ROI Analysis

ROI analysis

Setup

```r
library(easypackages)
libraries("ggplot2","patchwork","here")
source(here("code","cohens_d.R"))
options(stringsAsFactors=FALSE)
```

Read in data

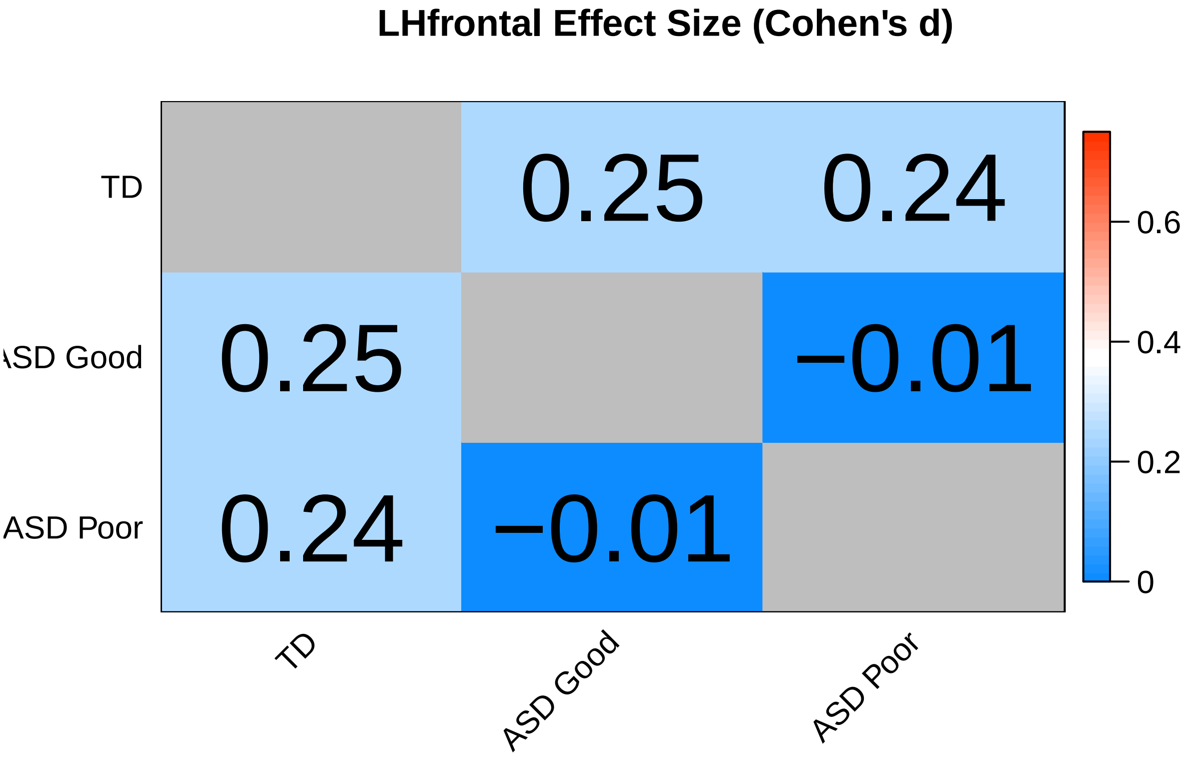
```r
# read in ROI data
fname = file.path(here("data","tidy","tidy_roidata.csv"))
data = read.csv(fname)
```

