of cortical thickness. Thickness is used for cortical structure because the cortex is represented as a set of two two-dimensional surfaces, and the thickness is calculated as the eucilidean distance between corresponding vertices of these two surfaces.

To control for the effect of head size and brain size all of the subcortical volumes, we scale by the total intracranial volume estimated by FreeSurfer.

HIDE

```
fs_scaled <- as.data.frame(sapply(data2$full[ , grep("FS_.*_Vol", names(data2$full))],
 function(x) {
  x/data2$full$FS_IntraCranial_Vol
}))
fs_scaled <- fs_scaled[,-1]
names(fs_scaled) <- paste0(names(fs_scaled), "_scaled")
```

The scaled volumes are retained, and all unscaled volumes excluding total intracranial volume are dropped.

HIDE

```
data3 <- lapply(data2, function(x) {
  cbind(x, fs_scaled)
})

data3 <- lapply(data3, function(x) {
  x[,-grep("FS_.*_Vol$", names(x))[-1]]
})
```

Separation of NEO data and HCP data into distinct dataframes

Next, for use in our analyses, we separate the HCP data from the NEO-FFI data.

First, we save out the Five Factor score data and raw NEO-FFI data each to a separate dataframe.

HIDE

```
neoFFData <- data3$full[, grep("NEOFAC.*", names(data3$full))]
neoRData <- data3$full[, grep("NEORAW.*", names(data3$full))]
```

Last, we remove all raw NEO data and Five Factor Score data from the original matrix, retaining only HCP measures.

HIDE

```
data4 <- lapply(data3, function(x) {
  x[,-grep("NEO*", names(x))]
})
lapply(data4, dim)
```

```
## $full
## [1]   810 11734
##
## $partial
## [1]   810 11734
```

Our final cleaned HCP dataset has 810 observations and 11734 features.

Split into Training and Test Samples

We split the dataset into training and testing samples. We choose to make this split to separate the testing data from data used to fit parameters, despite the fact that random forest's out of bag predictions can be used for an out of sample prediction The training dataset includes 710 observations and the testing set includes 100. We chose a rather high training fraction because we have a wide dataset and would like to retain as many observations as possible for training. One hundred observations should be sufficient to estimate the out-of-sample testing error.

HIDE

```
set.seed(1)
idx = sample(1:810, size=100)

data.train <- lapply(data4, function(x){
  x[-idx,]
})

data.test <- lapply(data4, function(x){
  x[idx,]
})

neoFFData.train <- neoFFData[-idx,]
neoFFData.test <- neoFFData[idx,]
```

Exploratory Data Analysis

Demographic variables of participant sample

Age

Before making any predictions, we first want to explore some of the more intuitive variables in the dataset.

We note that the age of most participants ranges from 22-35, with a few older individuals.

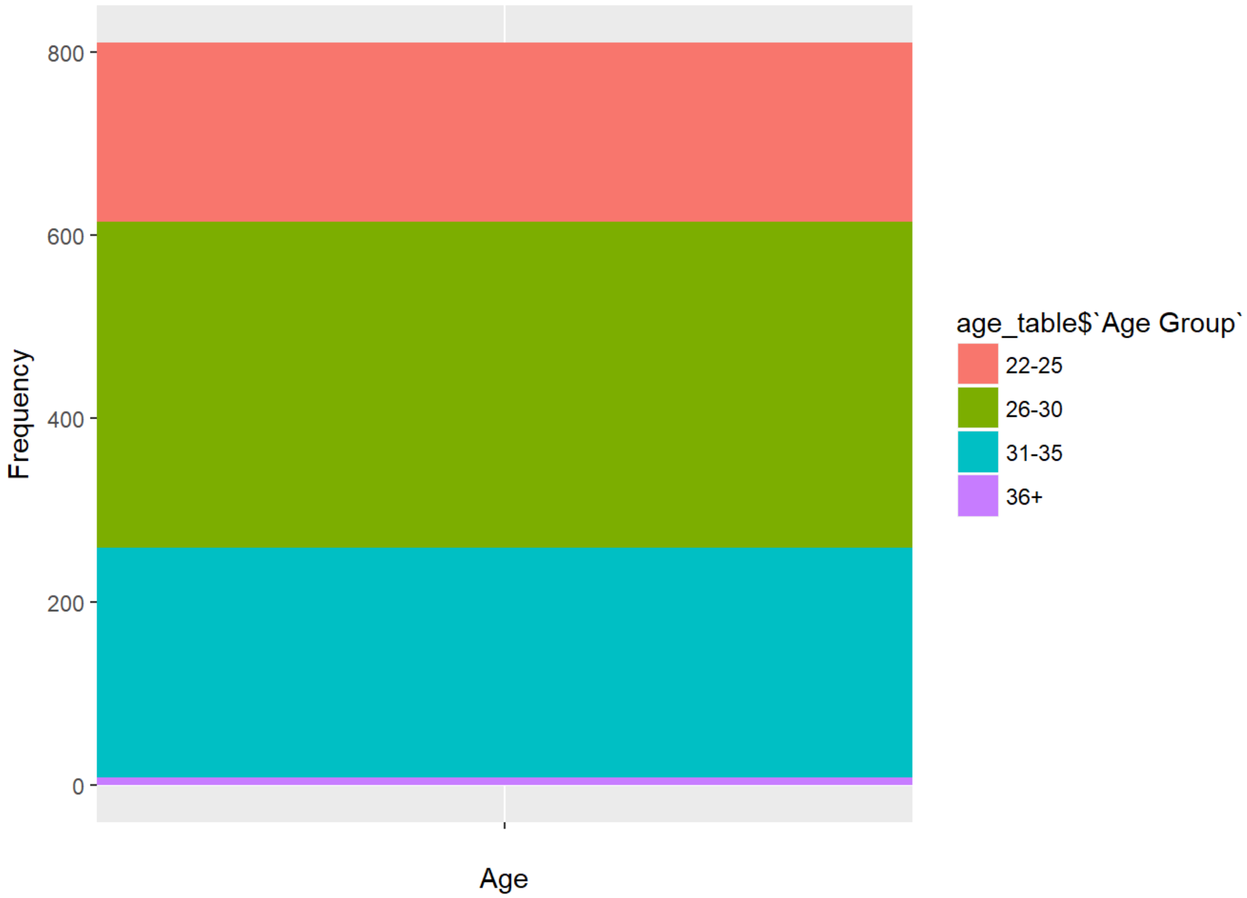
HIDE

```
age_table <- as.data.frame(table(data4$full$Age))
colnames(age_table) <- c("Age Group", "Frequency")
age_table
```

```
##   Age Group Frequency
## 1     22-25       195
## 2     26-30       356
## 3     31-35       251
## 4       36+         8
```

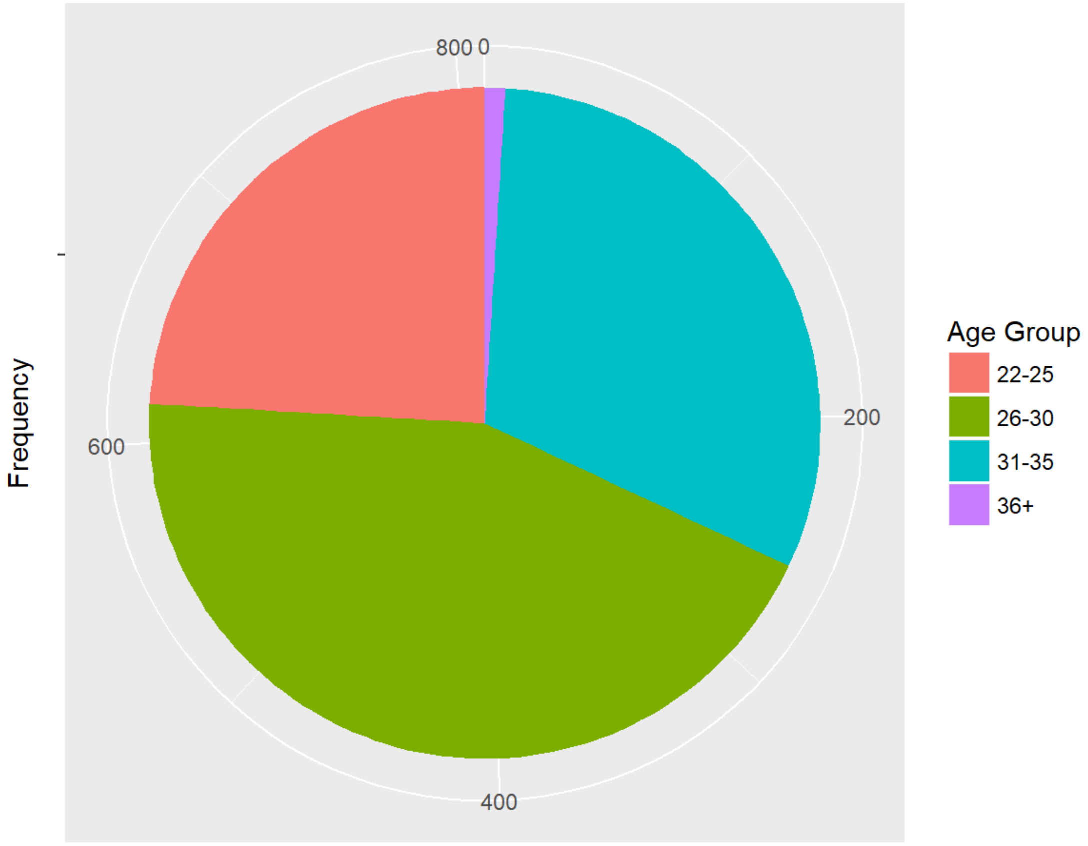
HIDE

```
age_bp<- ggplot(age_table, aes(x=" ", y=age_table$Frequency, fill=age_table$`Age Group
`))+
geom_bar(width = 1, stat = "identity") +
  ylab("Frequency") +
  xlab("Age")
age_bp
```



HIDE

```
age_pie <-  age_bp+coord_polar("y", start=0)+
    ylab("") +
    xlab("Frequency")+
    scale_fill_discrete(name="Age Group")
age_pie
```



### Gender

Sex is almost evenly split between male and female.

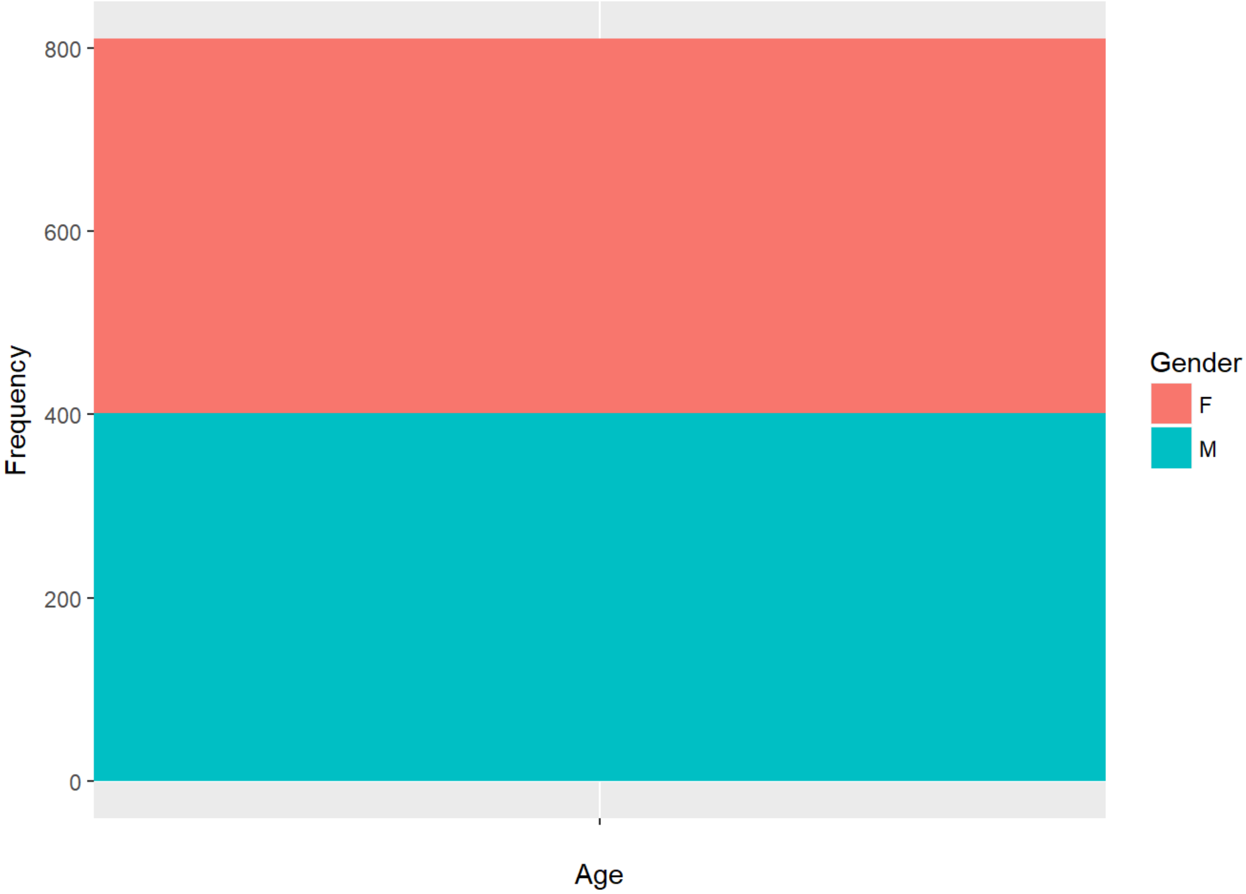
HIDE

```
gender_table <- as.data.frame(table(data4$full$Gender))
colnames(gender_table) <- c("Gender", "Frequency")
gender_table
```

```
##   Gender Frequency
## 1      F       408
## 2      M       402
```

