Reproducible Analysis

Fully reproducible analysis stream

Lombardo et al., eLife

```
# Load necessary libraries
library(easypackages)
libraries("nlme","matlabr","ggplot2","multcomp","readxl",
          "heplots","effsize","reshape2","plyr","psych",
          "MASS","here","rsample")

# paths
codedir = here("code")
tidydatadir = here("data","tidy")
resultdir = here("results")
plotdir = here("plots")

# allows for selecting n colors from color wheel like ggplot2
source(file.path(codedir,"get_ggColorHue.R"))

# for running clinical trajectory analyses
source(file.path(codedir,"functions4trajanalysis.R"))

# Set options and other stuff
options(stringsAsFactors = FALSE)
options(tibble.print_max = Inf)
options(matlab.path = "/Applications/MATLAB_R2018b.app/bin")

fontSize = 10
dotSize = 3
fdr_thresh = 0.05
RUNMATLAB = FALSE
```

Read in pheno data

```
Dfmri = read_excel(file.path(tidydatadir,"final_allETrsfMRIsubs_phenodata04_ASDTDLDDDSIB.xlsx"))
Dfmri$subjectId = factor(Dfmri$subjectId)
Dfmri$subgrp2 = factor(Dfmri$subgrp2)
Dfmri$subgrp2 = factor(Dfmri$subgrp2,levels(Dfmri$subgrp2)[c(2,4,3,6,5,1)])

colours2use = get_ggColorHue(6)
```

Analyze demographics

```
demoVar_sub = Dfmri[,c("subgrp2","scan_age","sex",
                       "ET1_Age","meanFD",
                       "meanDVARSraw","meanDVARSwavelet")]
describeBy(demoVar_sub, group = "subgrp2")
```

```
##
##  Descriptive statistics by group
## subgrp2: GeoASD
##                  vars  n  mean   sd median trimmed   mad   min   max range
## subgrp2*            1 16  1.00 0.00   1.00    1.00  0.00  1.00  1.00  0.00
## scan_age            2 16 29.93 8.72  28.04   30.06 11.50 14.16 43.79 29.63
## sex*                3 16   NaN   NA     NA     NaN    NA   Inf  -Inf  -Inf
## ET1_Age             4 16 28.38 7.78  27.00   28.43  7.41 15.00 41.00 26.00
## meanFD              5 16  0.06 0.02   0.06    0.06  0.02  0.03  0.11  0.08
## meanDVARSraw        6 16  6.87 1.14   6.99    6.94  1.26  4.45  8.38  3.93
## meanDVARSwavelet    7 16  5.31 0.87   5.33    5.32  0.79  3.79  6.72  2.93
##                   skew kurtosis   se
## subgrp2*           NaN      NaN 0.00
## scan_age          0.03    -1.20 2.18
## sex*                NA       NA   NA
## ET1_Age           0.07    -1.12 1.94
## meanFD            0.60    -1.14 0.01
## meanDVARSraw     -0.55    -0.85 0.28
## meanDVARSwavelet -0.04    -1.15 0.22
## --------------------------------------------------------
## subgrp2: nonGeoASD
##                  vars  n  mean   sd median trimmed  mad   min   max range
## subgrp2*            1 62  2.00 0.00   2.00    2.00 0.00  2.00  2.00  0.00
## scan_age            2 62 29.37 8.35  30.47   29.65 9.04 12.35 44.06 31.70
## sex*                3 62   NaN   NA     NA     NaN   NA   Inf  -Inf  -Inf
## ET1_Age             4 62 26.81 8.35  26.00   26.60 7.41 12.00 44.00 32.00
## meanFD              5 62  0.10 0.12   0.06    0.07 0.02  0.03  0.93  0.90
## meanDVARSraw        6 62  7.00 1.61   6.70    6.86 1.00  4.14 15.17 11.03
## meanDVARSwavelet    7 62  5.34 0.79   5.27    5.31 0.75  3.65  7.30  3.65
##                   skew kurtosis   se
## subgrp2*           NaN      NaN 0.00
## scan_age         -0.35    -0.84 1.06
## sex*                NA       NA   NA
## ET1_Age           0.24    -0.74 1.06
## meanFD            5.32    32.56 0.02
## meanDVARSraw      2.11     8.55 0.20
## meanDVARSwavelet  0.32     0.00 0.10
## --------------------------------------------------------
## subgrp2: LD_DD
##                  vars  n  mean   sd median trimmed  mad   min   max range
## subgrp2*            1 15  3.00 0.00   3.00    3.00 0.00  3.00  3.00  0.00
## scan_age            2 15 25.12 7.97  23.95   24.90 9.35 13.37 39.75 26.38
## sex*                3 15   NaN   NA     NA     NaN   NA   Inf  -Inf  -Inf
## ET1_Age             4 11 19.36 4.15  20.00   19.44 4.45 13.00 25.00 12.00
## meanFD              5 15  0.10 0.05   0.08    0.09 0.04  0.03  0.19  0.15
## meanDVARSraw        6 15  7.41 2.14   6.82    7.08 1.10  5.16 14.00  8.84
## meanDVARSwavelet    7 15  5.55 1.22   5.39    5.37 0.78  4.11  9.33  5.21
##                   skew kurtosis   se
## subgrp2*           NaN      NaN 0.00
## scan_age          0.28    -1.06 2.06
## sex*                NA       NA   NA
## ET1_Age          -0.26    -1.44 1.25
## meanFD            0.60    -1.17 0.01
## meanDVARSraw      1.88     3.22 0.55
```

```
## meanDVARSwavelet  1.80     3.28 0.32
## --------------------------------------------------------
## subgrp2: TypSibASD
##                  vars  n  mean   sd median trimmed  mad   min   max range
## subgrp2*            1 16  4.00 0.00   4.00    4.00 0.00  4.00  4.00  0.00
## scan_age            2 16 26.74 9.38  27.89   26.51 9.74 12.52 44.09 31.57
## sex*                3 16   NaN   NA     NA     NaN   NA   Inf  -Inf  -Inf
## ET1_Age             4 14 19.79 6.20  19.50   19.42 8.15 13.00 31.00 18.00
## meanFD              5 16  0.08 0.04   0.07    0.08 0.03  0.03  0.18  0.15
## meanDVARSraw        6 16  6.82 1.57   6.47    6.69 0.82  4.64 10.77  6.13
## meanDVARSwavelet    7 16  5.28 0.88   5.14    5.22 0.49  3.91  7.44  3.53
##                  skew kurtosis   se
## subgrp2*          NaN      NaN 0.00
## scan_age         0.02    -1.01 2.35
## sex*               NA       NA   NA
## ET1_Age          0.41    -1.31 1.66
## meanFD           0.75    -0.63 0.01
## meanDVARSraw     0.99     0.31 0.39
## meanDVARSwavelet 0.90     0.39 0.22
## --------------------------------------------------------
## subgrp2: TD
##                  vars  n  mean    sd median trimmed   mad   min   max
## subgrp2*            1 55  5.00  0.00   5.00    5.00  0.00  5.00  5.00
## scan_age            2 55 29.61 10.14  30.92   29.70 11.69 13.17 47.93
## sex*                3 55   NaN    NA     NA     NaN    NA   Inf  -Inf
## ET1_Age             4 55 23.07  9.07  22.00   22.29 11.86 12.00 45.00
## meanFD              5 55  0.11  0.11   0.07    0.08  0.03  0.04  0.59
## meanDVARSraw        6 55  7.00  1.44   6.85    6.87  1.52  4.63 11.32
## meanDVARSwavelet    7 55  5.19  0.61   5.11    5.17  0.66  3.89  6.58
##                  range  skew kurtosis   se
## subgrp2*          0.00   NaN      NaN 0.00
## scan_age         34.76 -0.21    -1.12 1.37
## sex*              -Inf    NA       NA   NA
## ET1_Age          33.00  0.57    -0.68 1.22
## meanFD            0.55  2.66     7.20 0.01
## meanDVARSraw      6.69  0.79     0.34 0.19
## meanDVARSwavelet  2.69  0.25    -0.48 0.08
## --------------------------------------------------------
## subgrp2: ASDnoET
##                  vars  n  mean   sd median trimmed   mad   min   max range
## subgrp2*            1 31  6.00 0.00   6.00    6.00  0.00  6.00  6.00  0.00
## scan_age            2 31 29.69 8.88  30.03   29.81 11.50 13.21 43.63 30.42
## sex*                3 31   NaN   NA     NA     NaN    NA   Inf  -Inf  -Inf
## ET1_Age             4  0   NaN   NA     NA     NaN    NA   Inf  -Inf  -Inf
## meanFD              5 31  0.09 0.08   0.06    0.07  0.02  0.03  0.41  0.37
## meanDVARSraw        6 31  6.98 1.46   6.91    6.93  1.05  4.00 11.63  7.63
## meanDVARSwavelet    7 31  5.24 0.76   5.11    5.26  0.71  3.57  6.56  2.99
##                   skew kurtosis   se
## subgrp2*           NaN      NaN 0.00
## scan_age         -0.05    -1.22 1.59
## sex*                NA       NA   NA
## ET1_Age             NA       NA   NA
## meanFD            2.42     5.45 0.01
## meanDVARSraw      0.73     1.74 0.26
```

```
## meanDVARSwavelet -0.11    -0.58 0.14
```

Scan Age ANOVA

```r
mod2use = lm(scan_age ~ subgrp2, data = demoVar_sub)
anova(mod2use)
```

```
## Analysis of Variance Table
##
## Response: scan_age
##            Df Sum Sq Mean Sq F value Pr(>F)
## subgrp2     5    366  73.208  0.8914 0.4879
## Residuals 189  15522  82.127
```