Run analyses on each ROI

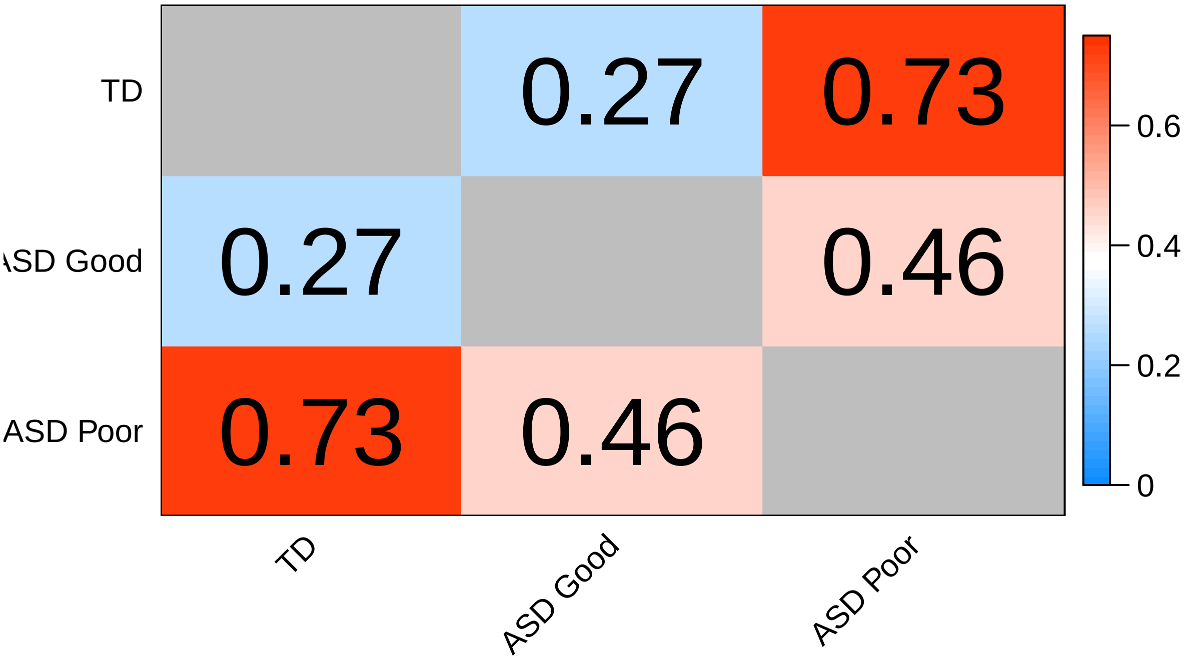
```r
# names of ROIs
roiname = c("LHfrontal","LHtemporal","RHfrontal","RHtemporal")

# pre-allocate data frame for storing subtype model results
res_colnames = c("fstat","pval","fdr",
  "d_TD_vs_ASDGood","t_TD_vs_ASDGood","p_TD_vs_ASDGood",
  "d_TD_vs_ASDPoor","t_TD_vs_ASDPoor","p_TD_vs_ASDPoor",
  "d_ASDGood_vs_ASDPoor","t_ASDGood_vs_ASDPoor","p_ASDGood_vs_ASDPoor")
res = data.frame(matrix(nrow = length(roiname), ncol = length(res_colnames)))
colnames(res) = res_colnames
rownames(res) = roiname

# pre-allocate data frame for storing case-control model results
res_colnames = c("fstat","pval","fdr",
                 "d_TD_vs_ASD","t_TD_vs_ASD","p_TD_vs_ASD")
res_casecontrol = data.frame(matrix(nrow = length(roiname),
                                    ncol = length(res_colnames)))
colnames(res_casecontrol) = res_colnames
rownames(res_casecontrol) = roiname

for (i in 1:length(roiname)){
  # run subtype model
  form2use = as.formula(sprintf("%s ~ subgrp + sex",roiname[i]))
  mod2use = lm(formula = form2use, data = data)
  tmp_res = anova(mod2use)
  res$fstat[i] = tmp_res[1,4]
  res$pval[i] = tmp_res[1,5]

  # run case-control model
```

```
form2use = as.formula(sprintf("%s ~ Dx + sex",roiname[i]))
mod2use = lm(formula = form2use, data = data)
tmp_res = anova(mod2use)
res_casecontrol$fstat[i] = tmp_res[1,4]
res_casecontrol$pval[i] = tmp_res[1,5]

# remove sex variation from data to compute effect sizes
form2use = as.formula(sprintf("%s ~ subgrp + sex",roiname[i]))
residmod = lm(formula = form2use, data = data)
full_model = model.matrix(~0+as.factor(subgrp) + as.factor(sex), data=data)
colnames(full_model) = c("Good","Poor","TD","sex")
cov_columns = c("sexM")
beta1 = residmod$coefficients[cov_columns, drop = FALSE]
beta1[is.na(beta1)] = 0
cov_columns = c("sex")
reduced_model = full_model[,cov_columns]
data$resid2use = t(data[,roiname[i]] - beta1 %*% t(reduced_model))

# case-control t-stats and effect sizes
mask1 = data$subgrp=="TD"
mask2 = data$subgrp=="Good" | data$subgrp=="Poor"
res_casecontrol$d_TD_vs_ASD[i] = cohens_d(data$resid2use[mask1],
                                          data$resid2use[mask2])
res_casecontrol$t_TD_vs_ASD[i] = t.test(data$resid2use[mask1],
                                        data$resid2use[mask2])$statistic
res_casecontrol$p_TD_vs_ASD[i] = t.test(data$resid2use[mask1],
                                        data$resid2use[mask2])$p.value

# subtype t-tstats and effect sizes
res$d_TD_vs_ASDGood[i] = cohens_d(data$resid2use[data$subgrp=="TD"],
                                  data$resid2use[data$subgrp=="Good"])
res$d_TD_vs_ASDPoor[i] = cohens_d(data$resid2use[data$subgrp=="TD"],
                                  data$resid2use[data$subgrp=="Poor"])
res$d_ASDGood_vs_ASDPoor[i] = cohens_d(data$resid2use[data$subgrp=="Good"],
                                       data$resid2use[data$subgrp=="Poor"])

res$t_TD_vs_ASDGood[i] = t.test(data$resid2use[data$subgrp=="TD"],
                                data$resid2use[data$subgrp=="Good"])$statistic
res$t_TD_vs_ASDPoor[i] = t.test(data$resid2use[data$subgrp=="TD"],
                                data$resid2use[data$subgrp=="Poor"])$statistic
res$t_ASDGood_vs_ASDPoor[i] = t.test(data$resid2use[data$subgrp=="Good"],
                                     data$resid2use[data$subgrp=="Poor"])$statistic

res$p_TD_vs_ASDGood[i] = t.test(data$resid2use[data$subgrp=="TD"],
                                data$resid2use[data$subgrp=="Good"])$p.value
res$p_TD_vs_ASDPoor[i] = t.test(data$resid2use[data$subgrp=="TD"],
                                data$resid2use[data$subgrp=="Poor"])$p.value
res$p_ASDGood_vs_ASDPoor[i] = t.test(data$resid2use[data$subgrp=="Good"],
                                     data$resid2use[data$subgrp=="Poor"])$p.value

# fill in effect size matrix
names2use = c("TD","ASD Good","ASD Poor")
es_mat = matrix(nrow = length(names2use), ncol = length(names2use))
```