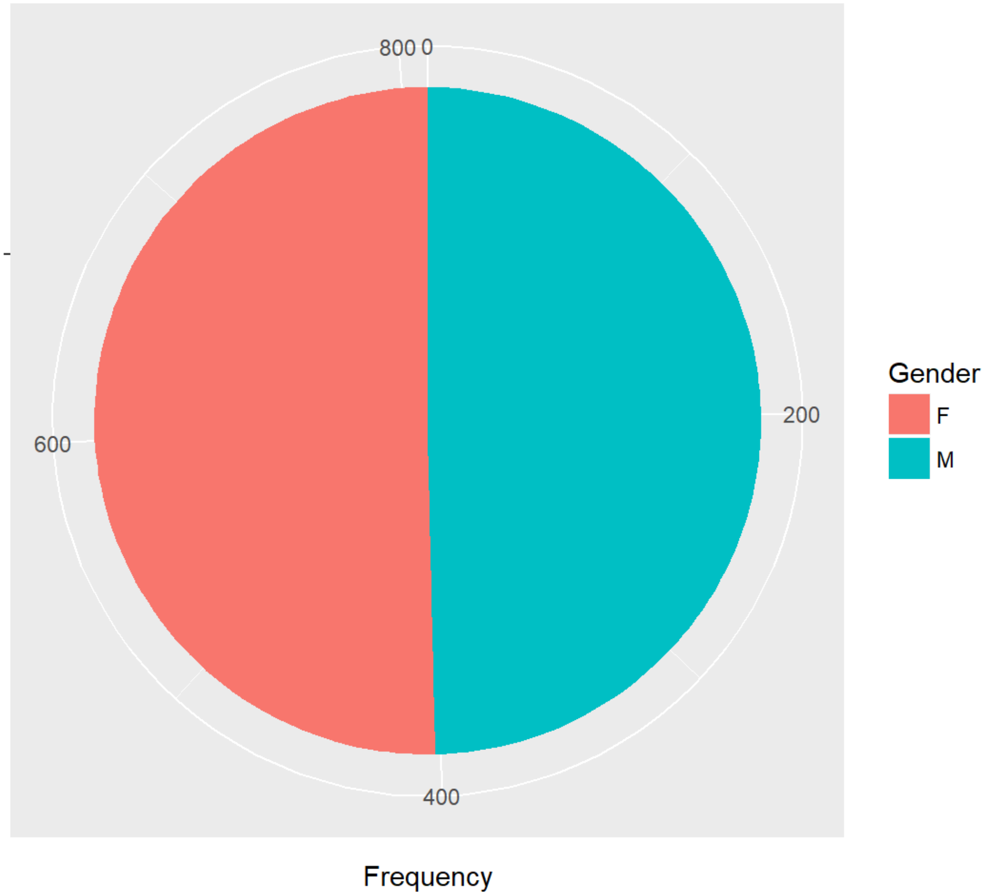
HIDE

```
gender_bp<- ggplot(gender_table, aes(x=" ", y=Frequency, fill=Gender))+
geom_bar(width = 1, stat = "identity")+
  ylab("Frequency") +
  xlab("Age")
gender_bp
```



HIDE

```
gender_pie <-  gender_bp+coord_polar("y", start=0) +
  xlab(" ")+
  ylab("Frequency")
gender_pie
```



Confirmatory Factor Analysis of NEO-FFI data

Here, we first double-check whether the Five Factor Model measures of Openness to Experience, Conscientiousness, Extraversion, Agreeableness, and Neuroticism indeed account for the maximal variance in the NEO data set by performing confirmatory factor analysis (CFA).

We first convert raw NEO data to numeric form for use in analysis.

HIDE

```
neoRData[] <- lapply(neoRData, as.character)
neoRData[neoRData=="SD"] <- 1
neoRData[neoRData=="D"]<- 2
neoRData[neoRData=="N"] <- 3
neoRData[neoRData=="A"] <- 4
neoRData[neoRData=="SA"] <- 5
neoRData[] <- lapply(neoRData, as.numeric)
```

We then create a model matrix to input into the CFA; here, we model the loadings of the NEO raw items onto each of the Five Factors.

HIDE

```
ffmodel <- '
# defining latent variables based on five factors
Neuroticism =~ NEORAW_01 + NEORAW_06 + NEORAW_11 + NEORAW_16 + NEORAW_21 + NEORAW_26 +
 NEORAW_31 + NEORAW_36 + NEORAW_41 + NEORAW_46 + NEORAW_51 + NEORAW_56

Extraversion =~ NEORAW_02 + NEORAW_07 + NEORAW_12 + NEORAW_17 + NEORAW_22 + NEORAW_27
 + NEORAW_32 + NEORAW_37 + NEORAW_42 + NEORAW_47 + NEORAW_52 + NEORAW_57

Openness =~ NEORAW_03 + NEORAW_08 + NEORAW_13 + NEORAW_18 + NEORAW_23 + NEORAW_28 + NE
ORAW_33 + NEORAW_38 + NEORAW_43 + NEORAW_48 + NEORAW_53 + NEORAW_58

Agreeableness =~ NEORAW_04 + NEORAW_09 + NEORAW_14 + NEORAW_19 + NEORAW_24 + NEORAW_29
 + NEORAW_34 + NEORAW_39 + NEORAW_44 + NEORAW_49 + NEORAW_54 + NEORAW_59

Conscientiousness =~ NEORAW_05 + NEORAW_10 + NEORAW_15 + NEORAW_20 + NEORAW_25 + NEORA
W_30 + NEORAW_35 + NEORAW_40 + NEORAW_45 + NEORAW_50 + NEORAW_55 + NEORAW_60
'
```

Now, we run the CFA.

HIDE

```
fit <- cfa(ffmodel, data=neoRData)
```

We observe that the Comparative Fit Index (CFI) 0.653 and the Tucker-Lewis Index (TLI) is 0.639, indicating that the specified model approximates a moderately good fit of the data (ideal CFI and TLI would be > 0.9). The summary showing this information is located in the Appendix.

The RMSEA (Root mean square error of approximation) measures the â□□error of approximationâ□□ (i.e. residuals) and measures how closely the specified model reproduces data patterns (i.e. the covariances among indicators). The RMSEA observed here is 0.059 with a 90% CI of 0.058, 0.061. The P-value associated with the observed RMSEA is < 0.05, suggesting that the Five Factor model fits the raw NEO data very well.

Based on these results, we proceed with using the Five Factor data rather than the raw NEO data for ease of interpretability.

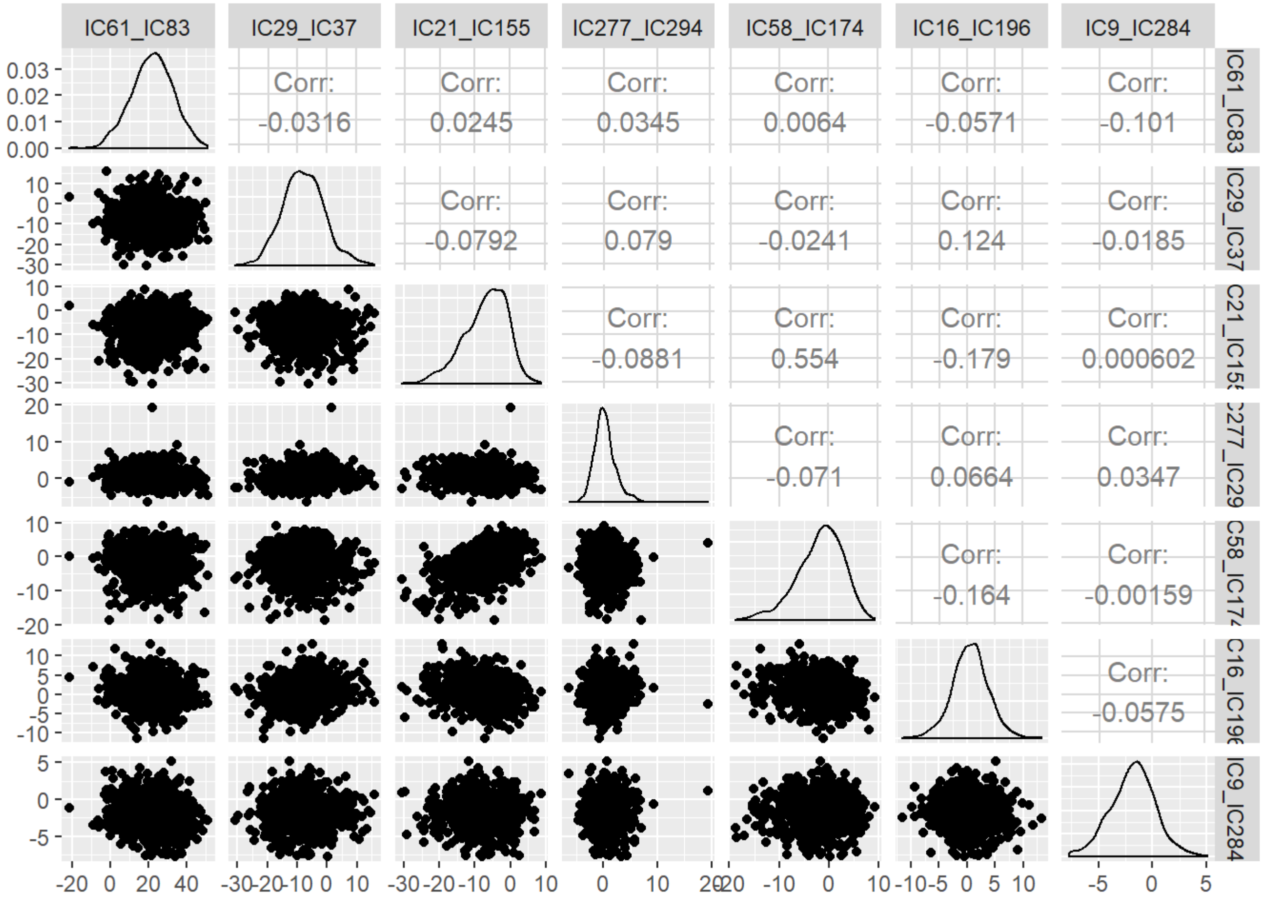
Correlation matrices of Network Measures

Brain activity naturally autocorrelates with itself over different brain regions, and because these networks are constructed by taking the pair-wise correlation or partial correlation between the time-series of each network node we expect the edge data to be highly correlated. The brain networks are composed of 149 nodes and pairwise connections between each node pair. Note that while there are 149 nodes included, the names range from 1 to 300, as a result of the original partition. Because the networks are undirected, this results in `149^2/2-149/2 = 11026` unique network edges. For example, feature `IC63_IC283` is the connectivity between nodes 63 and 283, while `IC63_IC145` is the connectivity between nodes 63 and 145. Because these components share a node it should be expected that they will correlate with each other. Since there are so many edges in the netwrok it is infeasible to explore the correlation between all possible pairs, so we will take multiple random samples of the edge data and plot the im GGPairs plots. The GGPairs plots are a useful way to display this EDA, because it shows distribution density plots of each variable, the correlations between variable pairs, and the scatter plots of the variable pairs.

Sample correlation of edges from full correlations:

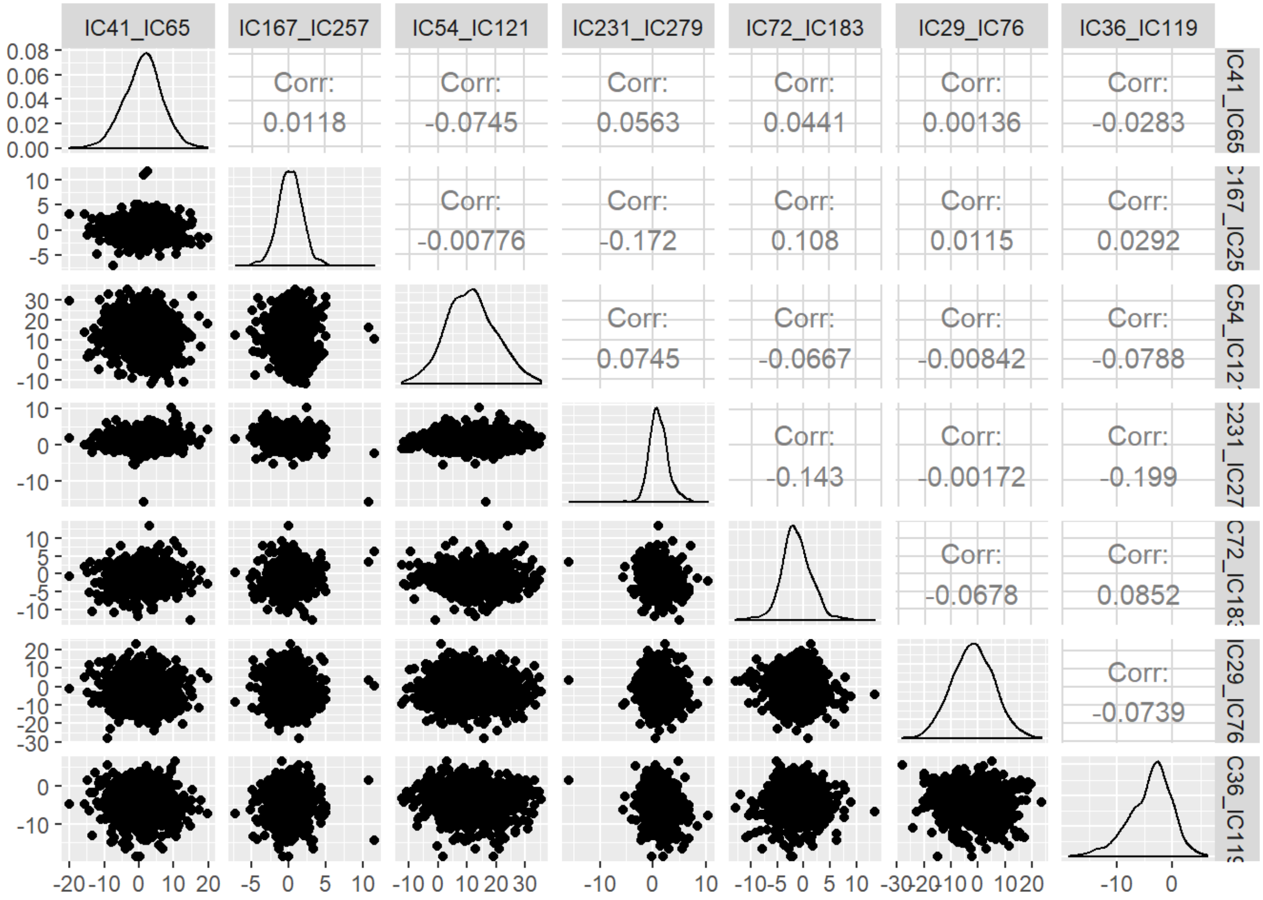
HIDE

```
ic_idx <- grep("IC.*_IC.*", names(data4$full))
ic_idx1 <- sample(ic_idx, 7)

ggpairs(data4$full[,ic_idx1])
```