Eye tracking age ANOVA

```r
mod2use = lm(ET1_Age ~ subgrp2, data = demoVar_sub)
anova(mod2use)
```

```
## Analysis of Variance Table
##
## Response: ET1_Age
##            Df  Sum Sq Mean Sq F value   Pr(>F)
## subgrp2     4  1283.4  320.84  4.7742 0.001174 **
## Residuals 153 10282.0   67.20
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Sex by Group Table for all individuals

```r
subgrp_sex_tab = table(demoVar_sub$subgrp2, demoVar_sub$sex, exclude = "NA")
subgrp_sex_chisq_res = chisq.test(subgrp_sex_tab)
subgrp_sex_tab
```

```
##
##               F  M
##   GeoASD      5 11
##   nonGeoASD  13 49
##   LD_DD       5 10
##   TypSibASD   8  8
##   TD         18 37
##   ASDnoET     4 27
```

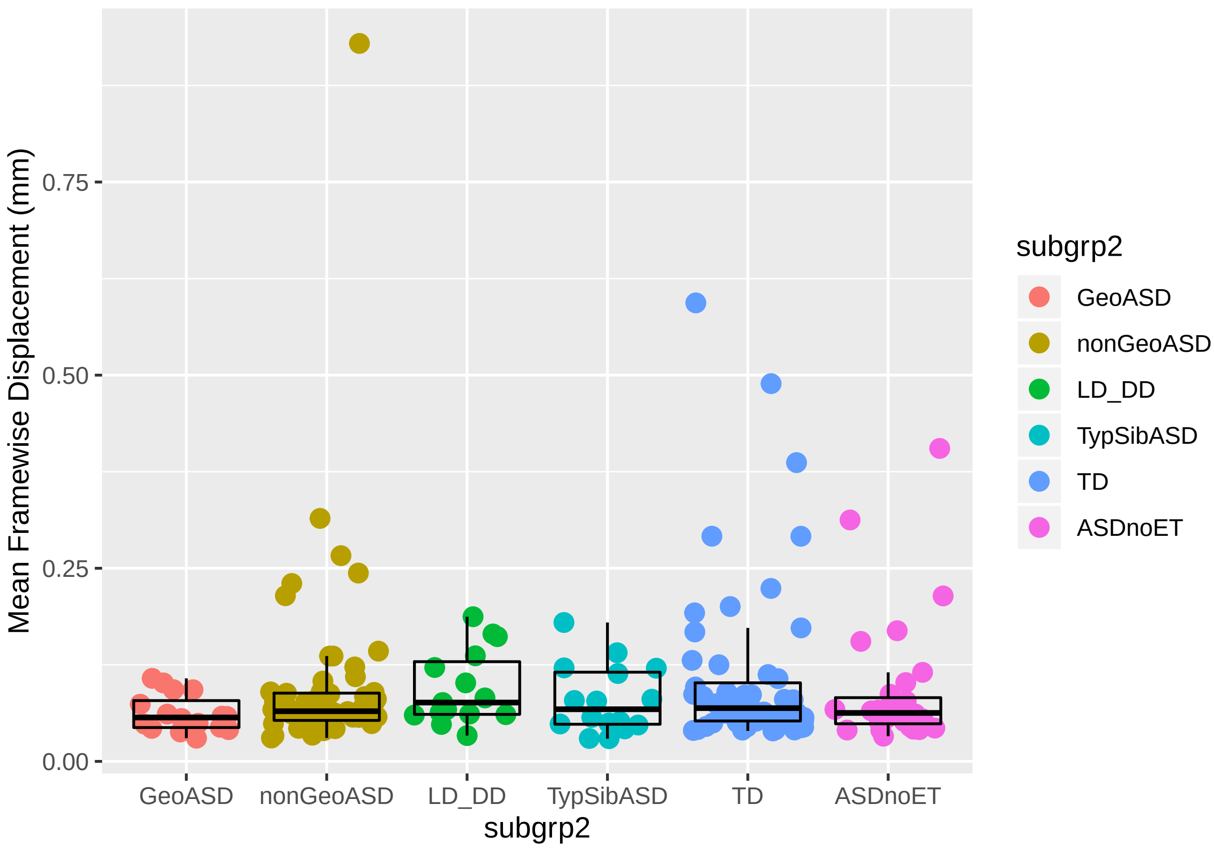
```r
subgrp_sex_chisq_res
```

```
##
##  Pearson's Chi-squared test
##
## data:  subgrp_sex_tab
## X-squared = 9.8871, df = 5, p-value = 0.0785
```

Analyze Head Motion Measures for Group Differences and make plots

Will look at mean framewise displacement (mean FD in mm), as well as DVARS measurements before and after wavelet denoising, just to show the impact wavelet denoising has on removing substantial amounts of artifact from the data.

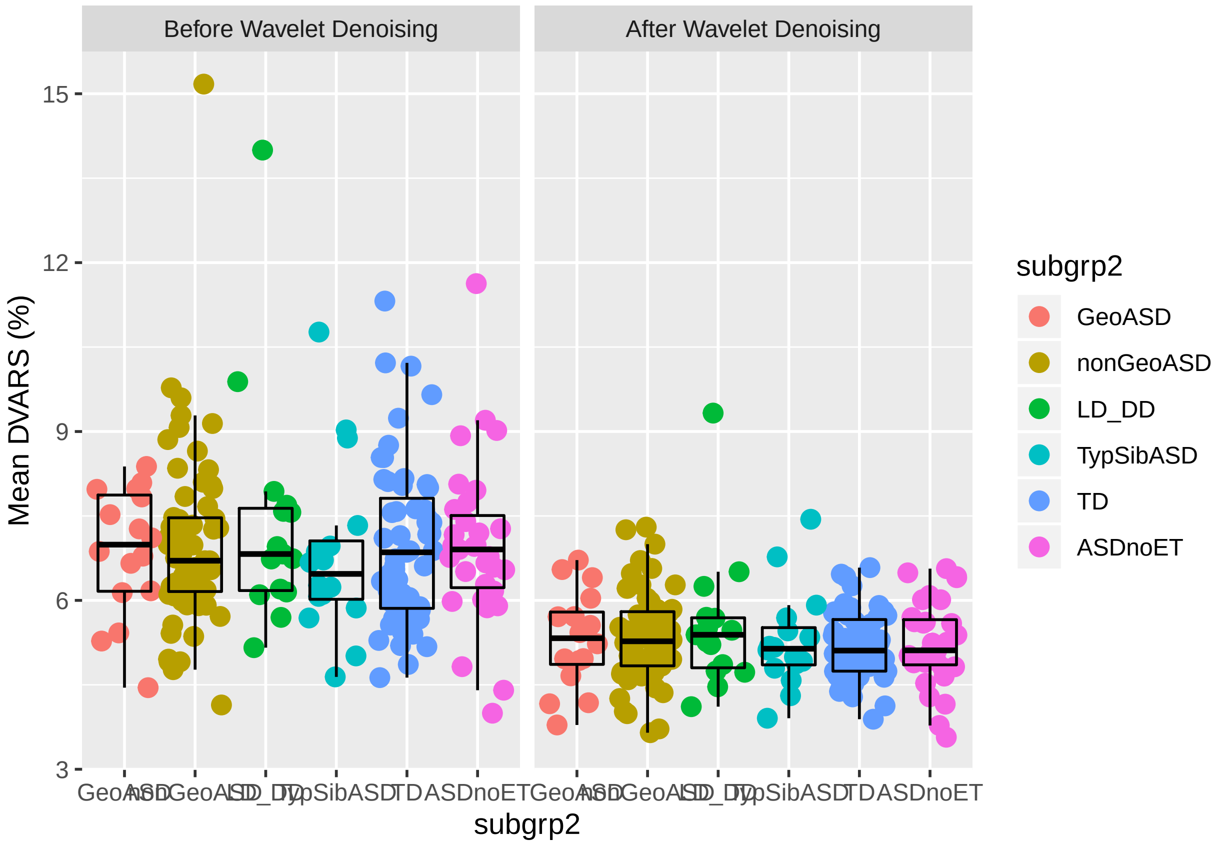
```
# Mean Framewise Displacement Plot
p = ggplot(data = Dfmri, aes(x = subgrp2, y = meanFD, colour = subgrp2))
p = p + geom_jitter(size = dotSize) +
        geom_boxplot(fill = NA, colour = "#000000", outlier.shape = NA)
p = p + scale_colour_manual(values = colours2use) + ylab("Mean Framewise Displacement (mm)")
ggsave(filename = file.path(plotdir,"meanFDplot.pdf"))
p
```



Make Mean DVARS Plot that shows DVARS before and after wavelet denoising

```
Dsub = melt(as.data.frame(Dfmri), id = c("subjectId","subgrp2"),
    measure = c("meanDVARSraw","meanDVARSwavelet"))
Dsub$value = as.numeric(Dsub$value)
Dsub$variable = factor(Dsub$variable)
Dsub$variable = revalue(Dsub$variable, c("meanDVARSraw"="Before Wavelet Denoising",
                                         "meanDVARSwavelet"="After Wavelet Denoising"))

p = ggplot(data = Dsub, aes(x = subgrp2, y = value, colour = subgrp2)) + facet_grid(. ~ variable)
p = p + geom_jitter(size = dotSize)+
        geom_boxplot(fill = NA, colour = "#000000", outlier.shape = NA)
```

```r
p = p + scale_colour_manual(values = colours2use) + ylab("Mean DVARS (%)")
ggsave(filename = file.path(plotdir,"meanDVARSplot.pdf"))
p
```



Run ANOVAs on measures of head motion (meanFD)

```r
# Mean FD
lm_formula = as.formula(sprintf("%s ~ %s","meanFD","subgrp2"))
mod2use = eval(substitute(lm(formula = lm_formula, data = Dfmri, na.action = na.omit)))
anova(mod2use)
```

```
## Analysis of Variance Table
##
## Response: meanFD
##            Df  Sum Sq   Mean Sq F value Pr(>F)
## subgrp2     5 0.03298 0.0065955  0.6759 0.6422
## Residuals 189 1.84420 0.0097577
```

Mean DVARS before and after wavelet denoising