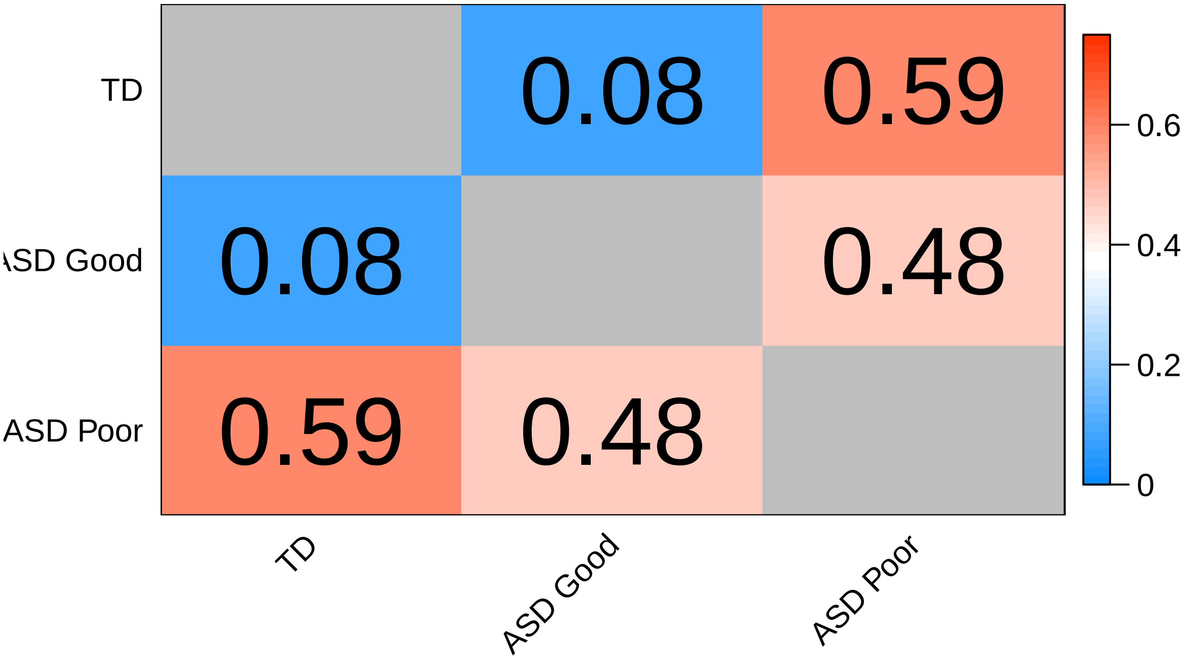
```
  colnames(es_mat) = names2use
  rownames(es_mat) = colnames(es_mat)
  es_mat[1,1] = NA
  es_mat[2,2] = NA
  es_mat[3,3] = NA

  es_mat[1,2] = res$d_TD_vs_ASDGood[i]
  es_mat[2,1] = res$d_TD_vs_ASDGood[i]
  es_mat[1,3] = res$d_TD_vs_ASDPoor[i]
  es_mat[3,1] = res$d_TD_vs_ASDPoor[i]
  es_mat[2,3] = res$d_ASDGood_vs_ASDPoor[i]
  es_mat[3,2] = res$d_ASDGood_vs_ASDPoor[i]

  #plot the effect size matrix as a heatmap
    WGCNA::labeledHeatmap(Matrix = es_mat,
        xLabels = rownames(es_mat), yLabels = colnames(es_mat),
        ySymbols = NULL, colorLabels = FALSE,
        colors = WGCNA::blueWhiteRed(50), textMatrix = round(es_mat,digits=2),
        setStdMargins = FALSE, cex.text = 3, zlim = c(0,0.75),
        main = sprintf("%s Effect Size (Cohen's d)",roiname[i]))
} # for (i in 1:length(roiname))
```