HIDE

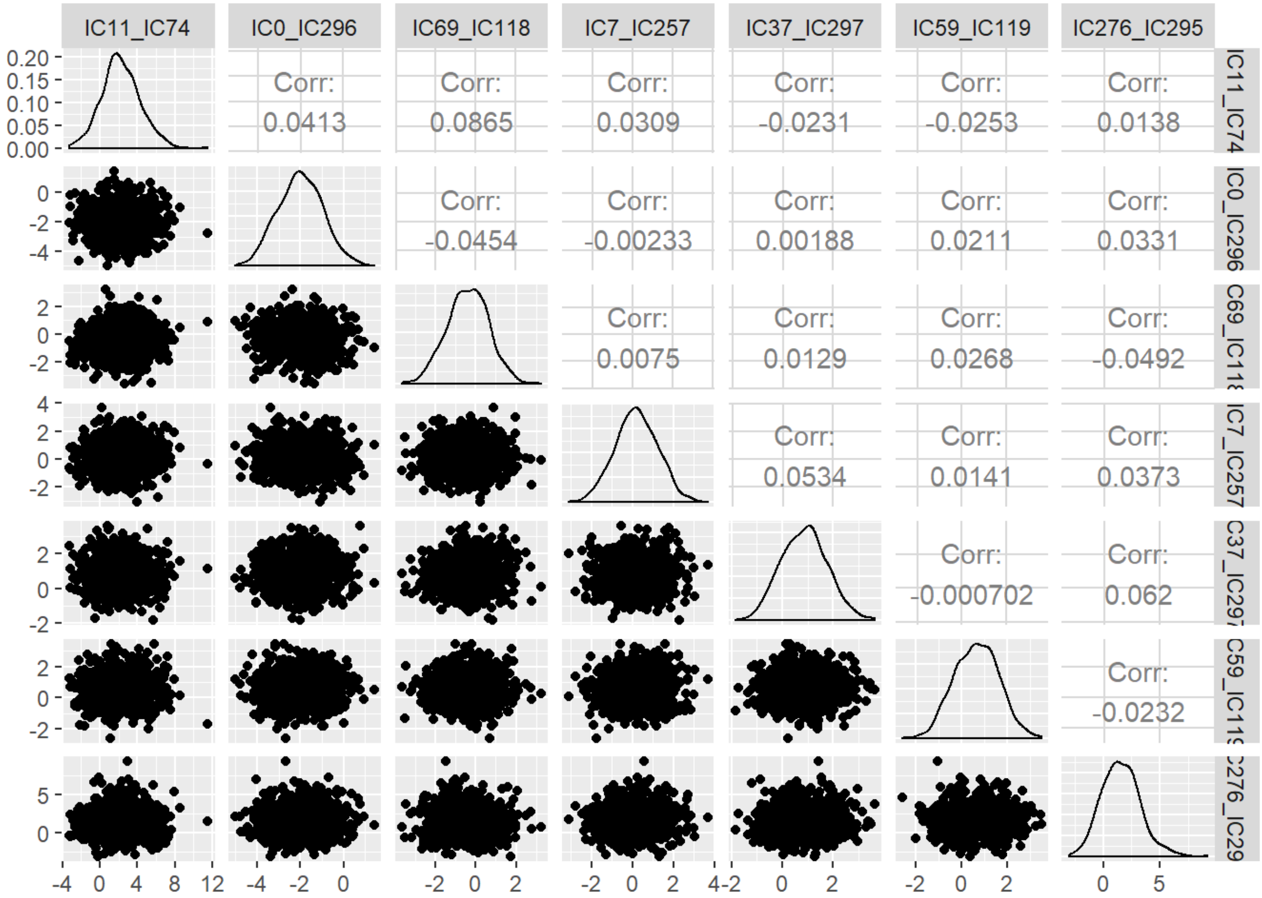
```
ic_idx <- grep("IC.*_IC.*", names(data4$full))
ic_idx1 <- sample(ic_idx, 7)

ggpairs(data4$full[,ic_idx1])
```



Sample correlation of edges from partial correlations:

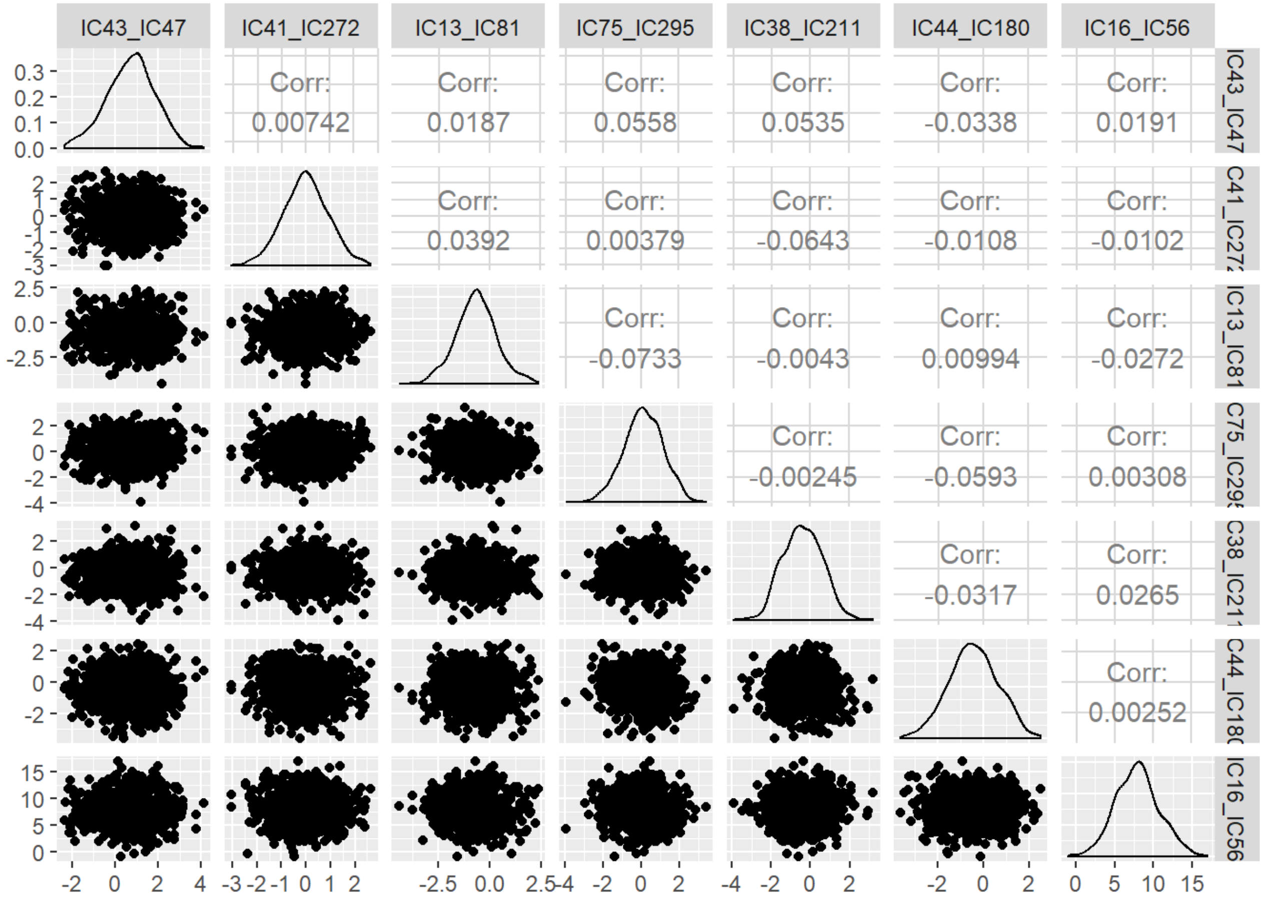
HIDE

```
ic_idx <- grep("IC.*_IC.*", names(data4$partial))
ic_idx1 <- sample(ic_idx, 7)

ggpairs(data4$partial[,ic_idx1])
```



HIDE

```
ic_idx <- grep("IC.*_IC.*", names(data4$partial))
ic_idx1 <- sample(ic_idx, 7)

ggpairs(data4$partial[,ic_idx1])
```

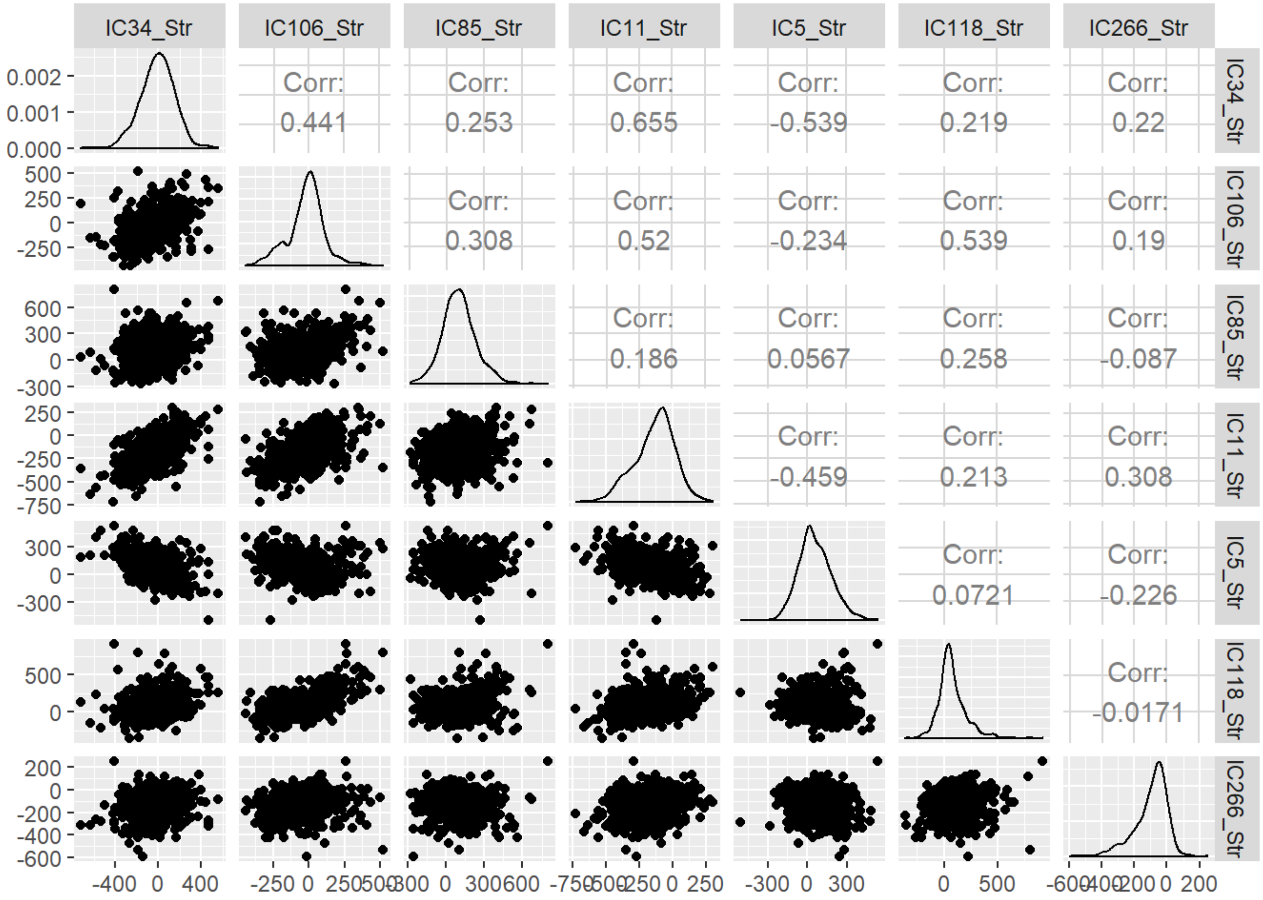


As expected, many of these edges are correlated with eachother. We next explore some of our node level graph statics in a similar format.