```r
# Before
lm_formula = as.formula(sprintf("%s ~ %s","meanDVARSraw","subgrp2"))
mod2use = eval(substitute(lm(formula = lm_formula, data = Dfmri, na.action = na.omit)))
anova(mod2use)
```

```
## Analysis of Variance Table
##
## Response: meanDVARSraw
##            Df Sum Sq Mean Sq F value Pr(>F)
## subgrp2     5   3.38 0.67586  0.2811 0.9231
## Residuals 189 454.45 2.40450
```

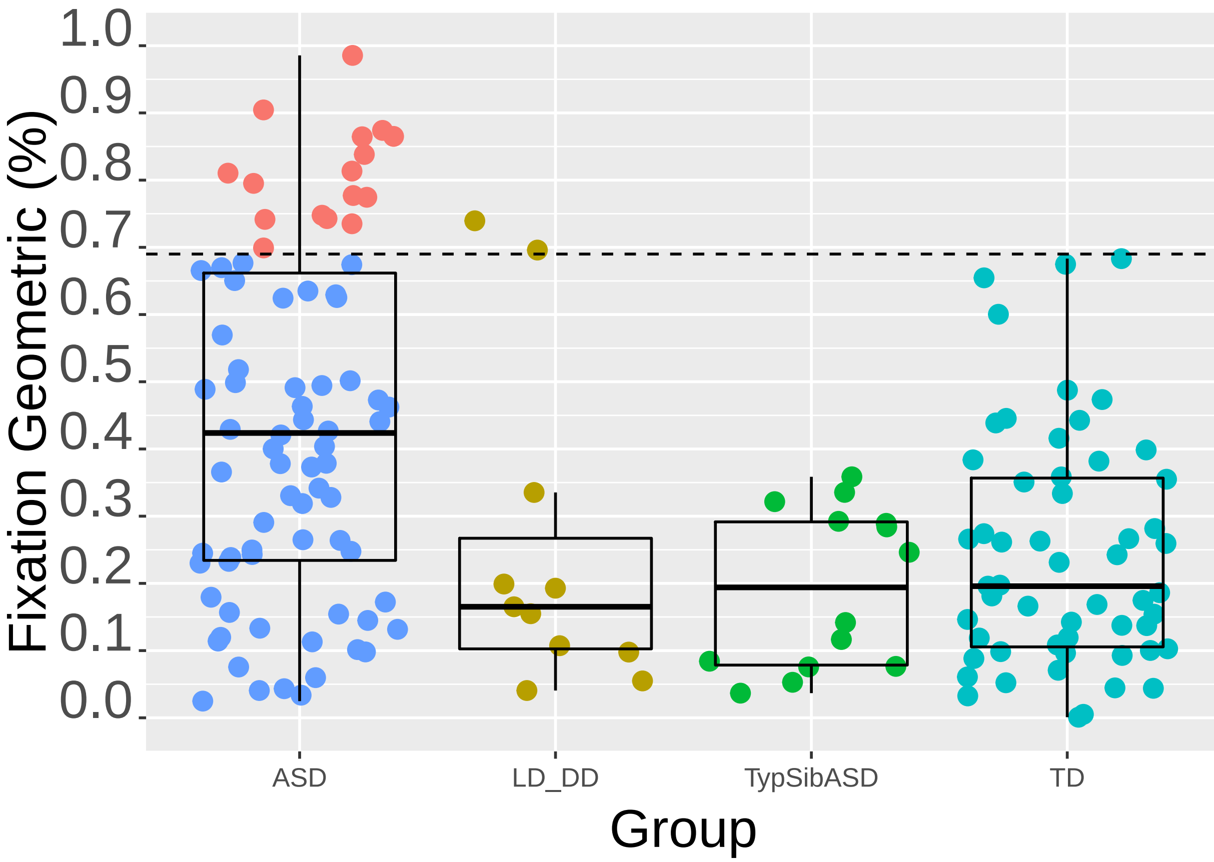
```r
# After
lm_formula = as.formula(sprintf("%s ~ %s","meanDVARSwavelet","subgrp2"))
mod2use = eval(substitute(lm(formula = lm_formula, data = Dfmri, na.action = na.omit)))
anova(mod2use)
```

```
## Analysis of Variance Table
##
## Response: meanDVARSwavelet
##            Df  Sum Sq Mean Sq F value Pr(>F)
## subgrp2     5   1.808 0.36151  0.5716 0.7217
## Residuals 189 119.530 0.63244
```

Make plot of eye tracking data

This plot is from eye tracking data from just the subjects who also have rsfMRI data available.

```r
Dsub = subset(Dfmri, !Dfmri$subgrp2=="ASDnoET", select = 1:ncol(Dfmri))
Dsub$subgrp2 = factor(Dsub$subgrp2)
Dsub$Dx = factor(Dsub$Dx)
Dsub$Dx = factor(Dsub$Dx,levels(Dsub$Dx)[c(1,2,4,3)])

c2use = c(colours2use[1], colours2use[5], colours2use[2:4])
xLabel = "Group"
yLabel = "Fixation Geometric (%)"
p = ggplot(data = Dsub, aes(x = Dx, y = Percent_Fixation_Geometric/100, colour = subgrp2))
p = p + geom_jitter(size = dotSize) +
        geom_boxplot(fill = NA, colour = "#000000", outlier.shape = NA) +
        guides(colour = FALSE)
p = p + scale_y_continuous(limits = c(0,1),breaks = round(seq(from = 0, to = 1, by = 0.1),digits=2))
p = p + geom_hline(yintercept = 0.69, linetype = 2) +
    scale_colour_manual(values = c2use) +
    xlab(xLabel) + ylab(yLabel) +
    theme(axis.text.x = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"),
        axis.text.y = element_text(size=fontSize+10,hjust=1,vjust=0,face="plain"),
        axis.title.x = element_text(size=fontSize+10,hjust=0.5,vjust=0,face="plain"),
        axis.title.y = element_text(size=fontSize+10,hjust=0.5,vjust=0.5,face="plain"),
        strip.text.x = element_text(size = fontSize+10,hjust=0.5,vjust=0.5,face="plain"),
        plot.title = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"))
ggsave(filename = file.path(plotdir,"eyetracking_geoFix_plot.pdf"))
p
```



Read in data for longitudinal clinical trajectory analyses

```
# Read in longitudinal Mullen, Vineland, and ADOS summary score data
Dlw = read_excel(file.path(tidydatadir,"LW_Report_12012016.xlsx"), na = "NULL")
```

Pull out data

```
# masks to pull out groups
na_mask = is.na(Dlw[,1])
asd_mask = is.element(Dlw$dxCode, ASDlabels) & !na_mask
td_mask = is.element(Dlw$dxCode, TDlabels) & !na_mask
dd_mask = is.element(Dlw$dxCode, DDlabels) & !na_mask
ld_mask = is.element(Dlw$dxCode, LDlabels) & !na_mask
other_mask = is.element(Dlw$dxCode, OTHERlabels) & !na_mask
asdfeat_mask = is.element(Dlw$dxCode, ASDFEATlabels) & !na_mask
typsib_mask = is.element(Dlw$dxCode, TYPSIBlabels) & !na_mask

# column labels of interesting data to extract
allLabels = c("VageMo","ComTotal_DomStd","DlyTotal_DomStd","SocTotal_DomStd",
              "MtrTotal_DomStd","AdapBehav_DomStd",
              "AageMo","CoSoTot","RRTot","CoSoTotRRTot",
              "MageMo","VRT","VR_Raw","FMT","FM_Raw",
              "RLT","RL_Raw","ELT","EL_Raw")

# grab the data for each group
```

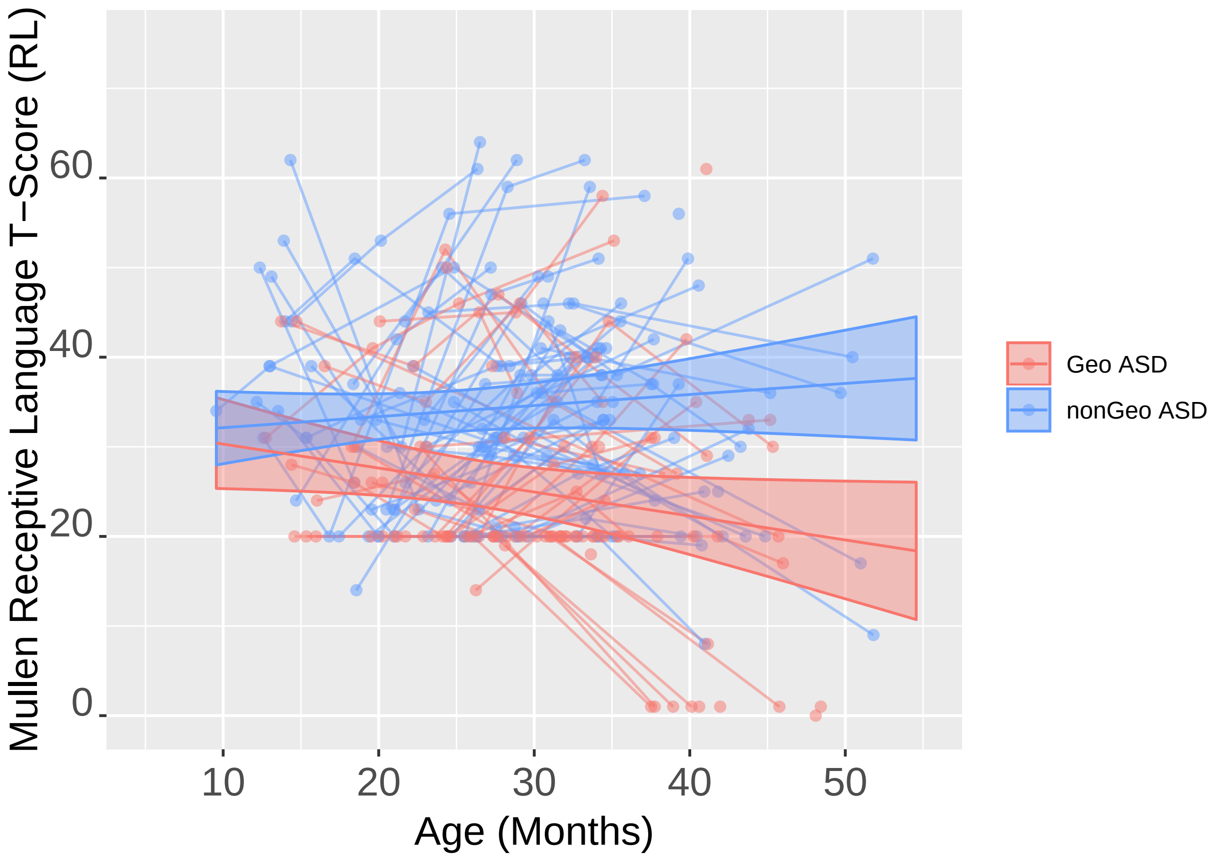
```r
# ASD
asd_df = grabGroupData(D = Dlw, submask = asd_mask, colLabels = allLabels, grpLabel = "ASD")
Det = read.delim(file.path(tidydatadir,"LW_Report_for_ET_N937.txt"),na.strings = c("NA","NULL"))
asd_df = getETsubgrp2(tidy_df = asd_df, full_df = Det, Dx = "ASD")
asd_df = subset(asd_df, asd_df$ETsubgrpDx!="ASD", select = 1:ncol(asd_df))
asd_df = cleanAgeErrors(asd_df)
asd_df$subjectId = factor(asd_df$subjectId)