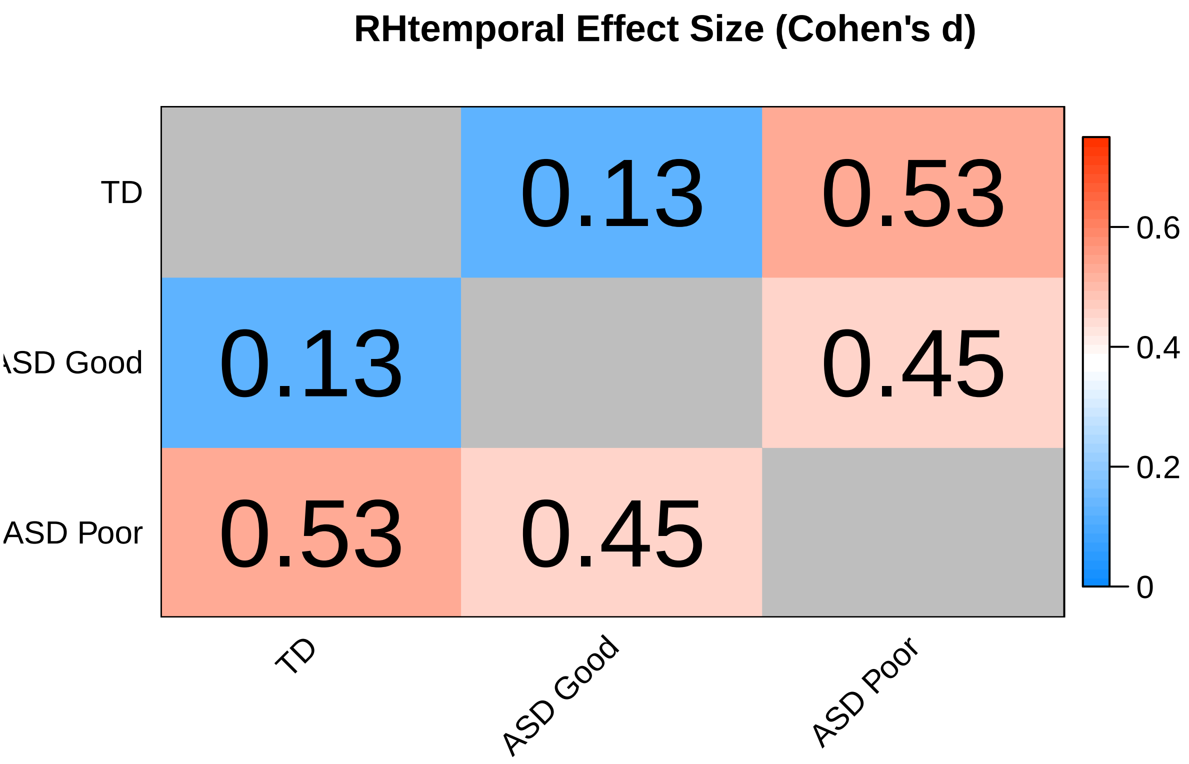


LHtemporal Effect Size (Cohen's d)