Sample correlation of nodewise strength and Betweenness Centrality from full correlations

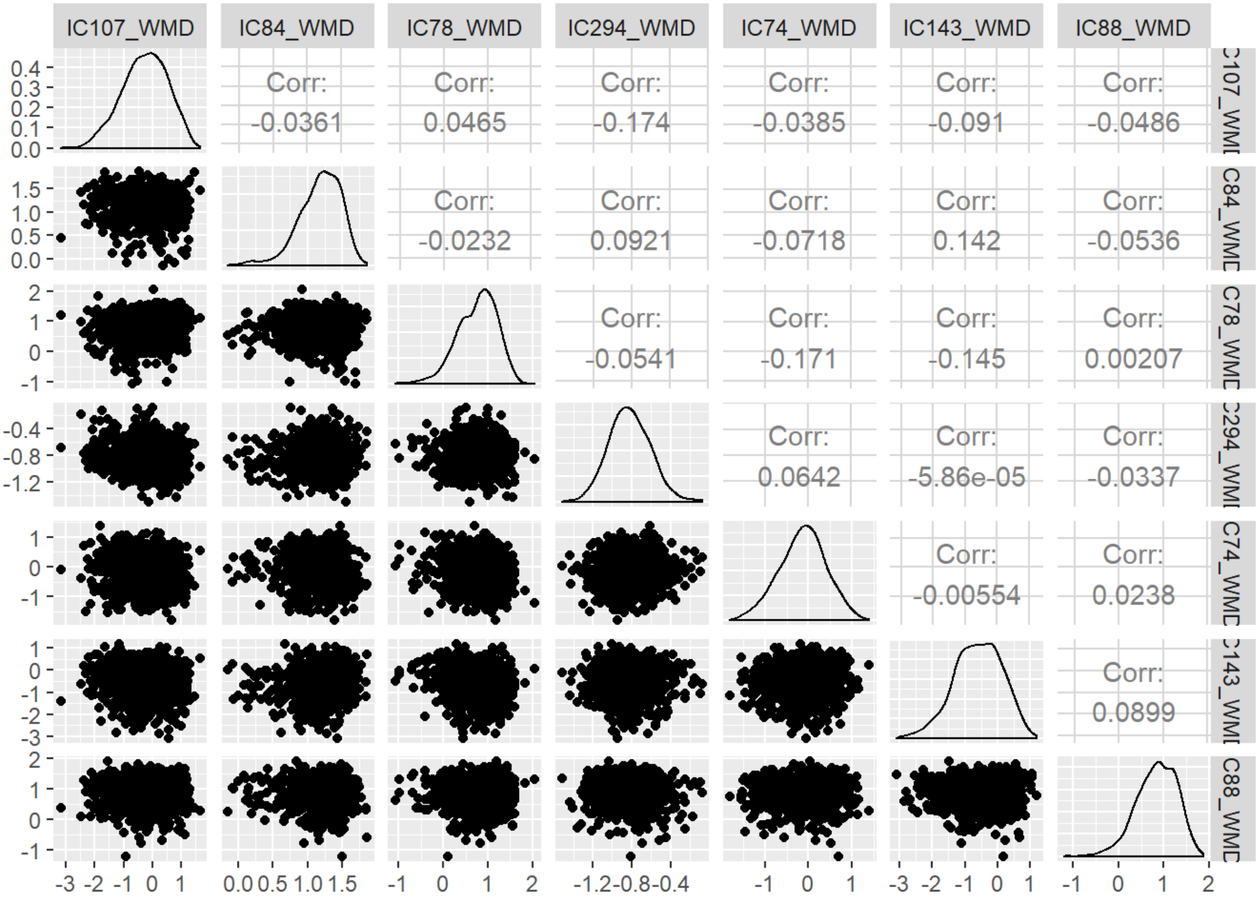
HIDE

```
ic_idx <- grep("IC.*_Str*", names(data4$full))
ic_idx1 <- sample(ic_idx, 7)

ggpairs(data4$full[,ic_idx1])
```



HIDE

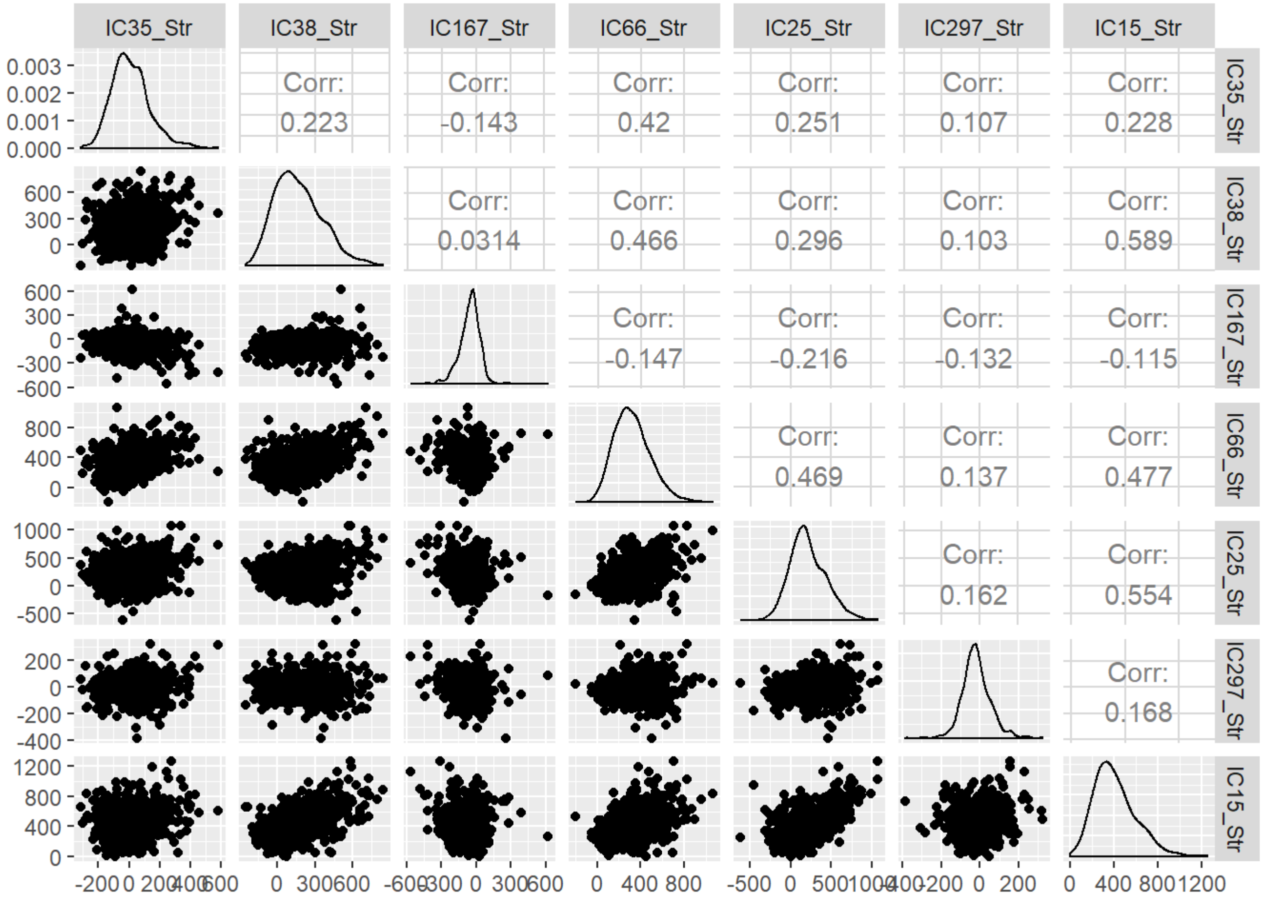
```
ic_idx <- grep("IC.*_WMD.*", names(data4$full))
ic_idx1 <- sample(ic_idx, 7)

ggpairs(data4$full[,ic_idx1])
```



Sample correlation of nodewise strength and betweeness centrality from partial correlations

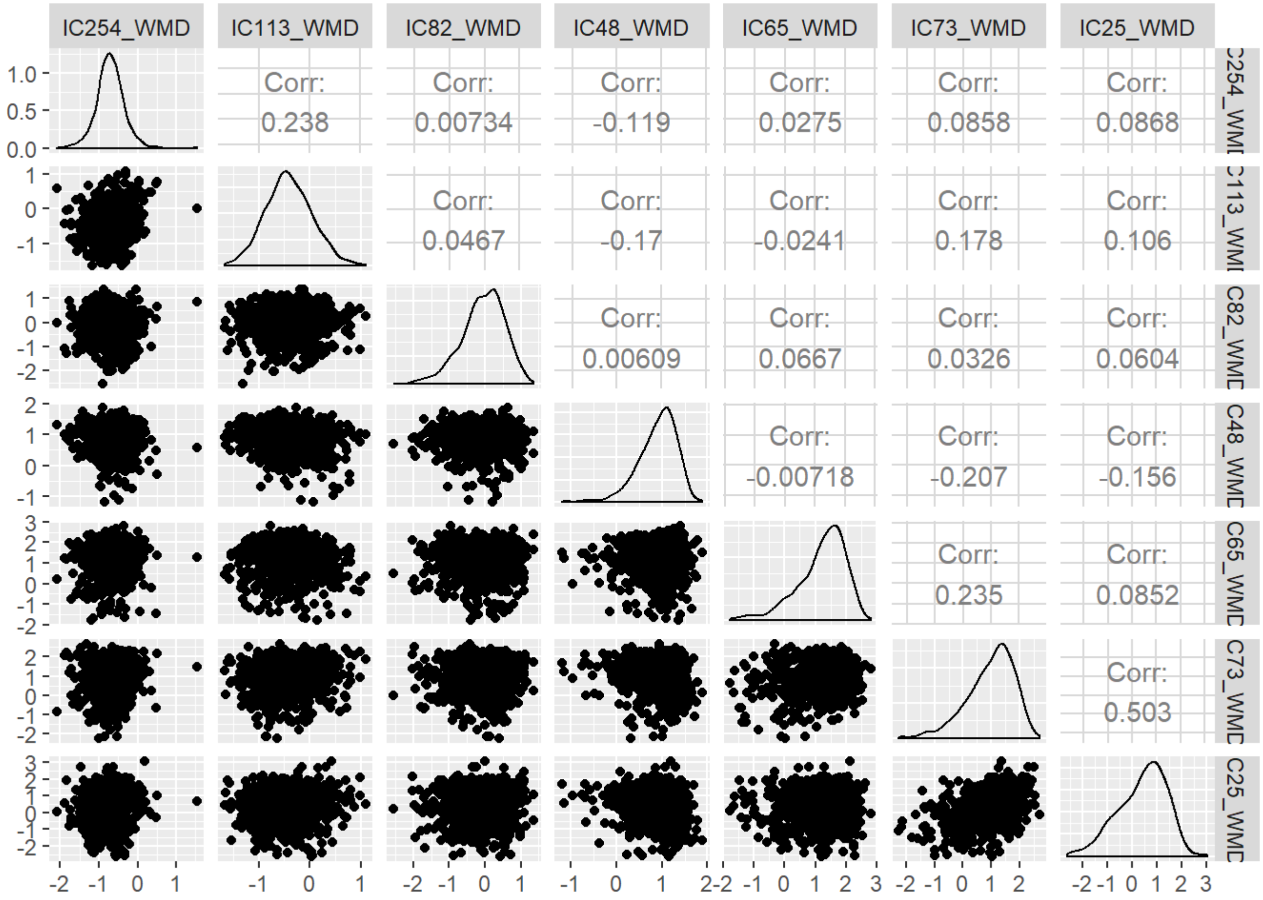
HIDE

```
ic_idx <- grep("IC.*_Str*", names(data4$partial))
ic_idx1 <- sample(ic_idx, 7)

ggpairs(data4$partial[,ic_idx1])
```



HIDE

```
ic_idx <- grep("IC.*_WMD.*", names(data4$partial))
ic_idx1 <- sample(ic_idx, 7)

ggpairs(data4$partial[,ic_idx1])
```