nongeo_df = subset(asd_df, is.element(asd_df$subjectId,Dfmri$subjectId) & asd_df$ETsubgrpDx=="nonGeo ASD
                   select = 1:ncol(asd_df))
geo_df = subset(asd_df, asd_df$ETsubgrpDx=="Geo ASD",
                select = 1:ncol(asd_df))
fmri_df = rbind(nongeo_df, geo_df)
fmri_df$ETsubgrpDx = factor(fmri_df$ETsubgrpDx)
fmri_df$p2f2 = factor(fmri_df$p2f2)
fmri_df$Dx = factor(fmri_df$Dx)

# set general stuff for the plot
cols2use = get_ggColorHue(6)
cols2use = c(cols2use[1], cols2use[5])

# what data and what subgroup do you want to analyze
df2use = fmri_df
df2use$ETsubgrpDx = as.factor(df2use$ETsubgrpDx)
subgrp_var = "ETsubgrpDx"
```

Mullen Receptive Language

```r
modelType = "linear"
xLabel = "Age (Months)"
plot_dots = TRUE
plot_lines = TRUE
ci_band = TRUE
dot_alpha = 5/10
line_alpha = 5/10
band_alpha = 4/10
standardize = TRUE

x_var = "MageMo"
xLimits = c(5,55)
yLimits = c(0,75)

yLabel = "Mullen Receptive Language T-Score (RL)"
y_var = "RLT"

RLT_all_quad = spaghettiPlot(df = df2use, x_var = x_var, y_var = y_var,
    subgrp_var = subgrp_var, xLabel = xLabel, modelType = modelType,
    yLabel = yLabel, fname2save = NULL,
    plot_dots = plot_dots, plot_lines = plot_lines, ci_band = ci_band,
    dot_alpha = dot_alpha, line_alpha = line_alpha, band_alpha = band_alpha,
    xLimits = xLimits, yLimits = yLimits, standardize = standardize)
anova(RLT_all_quad$lme_model)
```

```
##                        numDF denDF   F-value p-value
## (Intercept)                1   165  0.026293  0.8714
## MageMo                     1   165  0.714530  0.3992
## ETsubgrpDx                 1   117 21.927787  <.0001
## MageMo:ETsubgrpDx          1   165  5.223953  0.0236
```

```
# change the coloring of the groups to match other figures
p = RLT_all_quad$p
p = p + scale_colour_manual(values = cols2use) + scale_fill_manual(values = cols2use) +
        theme(axis.text.x = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            axis.text.y = element_text(size=fontSize+5,hjust=1,vjust=0,face="plain"),
            axis.title.x = element_text(size=fontSize+5,hjust=0.5,vjust=0,face="plain"),
            axis.title.y = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            strip.text.x = element_text(size = fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            plot.title = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"))
ggsave(filename = file.path(plotdir,sprintf("Mullen_%s_traj_plot.pdf",y_var)))
p
```



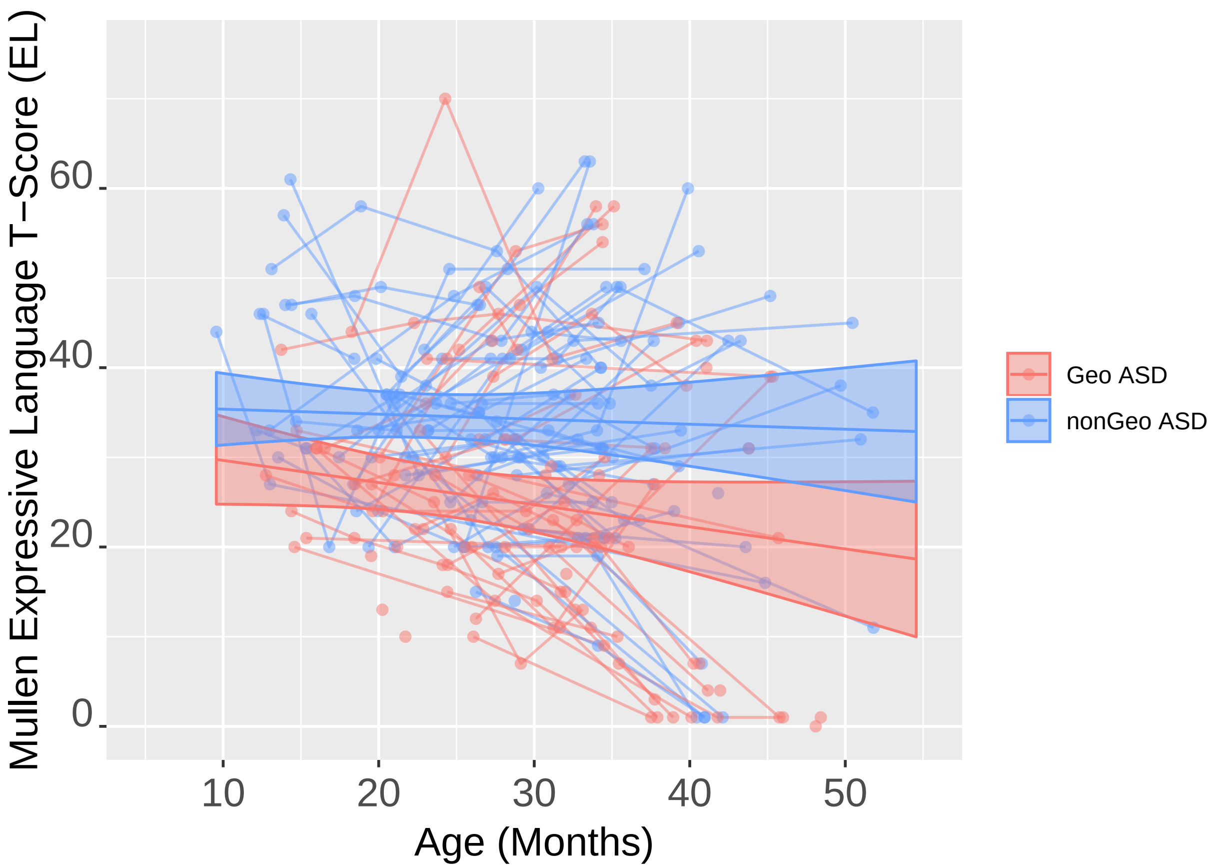
Mullen Expressive Language

```
yLabel = "Mullen Expressive Language T-Score (EL)"
y_var = "ELT"

ELT_all_quad = spaghettiPlot(df = df2use, x_var = x_var, y_var = y_var,
    subgrp_var = subgrp_var, xLabel = xLabel, modelType = modelType,
    yLabel = yLabel, fname2save = NULL,
```

```
    plot_dots = plot_dots, plot_lines = plot_lines, ci_band = ci_band,
    dot_alpha = dot_alpha, line_alpha = line_alpha, band_alpha = band_alpha,
    xLimits = xLimits, yLimits = yLimits, standardize = standardize)
anova(ELT_all_quad$lme_model)
```

```
##                    numDF denDF   F-value p-value
## (Intercept)            1   165  0.243046  0.6227
## MageMo                 1   165  3.417672  0.0663
## ETsubgrpDx             1   117 21.185912  <.0001
## MageMo:ETsubgrpDx      1   165  1.074913  0.3014
```