RHfrontal Effect Size (Cohen's d)





```r
# compute FDR
res_casecontrol$fdr = p.adjust(res_casecontrol$pval, method = "fdr")
res$fdr = p.adjust(res$pval, method = "fdr")
```

Results from subtype model

```r
knitr::kable(res[,c("fstat","pval","fdr")], digits = 4)
```

	fstat	pval	fdr
LHfrontal	0.7500	0.4747	0.4747
LHtemporal	6.1688	0.0029	0.0114
RHfrontal	3.6576	0.0289	0.0385
RHtemporal	4.5552	0.0125	0.0250

Results from case-control model

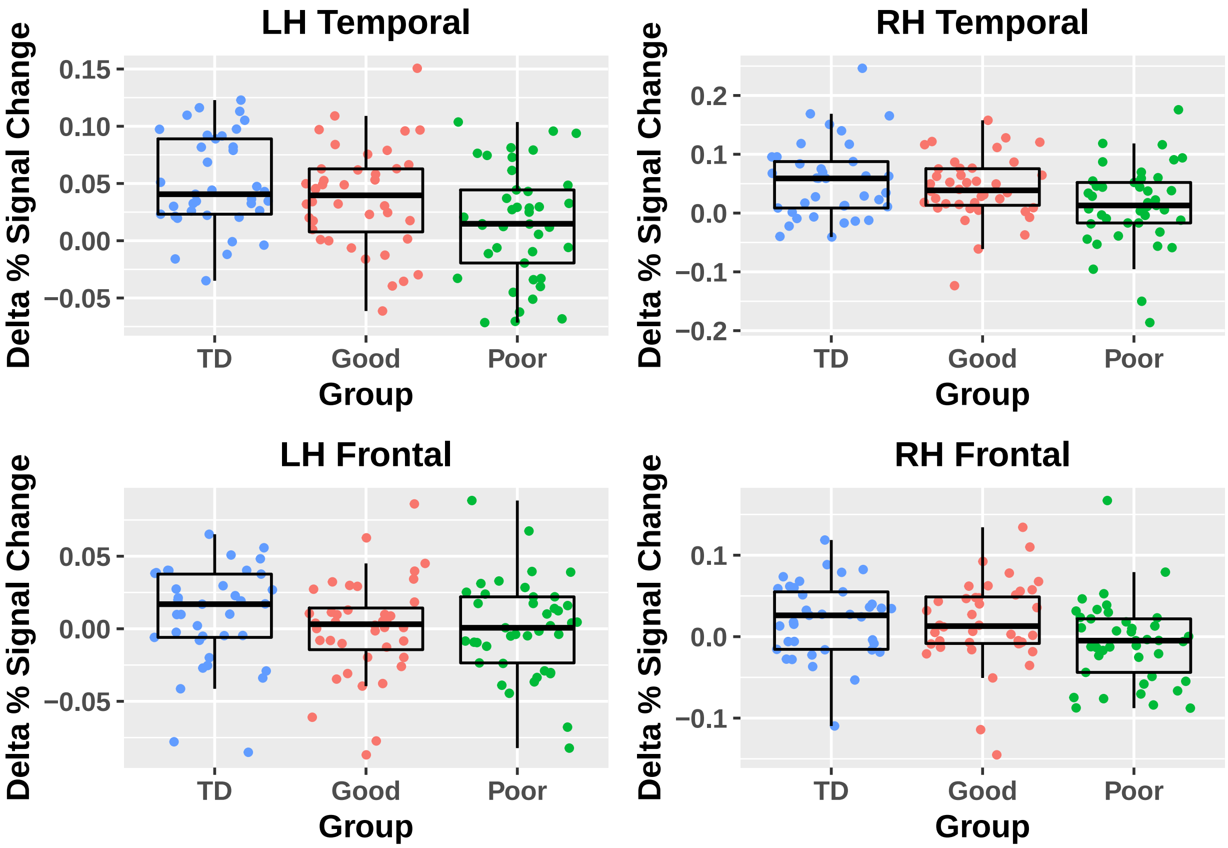
```r
knitr::kable(res_casecontrol, digits = 4)
```