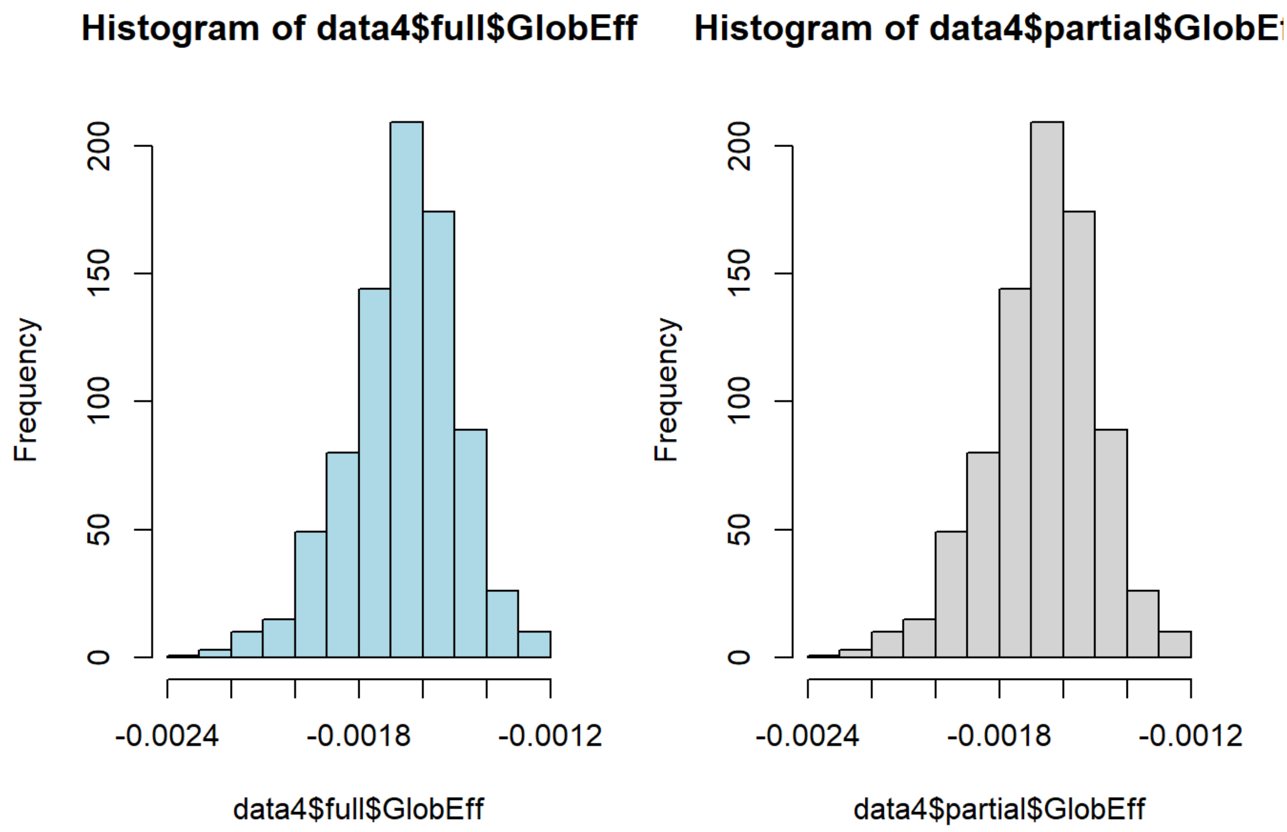


We note that strength appears to have notable correlation across nodes. Lastly, we will visualize out network wide graph statistic: global efficiency.

Network wide graph measures: Global Efficiency

HIDE

```
par(mfrow=c(1,2))
hist(data4$full$GlobEff, col = 'lightblue')
hist(data4$partial$GlobEff, col = 'lightgrey')
```



Both partial and full correlations have similar (nearly normal) distributions of global efficiency.

Analysis

We will first fit a random forest model to the data. Random forest models are ideal because they can capture nonlinear relationships, and because elastic net based methods are not designed for correlated variables. We also calculated boosting combined with random forest, hypothesizing that we will gain predictive power.

Parameter Selection

Random Forest

First we format the data

HIDE

```
data.train.full.neoC <- cbind(neoFFData.train$NEOFAC_C, data.train$full)

neoNames = c('NEOFAC_A', 'NEOFAC_O', 'NEOFAC_C', 'NEOFAC_N', 'NEOFAC_E')

df.full <- lapply(neoNames, function(x){
  NEO <- neoFFData.train[,which(names(neoFFData.train) == x)]
  cbind(NEO, data.train$full)
})
names(df.full) <- neoNames

df.partial <- lapply(neoNames, function(x){
  NEO <- neoFFData.train[,which(names(neoFFData.train) == x)]
  cbind(NEO, data.train$partial)
})
names(df.partial) <- neoNames
```

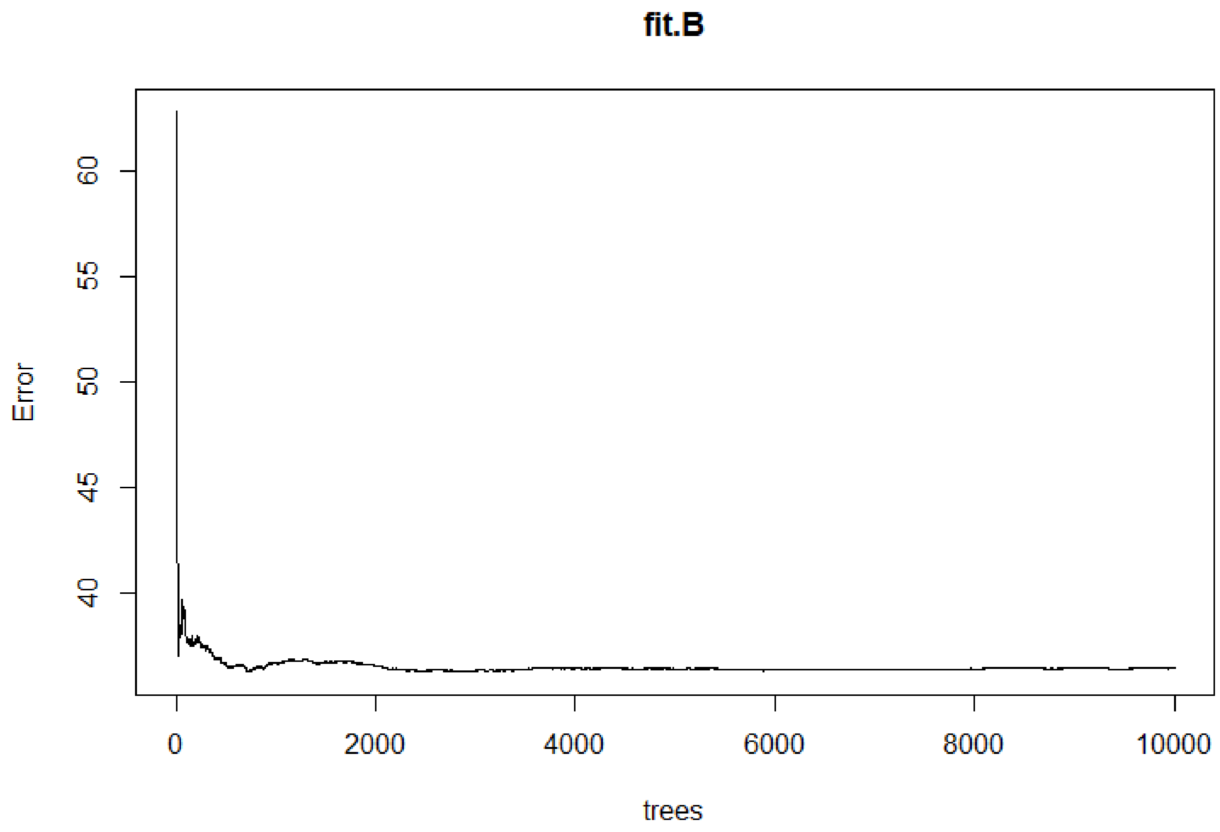
Here, we fit B, the number of trees. One predictor is shown for brevity, but other predictors looked similar. B equal to 1500 was selected.

HIDE

```
# tune B
fit.B = randomForest(NEOFAC_C~., data = df.full$NEOFAC_C, ntree = 10000, mtry = 3990)
# set mtry to p/3 for now
plot(fit.B)
B = 1500
```

HIDE

```
knitr::include_graphics("/Users/jenis/Documents/R/Final/B.png")
```



It is important to also tune mtry, the number of random variables selected for each split. However, due to the run time of these models, we were only able to tue mtry on one model, with 3 different random seeds. This model indicated little effect of the variable, and therefore we used the default value of mtry = p/3, where p is the number of predictors in successive analyses.

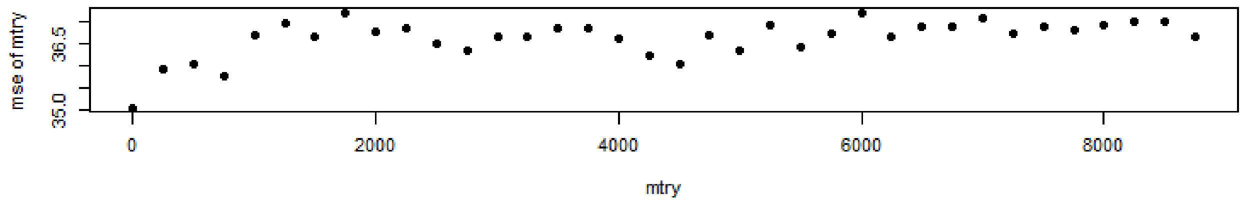
Tune mtry

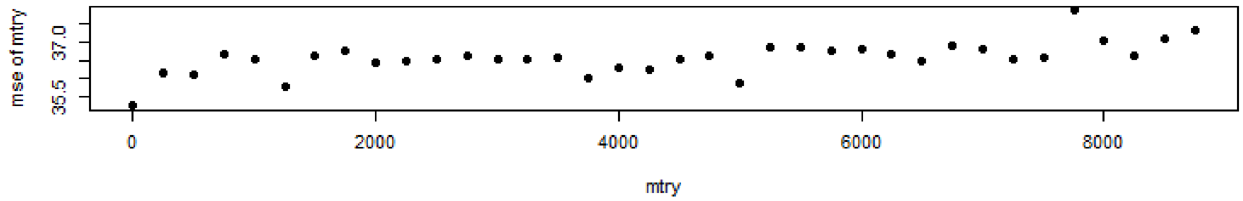
HIDE

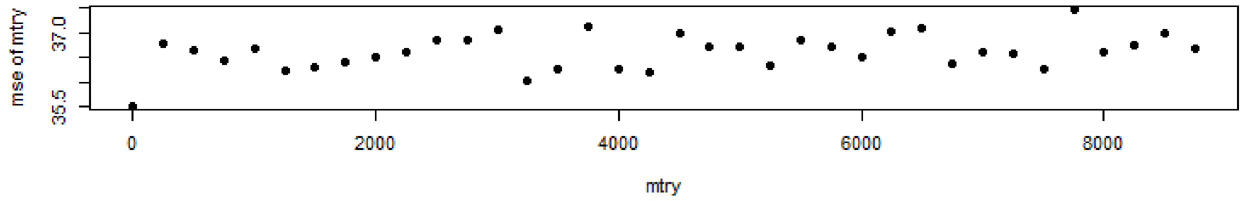
```
ps = seq(from = 1, to = 6000, by = 500)
rf.error.p <- numeric(length(ps))  # set up a vector
cnt = 0;
for (p in ps)  # repeat the following code inside { } 19 times
{
  print(p)
  cnt = cnt + 1
  fit.rf <-ranger(NEO~., df.partial$NEOFAC_E, mtry = mtry, num.trees = 1500)# set mtry
 to p/3 for now
  #plot(fit.rf, col= p, lwd = 3)
  rf.error.p[cnt] <- fit.rf$prediction.error  # collecting oob mse based on B trees
}
```

HIDE

```
knitr::include_graphics("/Users/jenis/Documents/R/Final/mtry.png")
```







Boosting

Here we attempt to optimize the hyper-parameter choices for our boosing model. This code is currently being run and we anticipate that the properly tuning the parameters of our boosting model will enable us to achieve better predictions on the data. The boosting models presented (code below) have not had paramteter tuning for number of trees or interaction depth, however they provide us some insight on the efficacy of the boosting model for out data.

HIDE

```
gbmGrid <- expand.grid(interaction.depth = c(1,2,4,6,8), n.trees = c(5000, 10000, 1500
0), shrinkage = .1, n.minobsinnode = 10)
fitControl <- trainControl(method = "repeatedcv", number = 10, repeats = 10)


fits.gbm.full <- lapply(df.full, function(x) {
  set.seed(10)
  train(NEO ~ ., data = x,
                 method = "gbm",
                 trControl = fitControl,
                 verbose = TRUE,
                 tuneGrid = gbmGrid,
                 metric = "RMSE",
                 distribution = "gaussian")
})

fits.gbm.partial <- lapply(df.partial, function(x) {
  set.seed(10)
  train(NEO ~ ., data = x,
                 method = "gbm",
                 trControl = fitControl,
                 verbose = TRUE,
                 tuneGrid = gbmGrid,
                 metric = "RMSE",
                 distribution = "gaussian")
})
```

Model Fits

All code below for fitting the final models has been evaluated elsewhere and the model fits were saved as `.Rdata` to be loaded later for evaluation.

Code for fitting random forest models. Mtry is p/3 and ntree is 1500 as evaluated in the parameter tuning section.