```
# change the coloring of the groups to match other figures
p = ELT_all_quad$p
p = p + scale_colour_manual(values = cols2use) + scale_fill_manual(values = cols2use) +
        theme(axis.text.x = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            axis.text.y = element_text(size=fontSize+5,hjust=1,vjust=0,face="plain"),
            axis.title.x = element_text(size=fontSize+5,hjust=0.5,vjust=0,face="plain"),
            axis.title.y = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            strip.text.x = element_text(size = fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            plot.title = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"))
ggsave(filename = file.path(plotdir,sprintf("Mullen_%s_traj_plot.pdf",y_var)))
p
```

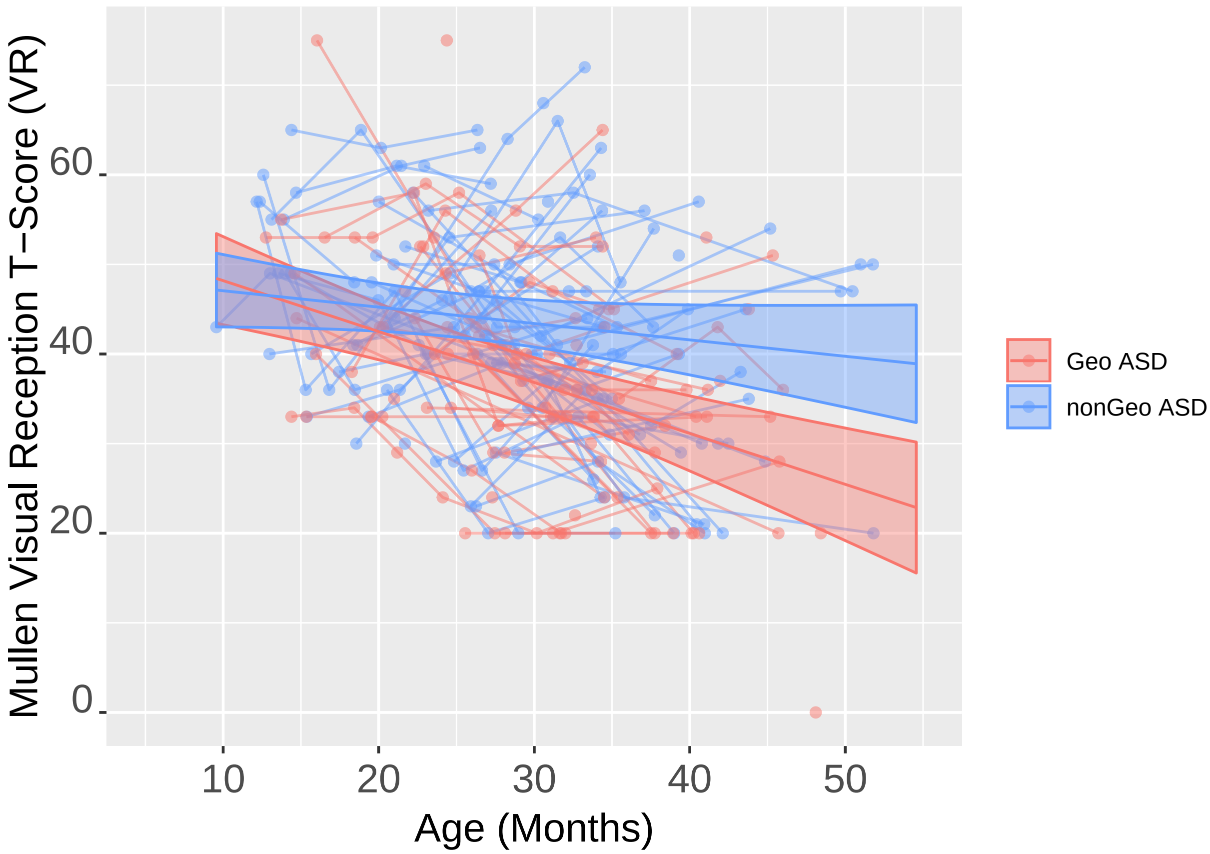


Mullen Visual Reception

```r
yLabel = "Mullen Visual Reception T-Score (VR)"
y_var = "VRT"

VRT_all_quad = spaghettiPlot(df = df2use, x_var = x_var, y_var = y_var,
    subgrp_var = subgrp_var, xLabel = xLabel, modelType = modelType,
    yLabel = yLabel, fname2save = NULL,
    plot_dots = plot_dots, plot_lines = plot_lines, ci_band = ci_band,
    dot_alpha = dot_alpha, line_alpha = line_alpha, band_alpha = band_alpha,
    xLimits = xLimits, yLimits = yLimits, standardize = standardize)
anova(VRT_all_quad$lme_model)
```

```
##                   numDF denDF   F-value p-value
## (Intercept)           1   165  1.297810  0.2563
## MageMo                1   165 20.157709  <.0001
## ETsubgrpDx            1   117  7.910200  0.0058
## MageMo:ETsubgrpDx     1   165  5.639771  0.0187
```

```r
# change the coloring of the groups to match other figures
p = VRT_all_quad$p
p = p + scale_colour_manual(values = cols2use) + scale_fill_manual(values = cols2use) +
        theme(axis.text.x = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            axis.text.y = element_text(size=fontSize+5,hjust=1,vjust=0,face="plain"),
            axis.title.x = element_text(size=fontSize+5,hjust=0.5,vjust=0,face="plain"),
            axis.title.y = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            strip.text.x = element_text(size = fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            plot.title = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"))
ggsave(filename = file.path(plotdir,sprintf("Mullen_%s_traj_plot.pdf",y_var)))
p
```



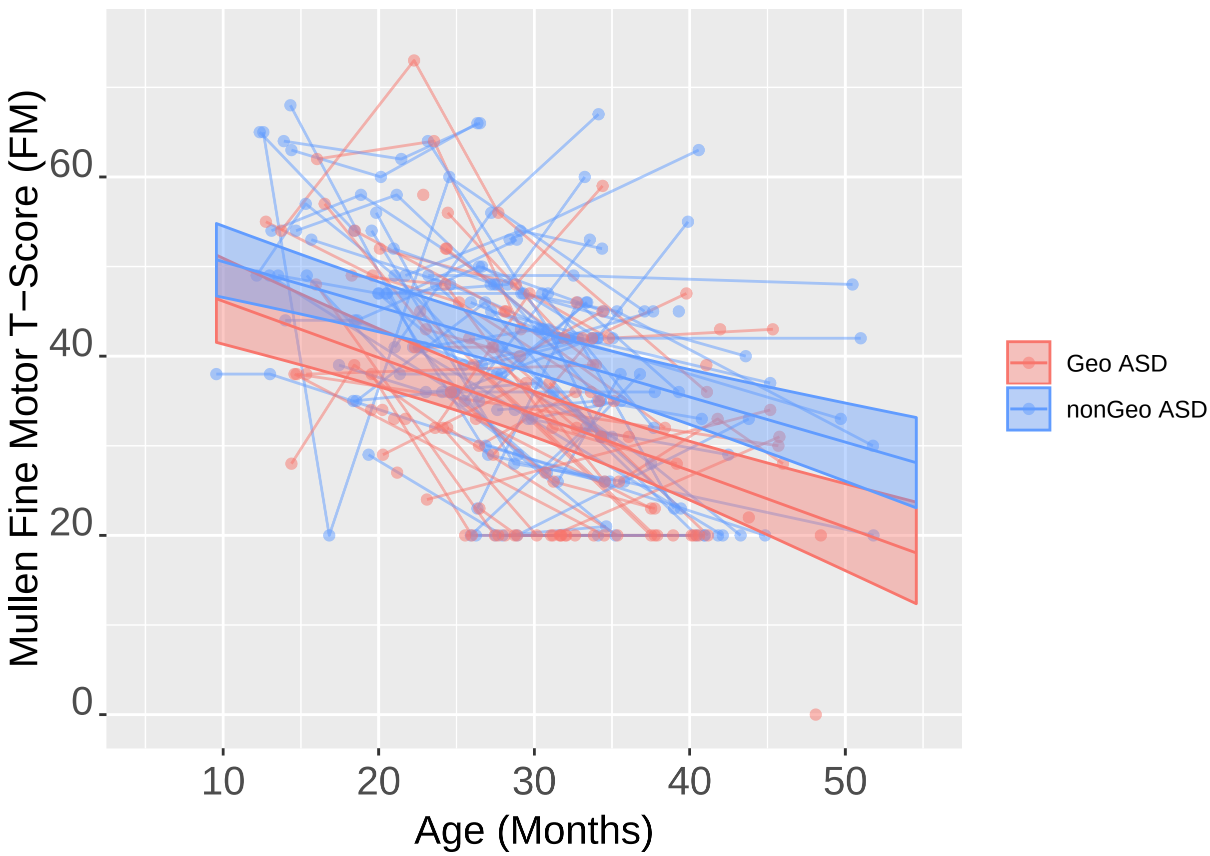
Mullen Fine Motor

```
yLabel = "Mullen Fine Motor T-Score (FM)"
y_var = "FMT"

FMT_all_quad = spaghettiPlot(df = df2use, x_var = x_var, y_var = y_var,
    subgrp_var = subgrp_var, xLabel = xLabel, modelType = modelType,
    yLabel = yLabel, fname2save = NULL,
    plot_dots = plot_dots, plot_lines = plot_lines, ci_band = ci_band,
    dot_alpha = dot_alpha, line_alpha = line_alpha, band_alpha = band_alpha,
    xLimits = xLimits, yLimits = yLimits, standardize = standardize)
anova(FMT_all_quad$lme_model)
```

```
##                    numDF denDF  F-value p-value
## (Intercept)            1   165  0.14507  0.7038
## MageMo                 1   165 76.64519  <.0001
## ETsubgrpDx             1   117 15.33978  0.0002
## MageMo:ETsubgrpDx      1   165  0.90434  0.3430
```

```
# change the coloring of the groups to match other figures
p = FMT_all_quad$p
p = p + scale_colour_manual(values = cols2use) + scale_fill_manual(values = cols2use) +
        theme(axis.text.x = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            axis.text.y = element_text(size=fontSize+5,hjust=1,vjust=0,face="plain"),
            axis.title.x = element_text(size=fontSize+5,hjust=0.5,vjust=0,face="plain"),
            axis.title.y = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
```