	fstat	pval	fdr	d_TD_vs_ASD	t_TD_vs_ASD	p_TD_vs_ASD
LHfrontal	1.5121	0.2213	0.2213	0.2475	1.2306	0.2227
LHtemporal	7.2353	0.0082	0.0328	0.4887	2.5954	0.0112
RHfrontal	2.4242	0.1222	0.1630	0.3202	1.7009	0.0929
RHtemporal	4.5727	0.0346	0.0692	0.3419	1.6850	0.0967

Make plots

```
dotSize = 1
yLabel = "Delta % Signal Change"
xLabel = "Group"
black.bold.italic.text = element_text(face = "bold",
                                      colour = "black")
black.bold.axis.title = element_text(face="bold",
                                     size = 12,
                                     colour = "black")
bold.axis.text = element_text(face="bold",
                              size = 10)

data$plotOrder = factor(data$plotOrder)

p1 = ggplot(data = data, aes(x = reorder(subgrp, as.numeric(plotOrder)),
                             y = LHtemporal, colour = subgrp))
p1 = p1 + geom_jitter(size = dotSize) +
  geom_boxplot(fill = NA, colour = "#000000", outlier.shape = NA) +
  guides(colour = FALSE)
p1 = p1 + xlab(xLabel) + ylab(yLabel) + labs(title = "LH Temporal") +
  theme(plot.title = element_text(hjust = 0.5))
p1 = p1 + theme(title = black.bold.italic.text) +
  theme(axis.title = black.bold.axis.title) +
  theme(axis.text.x = bold.axis.text) +
  theme(axis.text.y = bold.axis.text)

p2 = ggplot(data = data, aes(x = reorder(subgrp, as.numeric(plotOrder)),
                             y = RHtemporal, colour = subgrp))
p2 = p2 + geom_jitter(size = dotSize) +
  geom_boxplot(fill = NA, colour = "#000000", outlier.shape = NA) +
  guides(colour = FALSE)
p2 = p2 + xlab(xLabel) + ylab(yLabel) + labs(title = "RH Temporal") +
  theme(plot.title = element_text(hjust = 0.5))
p2 = p2 + theme(title = black.bold.italic.text) +
  theme(axis.title = black.bold.axis.title) +
  theme(axis.text.x = bold.axis.text) +
  theme(axis.text.y = bold.axis.text)

p3 = ggplot(data = data, aes(x = reorder(subgrp, as.numeric(plotOrder)),
                             y = LHfrontal, colour = subgrp))
p3 = p3 + geom_jitter(size = dotSize) +
  geom_boxplot(fill = NA, colour = "#000000", outlier.shape = NA) +
  guides(colour = FALSE)
p3 = p3 + xlab(xLabel) + ylab(yLabel) + labs(title = "LH Frontal") +
  theme(plot.title = element_text(hjust = 0.5))
p3 = p3 + theme(title = black.bold.italic.text) +
  theme(axis.title = black.bold.axis.title) +
  theme(axis.text.x = bold.axis.text) +
  theme(axis.text.y = bold.axis.text)

p4 = ggplot(data = data, aes(x = reorder(subgrp, as.numeric(plotOrder)),
                             y = RHfrontal, colour = subgrp))
```

```
p4 = p4 + geom_jitter(size = dotSize) +
  geom_boxplot(fill = NA, colour = "#000000", outlier.shape = NA) +
  guides(colour = FALSE)
p4 = p4 + xlab(xLabel) + ylab(yLabel) + labs(title = "RH Frontal") +
  theme(plot.title = element_text(hjust = 0.5))
p4 = p4 + theme(title = black.bold.italic.text) +
  theme(axis.title = black.bold.axis.title) +
  theme(axis.text.x = bold.axis.text) +
  theme(axis.text.y = bold.axis.text)

p_final = p1 + p2 + p3 + p4 + plot_layout(nrow = 2, ncol = 2)
p_final
```


LH Temporal
Delta % Signal Change
TD
Good
Poor
Group
RH Temporal
Delta % Signal Change
TD
Good
Poor
Group
LH Frontal
Delta % Signal Change
TD
Good
Poor
Group
RH Frontal
Delta % Signal Change
TD
Good
Poor
Group