HIDE

```
mtry = floor(dim(data.train$full)[2]/3)

fits.rf.full <- lapply(df.full, function(x) {
  randomForest(NEO~., x, mtry = mtry, ntree = 1500)
})

fits.rf.partial <- lapply(df.partial, function(x) {
  randomForest(NEO~., x, mtry = mtry, ntree = 1500)
})
```

Code for fitting the boosting models

HIDE

```
fits.gbm.full <- lapply(df.full, function(x) {
  gbm(NEO~., data = x, distribution = "gaussian", n.trees = 15000, train.fraction = 0.
7)
})

fits.gbm.partial <- lapply(df.partial, function(x) {
  gbm(NEO~., data = x, distribution = "gaussian", n.trees = 15000, train.fraction = 0.
7)
})
```

Predicting NEO agreeableness measure

Load random forest models

HIDE

```
load('./rf_models/fit_rf_full.RData')
load('./rf_models/fit_rf_partial.RData')
```

Here we use our RF model to make predictions on the testing dataset

HIDE

```
predict.full.A = predict(fits.rf.full$NEOFAC_A, data.test$full)
predict.partial.A = predict(fits.rf.partial$NEOFAC_A, data.test$partial)
predict.mean.A = rep(mean(neoFFData.test$NEOFAC_A), length(neoFFData.test$NEOFAC_A))
```

Below is the MSE for the prediction of NEO_A using the full and partial correlation data as well as the mean NEO_A

HIDE

```
mse.neoA.full <- mean((neoFFData.test$NEOFAC_A-predict.full.A)^2)
mse.neoA.partial <- mean((neoFFData.test$NEOFAC_A-predict.partial.A)^2)
mse.neoA.mean <- mean((neoFFData.test$NEOFAC_A-predict.mean.A)^2)

print("Table of mean squared error for random forest prediction of NEO-A")
```

```
## [1] "Table of mean squared error for random forest prediction of NEO-A"
```

HIDE

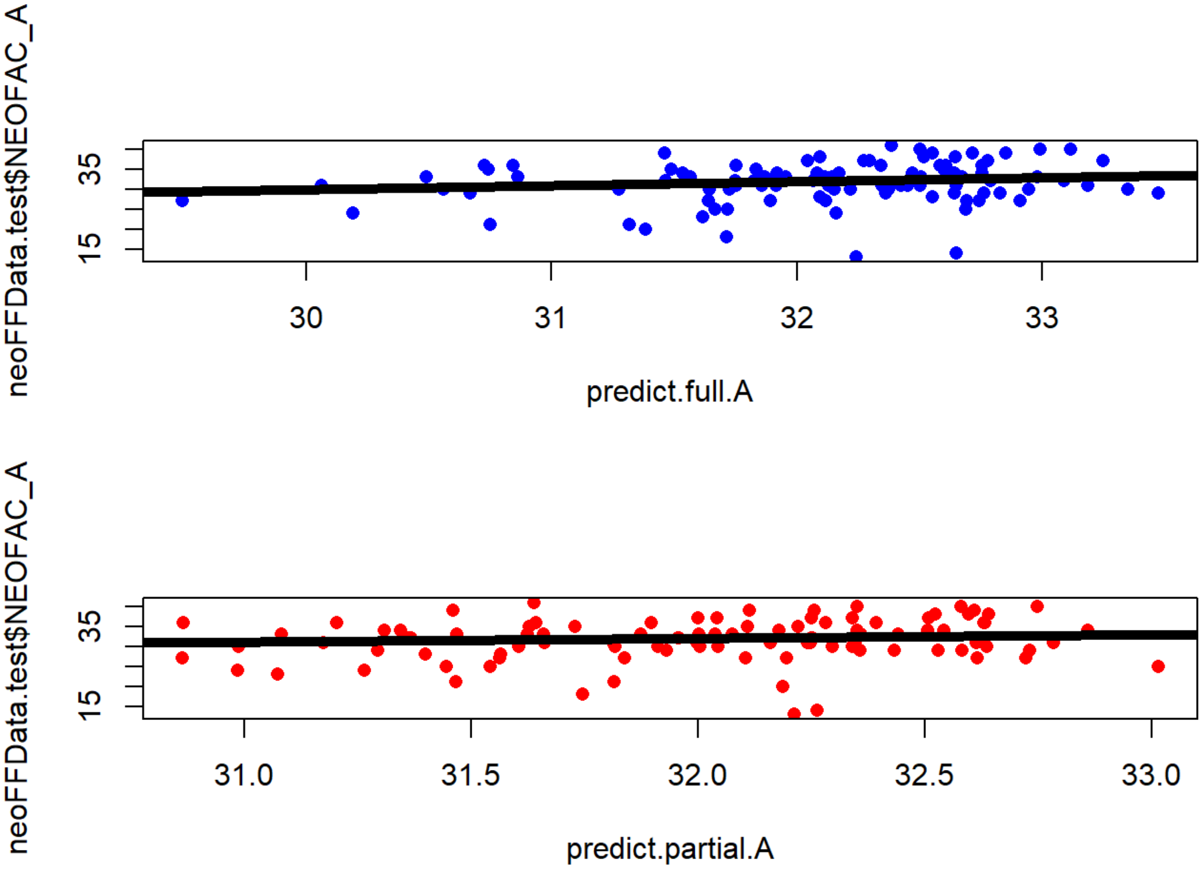
```
data.frame(Full = mse.neoA.full, Partial = mse.neoA.partial, mean = mse.neoA.mean)
```

```
##       Full  Partial    mean
## 1 27.49786 27.70348 28.1696
```

As we can see here, the models with both full correlation data and partial correlation data do very similarly, however they only do marginally better than the mean, indicating we are not doing a good job of predicting the scores.

HIDE

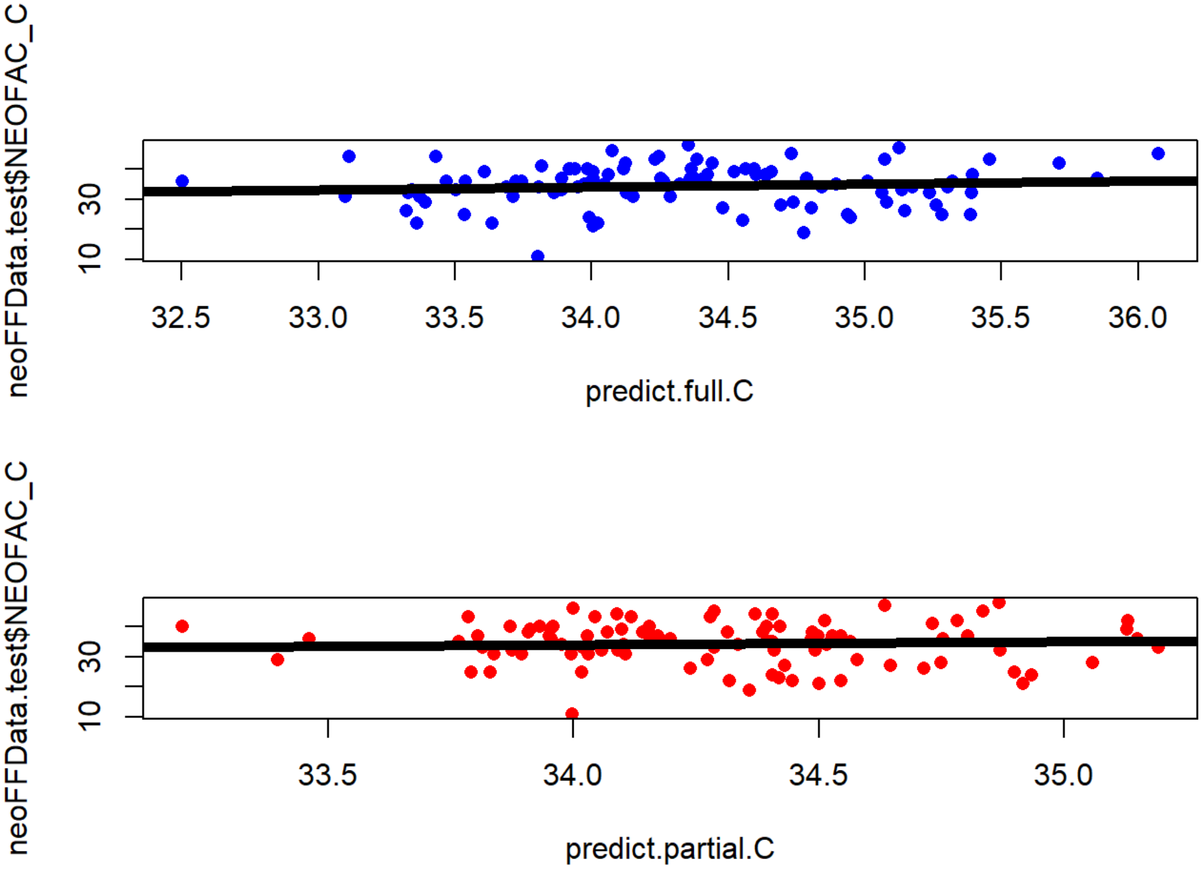
```
par(mfrow = c(2,1))
plot(predict.full.A, neoFFData.test$NEOFAC_A, pch = 16, col = 'blue')
abline(0, 1, lwd=5, col="black")
plot(predict.partial.A, neoFFData.test$NEOFAC_A, pch = 16, col = 'red')
abline(0, 1, lwd=5, col="black")
```



These predicted vs. measured plots appear to show a relativley good fit. The line is `Y = X` which would be a perfect fit. From these plots it is apparent that there is very low variability in the predicted values, ranging only from ~30 to ~33.

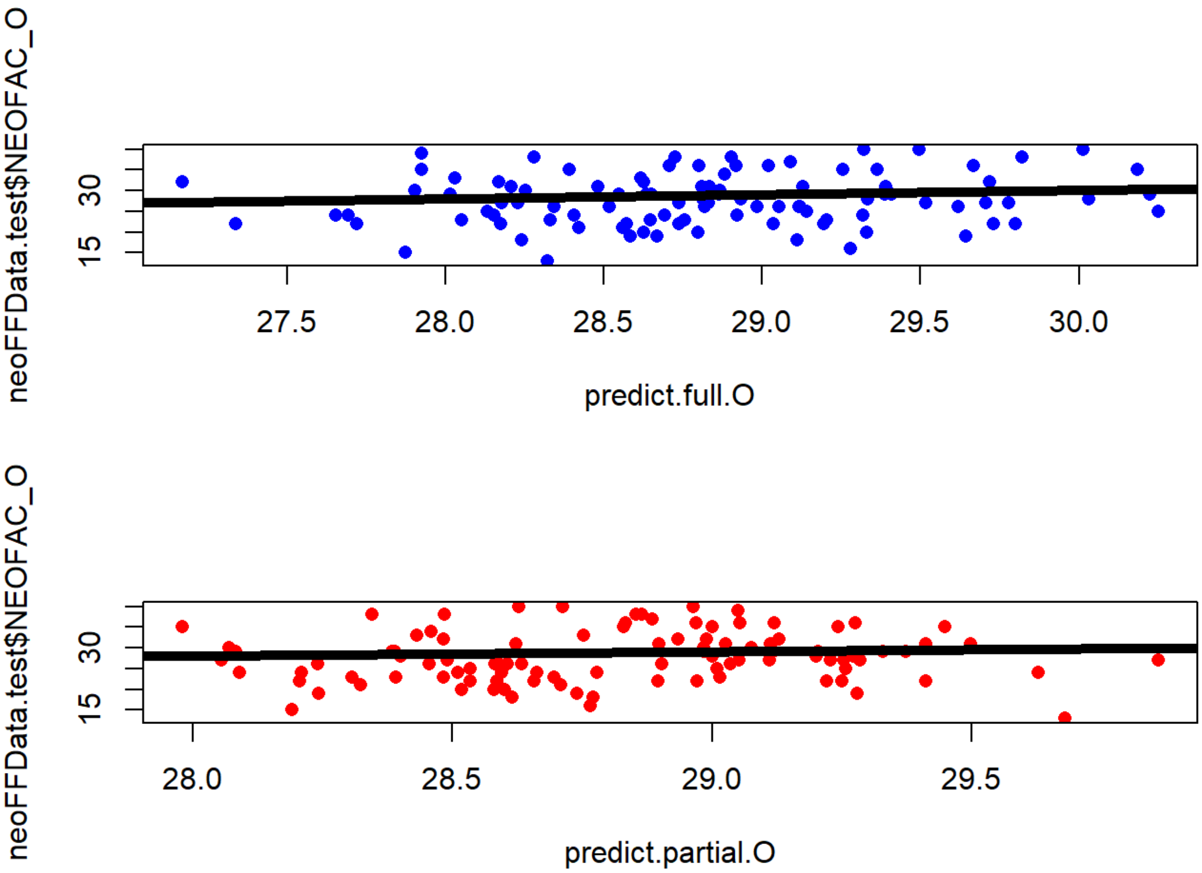
HIDE

```
par(mfrow = c(2,1))
predict.full.C = predict(fits.rf.full$NEOFAC_C, data.test$full)
predict.partial.C = predict(fits.rf.partial$NEOFAC_C, data.test$partial)
plot(predict.full.C, neoFFData.test$NEOFAC_C, pch = 16, col = 'blue')
abline(0, 1, lwd=5, col="black")
plot(predict.partial.C, neoFFData.test$NEOFAC_C, pch = 16, col = 'red')
abline(0, 1, lwd=5, col="black")
```