```
            strip.text.x = element_text(size = fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            plot.title = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"))
ggsave(filename = file.path(plotdir,sprintf("Mullen_%s_traj_plot.pdf",y_var)))
p
```



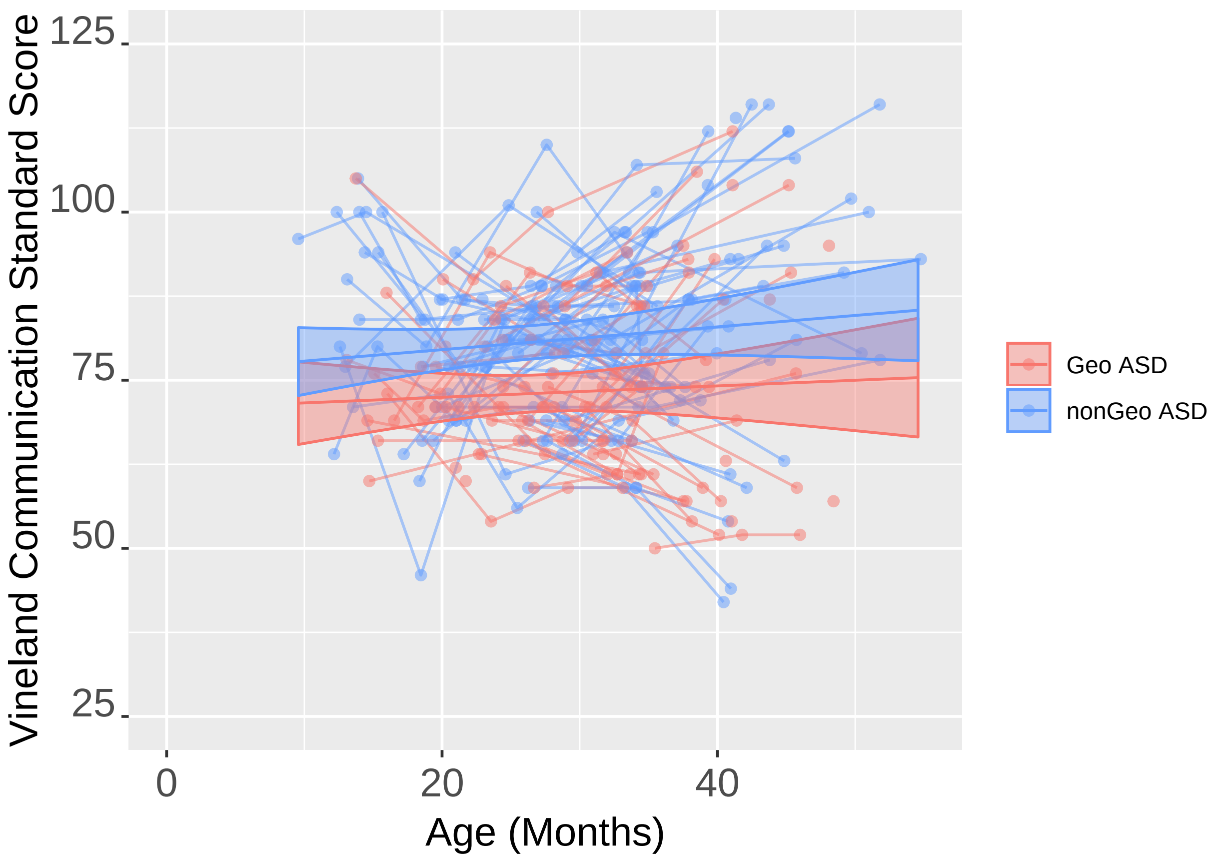
Vineland Communication

```
x_var = "VageMo"
xLimits = c(0,55)
yLimits = c(25,125)

yLabel = "Vineland Communication Standard Score"
y_var = "ComTotal_DomStd"

VineComm_all_quad = spaghettiPlot(df = df2use, x_var = x_var, y_var = y_var,
    subgrp_var = subgrp_var, xLabel = xLabel, modelType = modelType,
    yLabel = yLabel, fname2save = NULL,
    plot_dots = plot_dots, plot_lines = plot_lines, ci_band = ci_band,
    dot_alpha = dot_alpha, line_alpha = line_alpha, band_alpha = band_alpha,
    xLimits = xLimits, yLimits = yLimits, standardize = standardize)
anova(VineComm_all_quad$lme_model)
```

```
##                     numDF denDF   F-value p-value
## (Intercept)             1   185  3.439942  0.0652
## VageMo                  1   185  1.514647  0.2200
```

```
## ETsubgrpDx           1   120 17.142872  0.0001
## VageMo:ETsubgrpDx     1   185  0.179004  0.6727
```

```
# change the coloring of the groups to match other figures
p = VineComm_all_quad$p
p = p + scale_colour_manual(values = cols2use) + scale_fill_manual(values = cols2use) +
        theme(axis.text.x = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            axis.text.y = element_text(size=fontSize+5,hjust=1,vjust=0,face="plain"),
            axis.title.x = element_text(size=fontSize+5,hjust=0.5,vjust=0,face="plain"),
            axis.title.y = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            strip.text.x = element_text(size = fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            plot.title = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"))
ggsave(filename = file.path(plotdir,sprintf("Vineland_%s_traj_plot.pdf",y_var)))
p
```



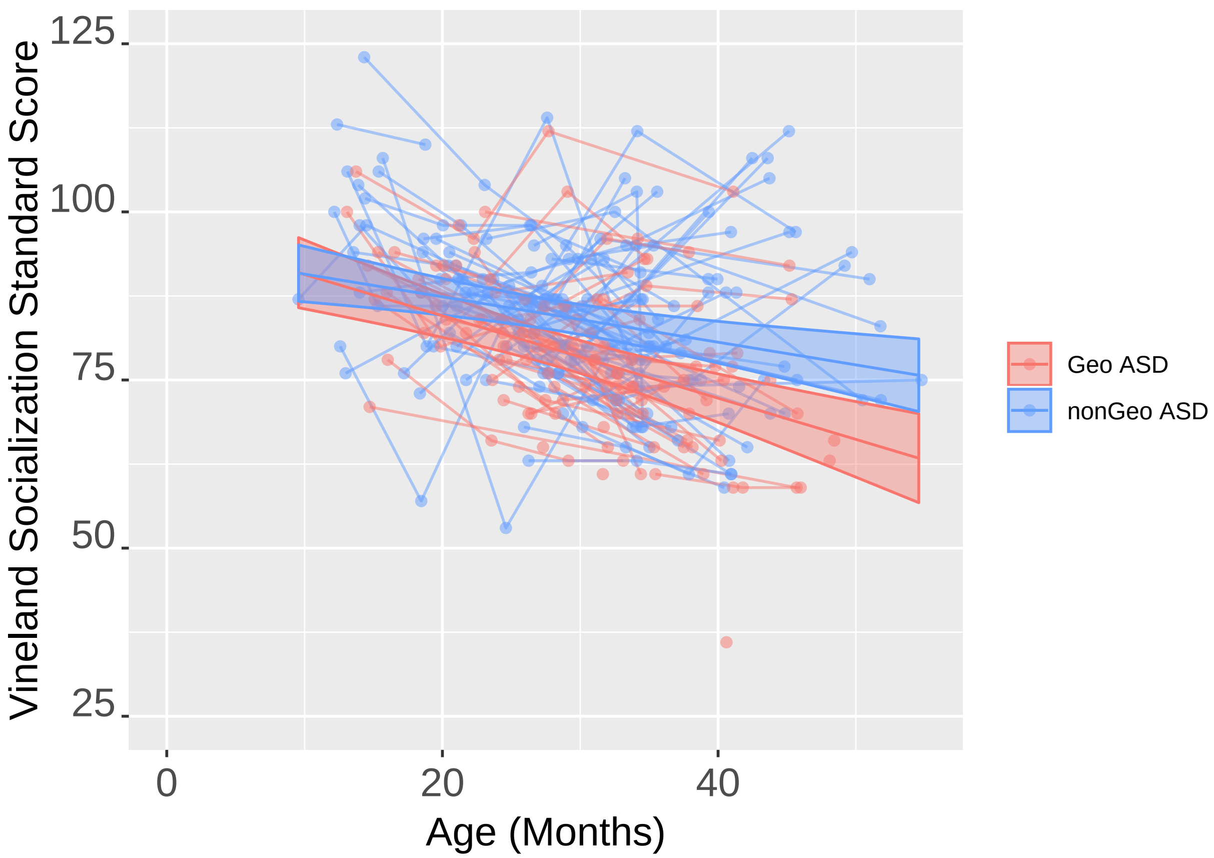
Vineland Socialization

```
yLabel = "Vineland Socialization Standard Score"
y_var = "SocTotal_DomStd"

VineSoc_all_quad = spaghettiPlot(df = df2use, x_var = x_var, y_var = y_var,
    subgrp_var = subgrp_var, xLabel = xLabel, modelType = modelType,
    yLabel = yLabel, fname2save = NULL,
    plot_dots = plot_dots, plot_lines = plot_lines, ci_band = ci_band,
    dot_alpha = dot_alpha, line_alpha = line_alpha, band_alpha = band_alpha,
    xLimits = xLimits, yLimits = yLimits, standardize = standardize)
```

```r
anova(VineSoc_all_quad$lme_model)
```

```
##                    numDF denDF  F-value p-value
## (Intercept)            1   185  0.00000  0.9985
## VageMo                 1   185 35.98014  <.0001
## ETsubgrpDx             1   120  9.91304  0.0021
## VageMo:ETsubgrpDx      1   185  3.23199  0.0738
```

```r
# change the coloring of the groups to match other figures
p = VineSoc_all_quad$p
p = p + scale_colour_manual(values = cols2use) + scale_fill_manual(values = cols2use) +
        theme(axis.text.x = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            axis.text.y = element_text(size=fontSize+5,hjust=1,vjust=0,face="plain"),
            axis.title.x = element_text(size=fontSize+5,hjust=0.5,vjust=0,face="plain"),
            axis.title.y = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            strip.text.x = element_text(size = fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            plot.title = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"))
ggsave(filename = file.path(plotdir,sprintf("Vineland_%s_traj_plot.pdf",y_var)))
p
```



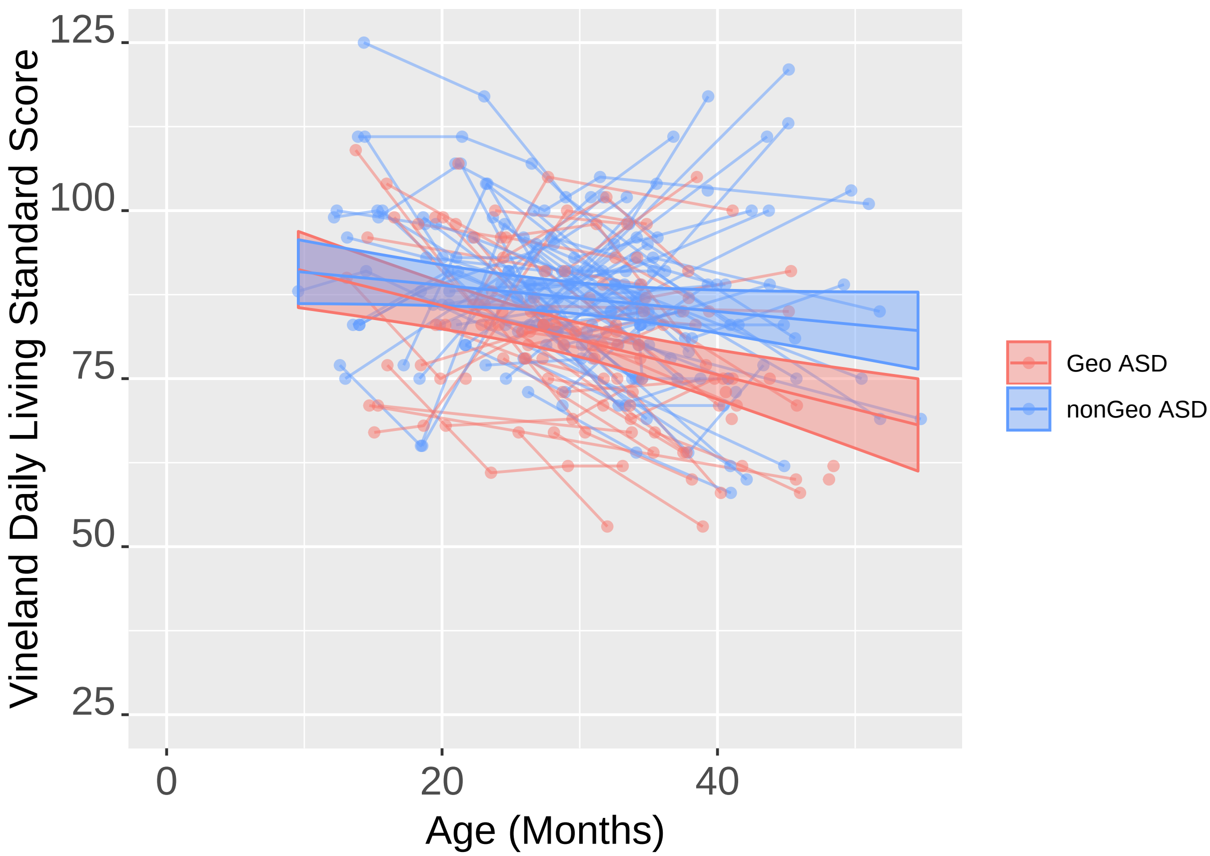
Vineland Daily Living

```r
yLabel = "Vineland Daily Living Standard Score"
y_var = "DlyTotal_DomStd"

VineDly_all_quad = spaghettiPlot(df = df2use, x_var = x_var, y_var = y_var,
```

```
    subgrp_var = subgrp_var, xLabel = xLabel, modelType = modelType,
    yLabel = yLabel, fname2save = NULL,
    plot_dots = plot_dots, plot_lines = plot_lines, ci_band = ci_band,
    dot_alpha = dot_alpha, line_alpha = line_alpha, band_alpha = band_alpha,
    xLimits = xLimits, yLimits = yLimits, standardize = standardize)
anova(VineDly_all_quad$lme_model)
```

```
##                    numDF denDF   F-value p-value
## (Intercept)            1   185  0.133603  0.7151
## VageMo                 1   185 15.883371  0.0001
## ETsubgrpDx             1   120 12.239522  0.0007
## VageMo:ETsubgrpDx      1   185  3.699051  0.0560
```

```
# change the coloring of the groups to match other figures
p = VineDly_all_quad$p
p = p + scale_colour_manual(values = cols2use) + scale_fill_manual(values = cols2use) +
        theme(axis.text.x = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            axis.text.y = element_text(size=fontSize+5,hjust=1,vjust=0,face="plain"),
            axis.title.x = element_text(size=fontSize+5,hjust=0.5,vjust=0,face="plain"),
            axis.title.y = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            strip.text.x = element_text(size = fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            plot.title = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"))
ggsave(filename = file.path(plotdir,sprintf("Vineland_%s_traj_plot.pdf",y_var)))
p
```

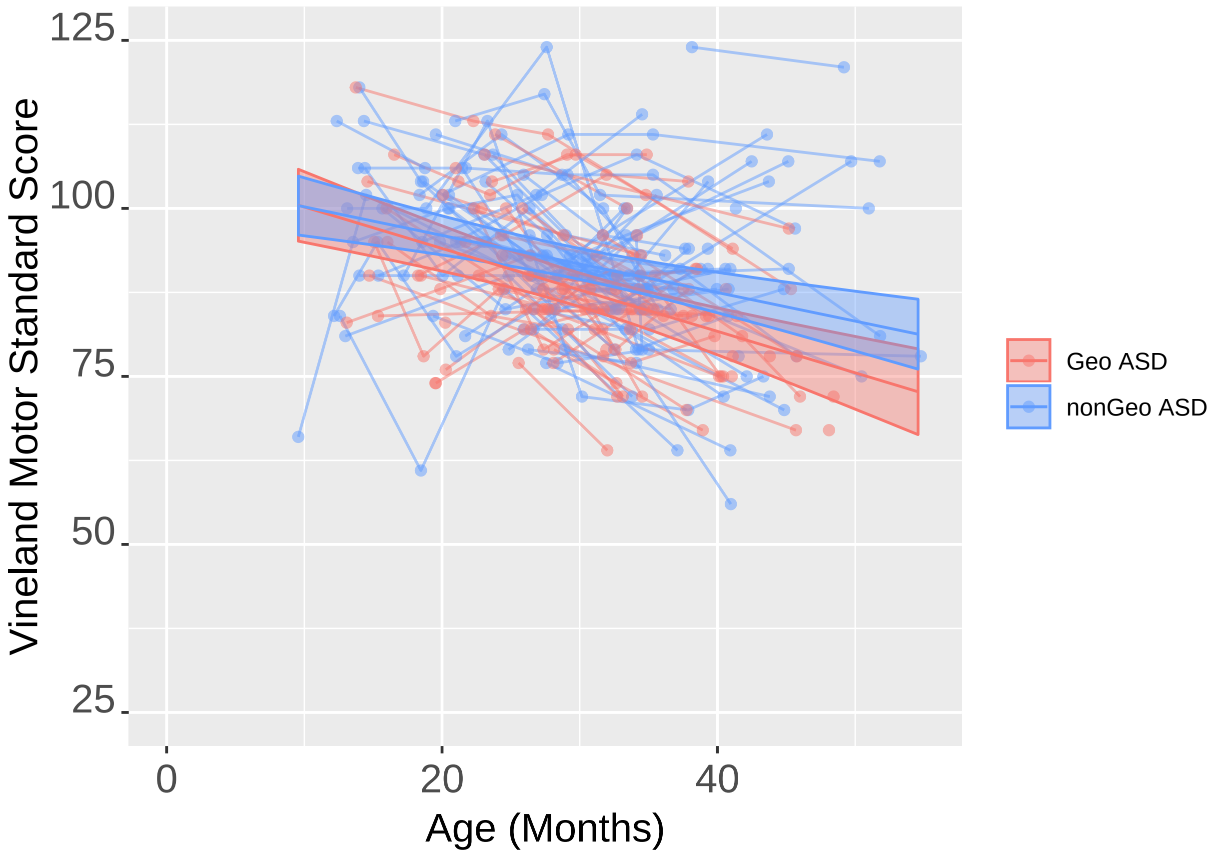


Vineland Motor

```
yLabel = "Vineland Motor Standard Score"
y_var = "MtrTotal_DomStd"

VineMtr_all_quad = spaghettiPlot(df = df2use, x_var = x_var, y_var = y_var,
    subgrp_var = subgrp_var, xLabel = xLabel, modelType = modelType,
    yLabel = yLabel, fname2save = NULL,
    plot_dots = plot_dots, plot_lines = plot_lines, ci_band = ci_band,
    dot_alpha = dot_alpha, line_alpha = line_alpha, band_alpha = band_alpha,
    xLimits = xLimits, yLimits = yLimits, standardize = standardize)
anova(VineMtr_all_quad$lme_model)
```

```
##                   numDF denDF  F-value p-value
## (Intercept)           1   185  0.62540  0.4301
## VageMo                1   185 46.10245  <.0001
## ETsubgrpDx            1   120  4.89719  0.0288
## VageMo:ETsubgrpDx     1   185  1.60509  0.2068
```