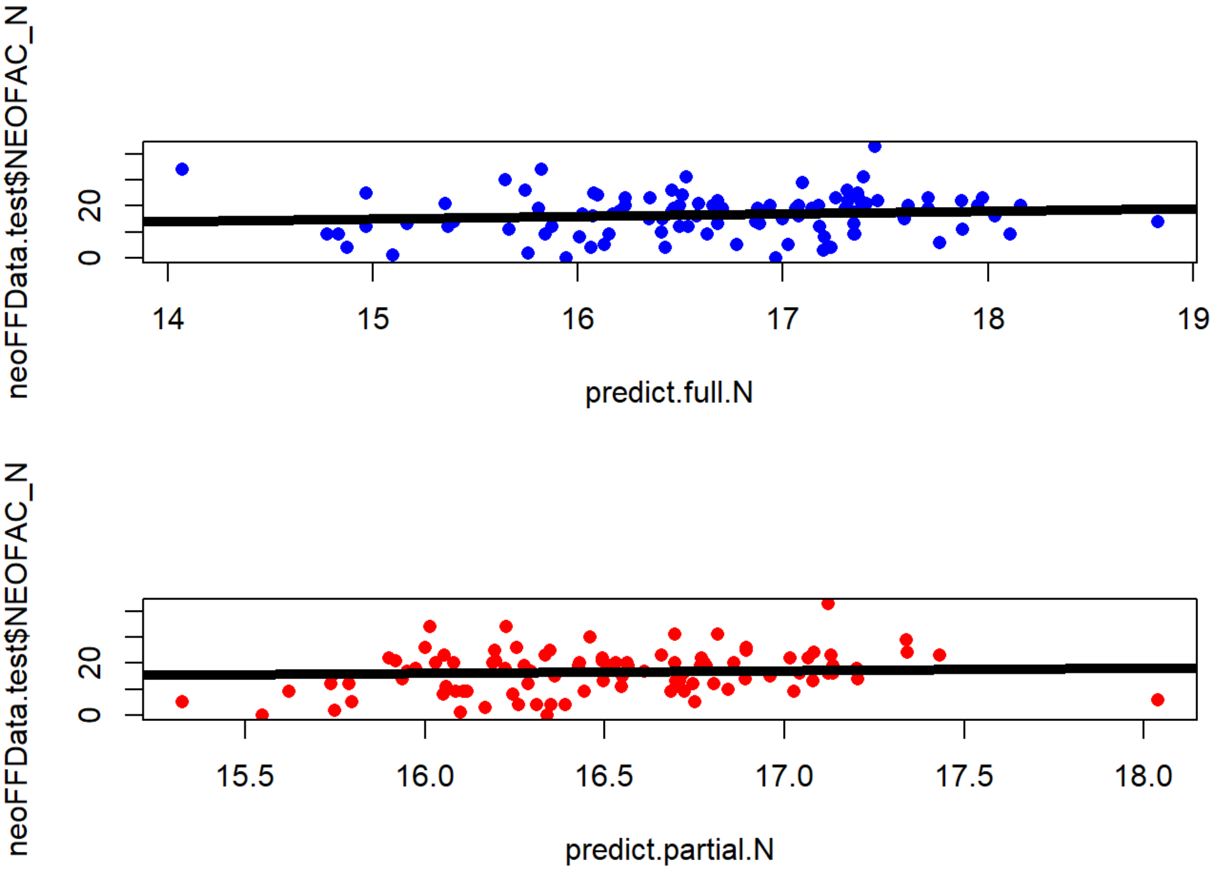
HIDE

```
par(mfrow = c(2,1))
predict.full.O = predict(fits.rf.full$NEOFAC_O, data.test$full)
predict.partial.O = predict(fits.rf.partial$NEOFAC_O, data.test$partial)
plot(predict.full.O, neoFFData.test$NEOFAC_O, pch = 16, col = 'blue')
abline(0, 1, lwd=5, col="black")
plot(predict.partial.O, neoFFData.test$NEOFAC_O, pch = 16, col = 'red')
abline(0, 1, lwd=5, col="black")
```



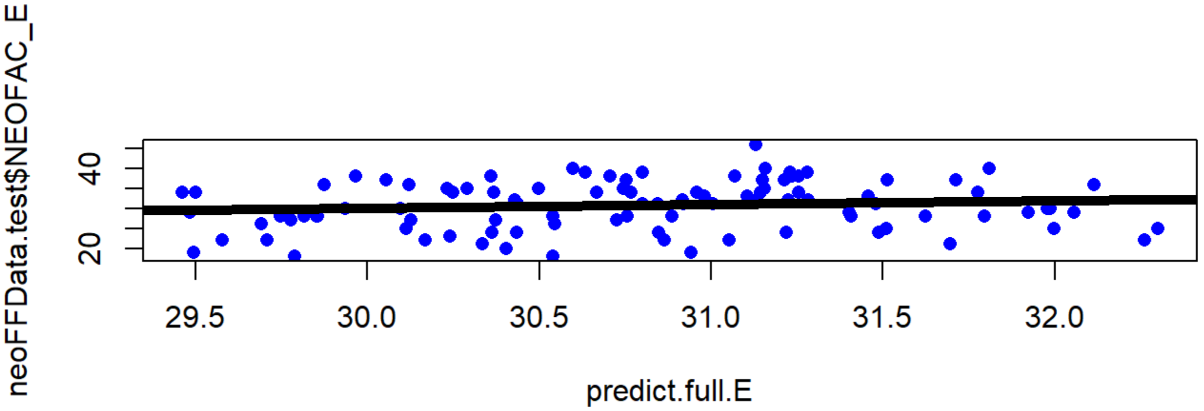
HIDE

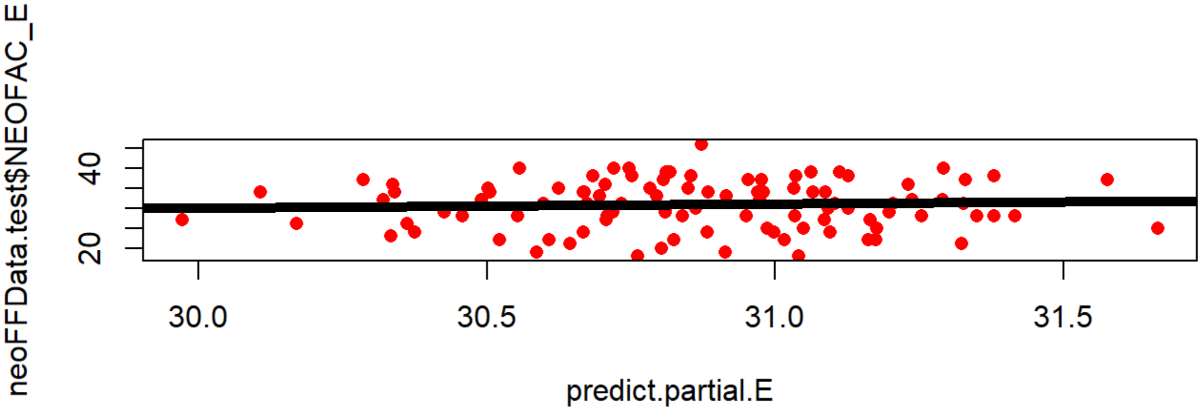
```
par(mfrow=c(2,1))
predict.full.N = predict(fits.rf.full$NEOFAC_N, data.test$full)
predict.partial.N = predict(fits.rf.partial$NEOFAC_N, data.test$partial)
plot(predict.full.N, neoFFData.test$NEOFAC_N, pch = 16, col = 'blue')
abline(0, 1, lwd=5, col="black")
plot(predict.partial.N, neoFFData.test$NEOFAC_N, pch = 16, col = 'red')
abline(0, 1, lwd=5, col="black")
```



HIDE

```
par(mfrow=c(2,1))
predict.full.E = predict(fits.rf.full$NEOFAC_E, data.test$full)
predict.partial.E = predict(fits.rf.partial$NEOFAC_E, data.test$partial)
plot(predict.full.E, neoFFData.test$NEOFAC_E, pch = 16, col = 'blue')
abline(0, 1, lwd=5, col="black")
plot(predict.partial.E, neoFFData.test$NEOFAC_E, pch = 16, col = 'red')
abline(0, 1, lwd=5, col="black")
```





Discussion

In this analysis, we show that personality data is not well predicted by this data. This is indicated by the fact that the MSE of our random forest models is similar to the MSE obtained by consistenly predicting the mean. To gain more intuition about why this might be, we fit a single tree to our training data. A single tree should overfit the data, and give a low MSE, but not be generalizeable. In our case, even the single tree gives a narrow range of predictions, although (as expected), with a lower MSE (~8) (see Appendix). Despite the lower MSE, the fact that our model still did not capture the full range of values suggests that the signal contained in our preditors might be low.

Future Steps

Boosting

Another future direction could be a different type of edge and node definition. Using ICA to define network edges inherantly destroys much of the correlated activity between regions, some of which might be noise, but some of which might be meaningful. Preprocessing step that more tailored to network analyses might give less noisy, and more interpretable results.

References

Costa PT & McCrae RR. Revised NEO Personality Inventory (NEO-PIR) and NEO Five Factor Inventory (NEO-FFI) professional manual. 1992; Odessa, FL: Psychological Assessment Resources.

Finn ES, Shen X, Scheinost D, Rosenberg MD, Huang J, Chun MM, et al. Functional connectome fingerprinting: identifying individuals using patterns of brain connectivity. Nature Neuroscience 2015; 18(11): 1664â□“1671.

Shen X, Finn ES, Scheinost D., Rosenberg MD, Chun MM, Papademetris X, & Constable RT. Using connectome-based predictive modeling to predict individual behavior from brain connectivity. Nature Protocols. 2017;12: 506â□“518.

Smith SM, Andersson J, Auerbach EJ, et al. Resting-state fMRI in the Human Connectome Project. NeuroImage. 2013;80:144-168.

Smith SM, Bandettini PA, Miller KL, Behrens TEJ, Friston KJ, David O, Liu T, Woolrich MW, Nichols TE. The danger of systematic bias in group-level FMRI-lag-based causality estimation. Neuroimage. 2012; 59:1228-1229

Smith SM, HyvÃ¤rinen A, Varoquaux G, Miller KL, Beckmann CF. Group-PCA for very large fMRI datasets. Neuroimage. 2014;101:738-749.

Trull, TJ. The Five-Factor Model of Personality Disorder and DSM-5. Journal of Personality. 2012; 80:1697â□“1720.

Van Essen DC, Smith SM, Barch DM, et al. The WU-Minn Human Connectome Project: An Overview. NeuroImage. 2013;80:62-79.

Appendix

CFA for Behavioral Data

HIDE

```
summary(fit, fit.measures=TRUE, standardized=TRUE)
```

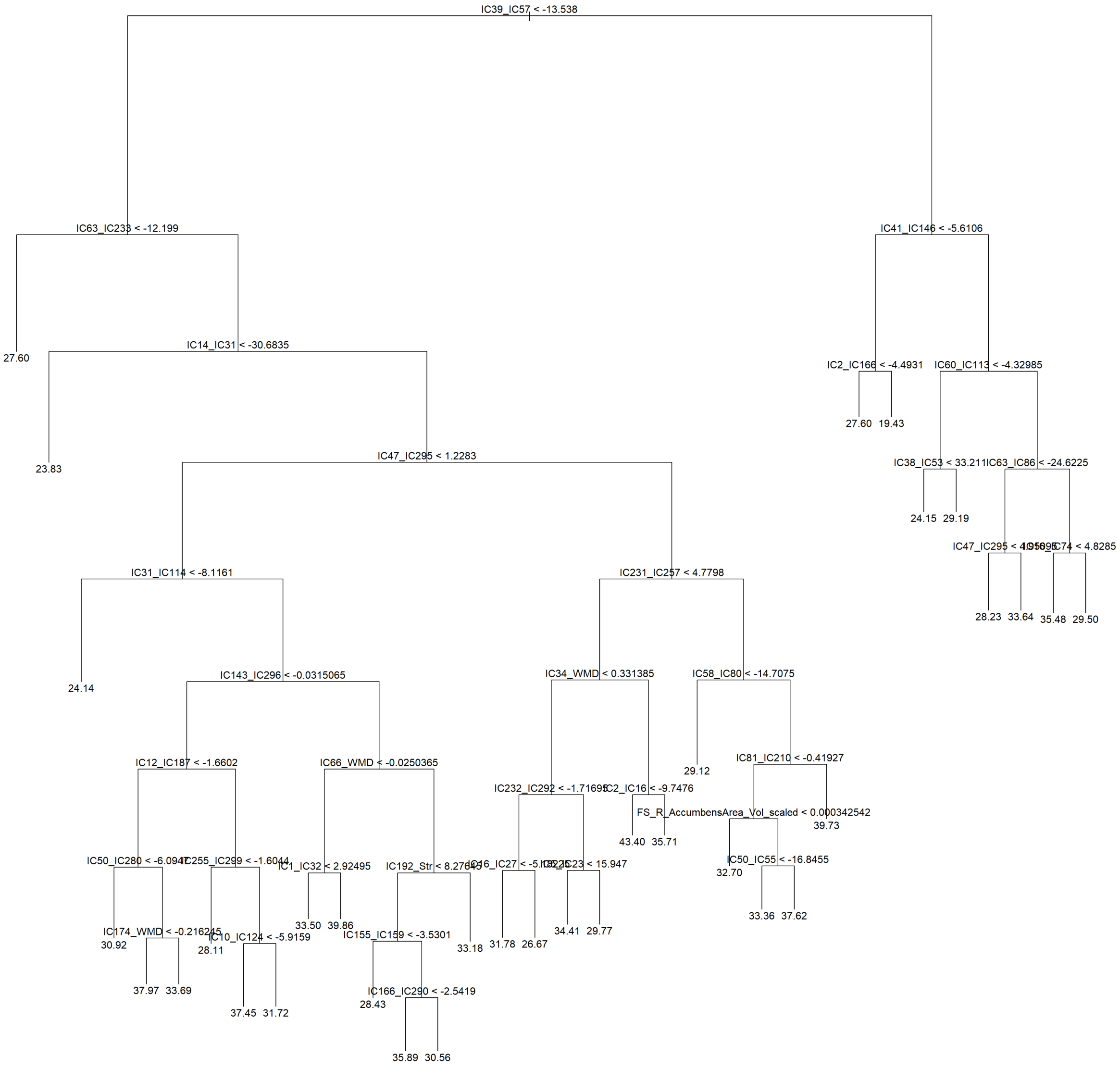
```
## lavaan (0.5-23.1097) converged normally after 123 iterations
## 
##   Number of observations                           810
## 
##   Estimator                                         ML
##   Minimum Function Test Statistic             6523.928
##   Degrees of freedom                              1700
##   P-value (Chi-square)                           0.000
## 
## Model test baseline model:
## 
##   Minimum Function Test Statistic            15675.937
##   Degrees of freedom                              1770
##   P-value                                        0.000
## 
## User model versus baseline model:
## 
##   Comparative Fit Index (CFI)                    0.653
##   Tucker-Lewis Index (TLI)                       0.639
## 
## Loglikelihood and Information Criteria:
## 
##   Loglikelihood user model (H0)             -60132.822
##   Loglikelihood unrestricted model (H1)     -56870.858
## 
##   Number of free parameters                        130
##   Akaike (AIC)                              120525.644
##   Bayesian (BIC)                            121136.259
##   Sample-size adjusted Bayesian (BIC)       120723.432
## 
## Root Mean Square Error of Approximation:
## 
##   RMSEA                                          0.059
##   90 Percent Confidence Interval          0.058  0.061
##   P-value RMSEA <= 0.05                          0.000
## 
## Standardized Root Mean Square Residual:
## 
##   SRMR                                           0.077
## 
## Parameter Estimates:
## 
##   Information                                 Expected
##   Standard Errors                             Standard
## 
## Latent Variables:
##                        Estimate  Std.Err  z-value  P(>|z|)   Std.lv
##   Neuroticism =~                                                     
##     NEORAW_01             1.000                               0.490
##     NEORAW_06            -1.083    0.108  -10.015    0.000   -0.531
##     NEORAW_11            -1.430    0.136  -10.549    0.000   -0.700
##     NEORAW_16             1.259    0.122   10.281    0.000    0.617
##     NEORAW_21            -1.201    0.113  -10.621    0.000   -0.588
##     NEORAW_26            -1.498    0.136  -10.973    0.000   -0.734
##     NEORAW_31             1.319    0.128   10.315    0.000    0.646
##     NEORAW_36            -0.847    0.095   -8.943    0.000   -0.415
##     NEORAW_41            -1.212    0.113  -10.733    0.000   -0.594
##     NEORAW_46             1.093    0.118    9.233    0.000    0.535
##     NEORAW_51            -0.967    0.093  -10.350    0.000   -0.474
##     NEORAW_56            -1.218    0.123   -9.908    0.000   -0.597
##   Extraversion =~
##     NEORAW_02             1.000                               0.460
##     NEORAW_07             0.821    0.087    9.439    0.000    0.377
##     NEORAW_12            -0.700    0.094   -7.431    0.000   -0.322
##     NEORAW_17             1.043    0.096   10.916    0.000    0.479
##     NEORAW_22             0.657    0.087    7.596    0.000    0.302
##     NEORAW_27            -0.854    0.100   -8.537    0.000   -0.393
##     NEORAW_32             0.985    0.103    9.606    0.000    0.453
##     NEORAW_37             1.305    0.110   11.879    0.000    0.600
##     NEORAW_42            -1.349    0.120  -11.240    0.000   -0.620
##     NEORAW_47             0.641    0.091    7.031    0.000    0.295
##     NEORAW_52             0.830    0.090    9.190    0.000    0.382
##     NEORAW_57            -0.880    0.103   -8.559    0.000   -0.405
##   Openness =~
##     NEORAW_03             1.000                               0.307
##     NEORAW_08             0.375    0.101    3.707    0.000    0.115
##     NEORAW_13            -2.112    0.284   -7.433    0.000   -0.649
##     NEORAW_18             0.757    0.137    5.528    0.000    0.233
##     NEORAW_23             2.337    0.317    7.365    0.000    0.718
##     NEORAW_28            -1.244    0.194   -6.400    0.000   -0.382
##     NEORAW_33             0.865    0.157    5.494    0.000    0.266
##     NEORAW_38             0.943    0.201    4.698    0.000    0.290
##     NEORAW_43            -2.555    0.343   -7.441    0.000   -0.785
##     NEORAW_48             1.878    0.260    7.235    0.000    0.577
##     NEORAW_53            -1.277    0.184   -6.951    0.000   -0.392
##     NEORAW_58            -2.117    0.289   -7.337    0.000   -0.650
##   Agreeableness =~
##     NEORAW_04             1.000                               0.262
##     NEORAW_09            -1.504    0.169   -8.927    0.000   -0.394
##     NEORAW_14            -2.380    0.235  -10.110    0.000   -0.624
##     NEORAW_19             1.034    0.163    6.337    0.000    0.271
##     NEORAW_24            -2.093    0.220   -9.533    0.000   -0.549
##     NEORAW_29            -1.799    0.216   -8.332    0.000   -0.472
##     NEORAW_34             0.975    0.113    8.657    0.000    0.256
##     NEORAW_39            -2.427    0.239  -10.168    0.000   -0.636
##     NEORAW_44            -1.773    0.210   -8.459    0.000   -0.465
##     NEORAW_49             0.935    0.108    8.628    0.000    0.245
##     NEORAW_54            -1.433    0.187   -7.676    0.000   -0.376
##     NEORAW_59            -1.874    0.212   -8.838    0.000   -0.491
##   Conscientiousness =~
##     NEORAW_05             1.000                               0.417
##     NEORAW_10             1.428    0.128   11.195    0.000    0.595
##     NEORAW_15            -0.562    0.088   -6.371    0.000   -0.234
##     NEORAW_20             0.526    0.066    7.968    0.000    0.219
##     NEORAW_25             1.200    0.110   10.948    0.000    0.500
##     NEORAW_30            -1.355    0.134  -10.082    0.000   -0.565
##     NEORAW_35             0.925    0.084   10.964    0.000    0.385
##     NEORAW_40             1.026    0.092   11.191    0.000    0.428
##     NEORAW_45            -1.709    0.150  -11.409    0.000   -0.712
##     NEORAW_50             1.025    0.087   11.819    0.000    0.427
##     NEORAW_55            -1.364    0.125  -10.957    0.000   -0.568
##     NEORAW_60             0.866    0.084   10.339    0.000    0.361
##   Std.all
##   
##     0.415
##    -0.560
##    -0.634
##     0.595
##    -0.646
##    -0.708
##     0.600
##    -0.448
##    -0.664
##     0.475
##    -0.605
##    -0.547
##   
##     0.468
##     0.461
##    -0.328
##     0.596
##     0.338
##    -0.396
##     0.474
##     0.724
##    -0.634
##     0.306
##     0.442
##    -0.398
##   
##     0.295
##     0.161
##    -0.669
##     0.289
##     0.637
##    -0.394
##     0.286
##     0.222
##    -0.673
##     0.585
##    -0.501
##    -0.625
##   
##     0.426
##    -0.476
##    -0.622
##     0.284
##    -0.543
##    -0.422
##     0.451
##    -0.631
##    -0.433
##     0.448
##    -0.371
##    -0.468
## 
##     0.458
##     0.613
##    -0.265
##     0.352
##     0.586
##    -0.503
##     0.588
##     0.613
##    -0.639
##     0.693
##    -0.587
##     0.526
## 
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Neuroticism ~~                                                        
##     Extraversion      0.095    0.014    6.600    0.000    0.421    0.421
##     Openness          0.004    0.006    0.636    0.525    0.027    0.027
##     Agreeableness     0.047    0.008    5.958    0.000    0.364    0.364
##     Conscientisnss    0.109    0.015    7.267    0.000    0.536    0.536
##   Extraversion ~~                                                       
##     Openness         -0.027    0.007   -3.683    0.000   -0.191   -0.191
##     Agreeableness     0.058    0.008    6.805    0.000    0.478    0.478
##     Conscientisnss    0.068    0.011    6.217    0.000    0.356    0.356
##   Openness ~~                                                           
##     Agreeableness    -0.011    0.004   -2.898    0.004   -0.142   -0.142
##     Conscientisnss    0.015    0.006    2.567    0.010    0.118    0.118
##   Agreeableness ~~                                                      
##     Conscientisnss    0.033    0.006    5.424    0.000    0.298    0.298
## 
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .NEORAW_01         1.152    0.059   19.523    0.000    1.152    0.828
##    .NEORAW_06         0.615    0.033   18.800    0.000    0.615    0.686
##    .NEORAW_11         0.729    0.040   18.174    0.000    0.729    0.598
##    .NEORAW_16         0.694    0.037   18.537    0.000    0.694    0.646
##    .NEORAW_21         0.484    0.027   18.052    0.000    0.484    0.583
##    .NEORAW_26         0.537    0.031   17.216    0.000    0.537    0.499
##    .NEORAW_31         0.743    0.040   18.497    0.000    0.743    0.640
##    .NEORAW_36         0.686    0.035   19.399    0.000    0.686    0.799
##    .NEORAW_41         0.447    0.025   17.837    0.000    0.447    0.559
##    .NEORAW_46         0.984    0.051   19.283    0.000    0.984    0.774
##    .NEORAW_51         0.389    0.021   18.455    0.000    0.389    0.634
##    .NEORAW_56         0.833    0.044   18.888    0.000    0.833    0.701
##    .NEORAW_02         0.754    0.040   18.883    0.000    0.754    0.781
##    .NEORAW_07         0.527    0.028   18.929    0.000    0.527    0.788
##    .NEORAW_12         0.858    0.044   19.591    0.000    0.858    0.892
##    .NEORAW_17         0.416    0.024   17.665    0.000    0.416    0.644
##    .NEORAW_22         0.709    0.036   19.556    0.000    0.709    0.886
##    .NEORAW_27         0.828    0.043   19.300    0.000    0.828    0.843
##    .NEORAW_32         0.707    0.038   18.839    0.000    0.707    0.775
##    .NEORAW_37         0.326    0.021   15.193    0.000    0.326    0.475
##    .NEORAW_42         0.572    0.033   17.124    0.000    0.572    0.598
##    .NEORAW_47         0.841    0.043   19.669    0.000    0.841    0.907
##    .NEORAW_52         0.600    0.031   19.049    0.000    0.600    0.805
##    .NEORAW_57         0.871    0.045   19.293    0.000    0.871    0.842
##    .NEORAW_03         0.992    0.050   19.691    0.000    0.992    0.913
##    .NEORAW_08         0.495    0.025   20.003    0.000    0.495    0.974
##    .NEORAW_13         0.520    0.032   16.391    0.000    0.520    0.553
##    .NEORAW_18         0.592    0.030   19.708    0.000    0.592    0.916
##    .NEORAW_23         0.756    0.045   16.985    0.000    0.756    0.595
##    .NEORAW_28         0.794    0.041   19.287    0.000    0.794    0.845
##    .NEORAW_33         0.791    0.040   19.718    0.000    0.791    0.918
##    .NEORAW_38         1.615    0.081   19.888    0.000    1.615    0.951
##    .NEORAW_43         0.744    0.046   16.304    0.000    0.744    0.547
##    .NEORAW_48         0.638    0.036   17.730    0.000    0.638    0.657
##    .NEORAW_53         0.459    0.025   18.588    0.000    0.459    0.749
##    .NEORAW_58         0.661    0.038   17.179    0.000    0.661    0.610
##    .NEORAW_04         0.310    0.016   19.009    0.000    0.310    0.818
##    .NEORAW_09         0.530    0.028   18.648    0.000    0.530    0.773
##    .NEORAW_14         0.618    0.036   16.946    0.000    0.618    0.614
##    .NEORAW_19         0.841    0.043   19.686    0.000    0.841    0.920
##    .NEORAW_24         0.721    0.040   18.022    0.000    0.721    0.706
##    .NEORAW_29         1.026    0.054   19.036    0.000    1.026    0.822
##    .NEORAW_34         0.257    0.014   18.844    0.000    0.257    0.797
##    .NEORAW_39         0.612    0.036   16.786    0.000    0.612    0.602
##    .NEORAW_44         0.937    0.049   18.966    0.000    0.937    0.813
##    .NEORAW_49         0.239    0.013   18.863    0.000    0.239    0.799
##    .NEORAW_54         0.887    0.046   19.326    0.000    0.887    0.863
##    .NEORAW_59         0.862    0.046   18.717    0.000    0.862    0.781
##    .NEORAW_05         0.655    0.034   19.236    0.000    0.655    0.790
##    .NEORAW_10         0.587    0.032   18.097    0.000    0.587    0.624
##    .NEORAW_15         0.726    0.037   19.872    0.000    0.726    0.930
##    .NEORAW_20         0.339    0.017   19.651    0.000    0.339    0.876
##    .NEORAW_25         0.478    0.026   18.368    0.000    0.478    0.657
##    .NEORAW_30         0.942    0.050   18.989    0.000    0.942    0.747
##    .NEORAW_35         0.282    0.015   18.352    0.000    0.282    0.655
##    .NEORAW_40         0.304    0.017   18.102    0.000    0.304    0.624
##    .NEORAW_45         0.736    0.041   17.805    0.000    0.736    0.592
##    .NEORAW_50         0.197    0.012   17.006    0.000    0.197    0.519
##    .NEORAW_55         0.615    0.033   18.359    0.000    0.615    0.656
##    .NEORAW_60         0.341    0.018   18.843    0.000    0.341    0.724
##     Neuroticism       0.240    0.041    5.800    0.000    1.000    1.000
##     Extraversion      0.211    0.033    6.405    0.000    1.000    1.000
##     Openness          0.094    0.024    3.877    0.000    1.000    1.000
##     Agreeableness     0.069    0.012    5.701    0.000    1.000    1.000
##     Conscientisnss    0.174    0.027    6.399    0.000    1.000    1.000
```

Single Tree

Fit and plot single tree

HIDE

```
fit1.single <- tree(NEO~., df.full$NEOFAC_A) # The order plays no role

plot(fit1.single)
text(fit1.single, pretty=0)
```



HIDE

```
# pretty=0 only affect the categorical var's. The names will be shown.
# It has 6 terminal nodes. That means we partition CAtBat and Chits into six boxes.
# The predicted values are the sample means in each box.
```

Look at results

HIDE