```
# change the coloring of the groups to match other figures
p = VineMtr_all_quad$p
p = p + scale_colour_manual(values = cols2use) + scale_fill_manual(values = cols2use) +
        theme(axis.text.x = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            axis.text.y = element_text(size=fontSize+5,hjust=1,vjust=0,face="plain"),
            axis.title.x = element_text(size=fontSize+5,hjust=0.5,vjust=0,face="plain"),
            axis.title.y = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            strip.text.x = element_text(size = fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            plot.title = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"))
ggsave(filename = file.path(plotdir,sprintf("Vineland_%s_traj_plot.pdf",y_var)))
p
```



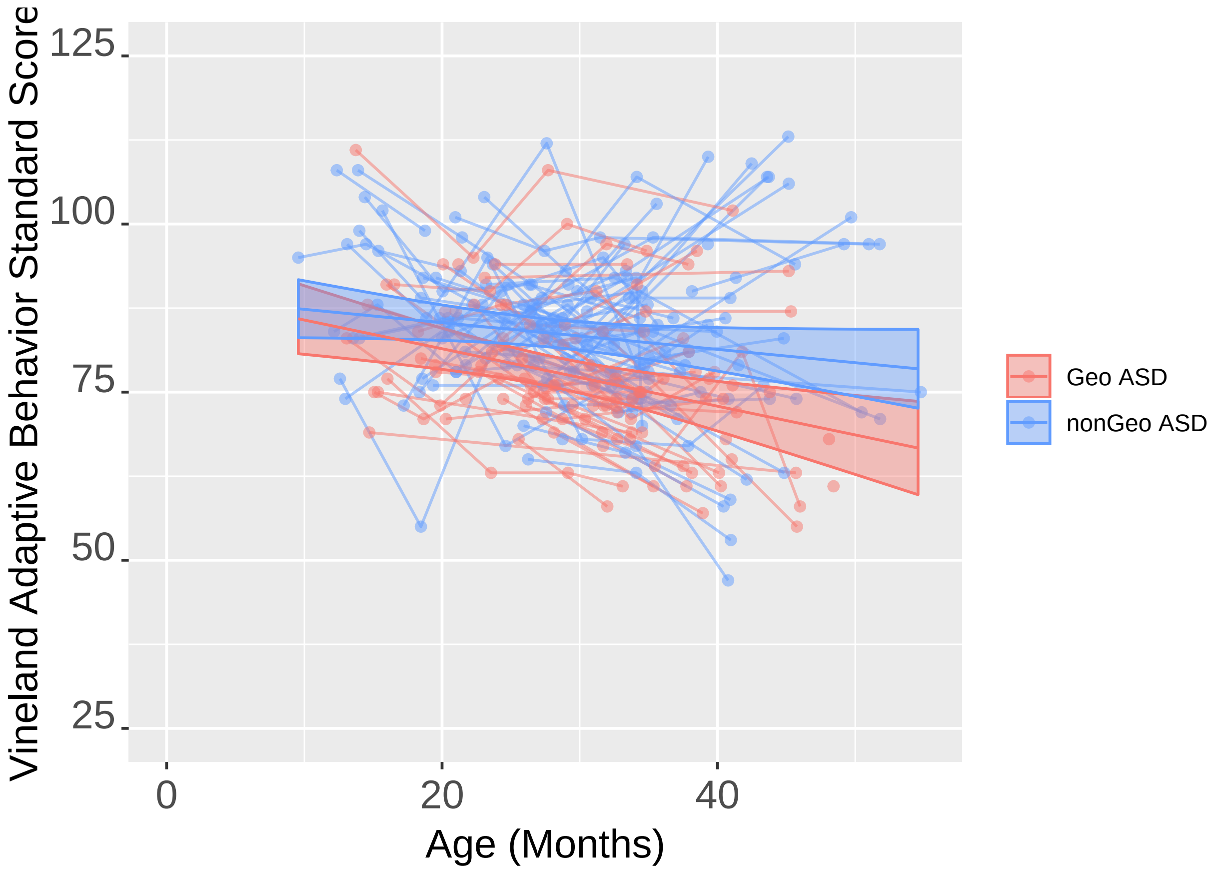
Vineland Adaptive Behavior

```
yLabel = "Vineland Adaptive Behavior Standard Score"
y_var = "AdapBehav_DomStd"

VineAdapBehav_all_quad = spaghettiPlot(df = df2use, x_var = x_var, y_var = y_var,
    subgrp_var = subgrp_var, xLabel = xLabel, modelType = modelType,
    yLabel = yLabel, fname2save = NULL,
    plot_dots = plot_dots, plot_lines = plot_lines, ci_band = ci_band,
    dot_alpha = dot_alpha, line_alpha = line_alpha, band_alpha = band_alpha,
    xLimits = xLimits, yLimits = yLimits, standardize = standardize)
anova(VineAdapBehav_all_quad$lme_model)
```

```
##                     numDF denDF   F-value p-value
## (Intercept)             1   185  0.229060  0.6328
## VageMo                  1   185 13.748281  0.0003
## ETsubgrpDx              1   120 13.889196  0.0003
## VageMo:ETsubgrpDx       1   185  1.967818  0.1624
```

```
# change the coloring of the groups to match other figures
p = VineAdapBehav_all_quad$p
p = p + scale_colour_manual(values = cols2use) + scale_fill_manual(values = cols2use) +
        theme(axis.text.x = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            axis.text.y = element_text(size=fontSize+5,hjust=1,vjust=0,face="plain"),
            axis.title.x = element_text(size=fontSize+5,hjust=0.5,vjust=0,face="plain"),
            axis.title.y = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
```

```r
            strip.text.x = element_text(size = fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            plot.title = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"))
ggsave(filename = file.path(plotdir,sprintf("Vineland_%s_traj_plot.pdf",y_var)))
p
```



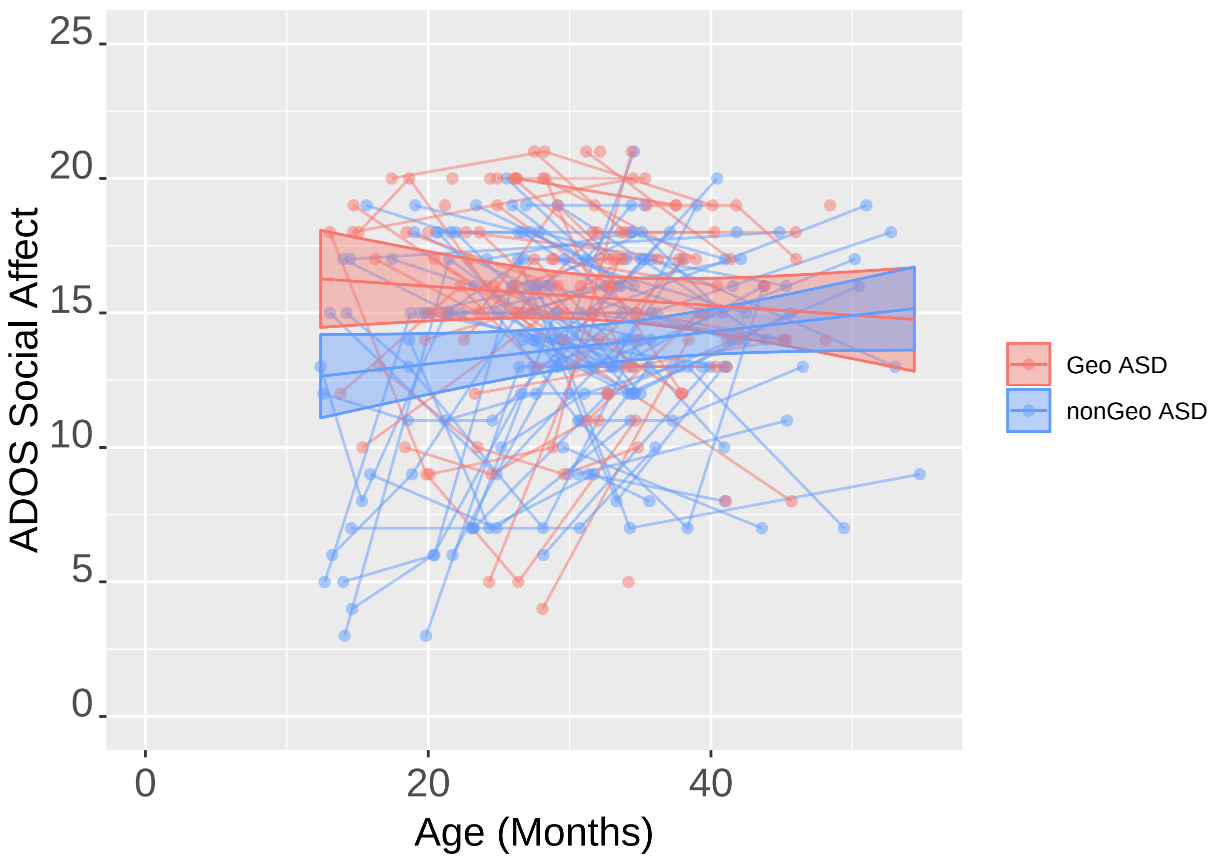
ADOS Social Affect

```r
x_var = "AageMo"
xLimits = c(0,55)
yLimits = c(0,25)

yLabel = "ADOS Social Affect"
y_var = "CoSoTot"

ADOSCoSo_all_quad = spaghettiPlot(df = df2use, x_var = x_var, y_var = y_var,
    subgrp_var = subgrp_var, xLabel = xLabel, modelType = modelType,
    yLabel = yLabel, fname2save = NULL,
    plot_dots = plot_dots, plot_lines = plot_lines, ci_band = ci_band,
    dot_alpha = dot_alpha, line_alpha = line_alpha, band_alpha = band_alpha,
    xLimits = xLimits, yLimits = yLimits, standardize = standardize)
anova(ADOSCoSo_all_quad$lme_model)
```

```
##                      numDF denDF  F-value p-value
## (Intercept)              1   181 0.975626  0.3246
## AageMo                   1   181 0.549231  0.4596
```

```
## ETsubgrpDx            1   120 9.074773  0.0032
## AageMo:ETsubgrpDx     1   181 3.418836  0.0661
```

```
# change the coloring of the groups to match other figures
p = ADOSCoSo_all_quad$p
p = p + scale_colour_manual(values = cols2use) + scale_fill_manual(values = cols2use) +
        theme(axis.text.x = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            axis.text.y = element_text(size=fontSize+5,hjust=1,vjust=0,face="plain"),
            axis.title.x = element_text(size=fontSize+5,hjust=0.5,vjust=0,face="plain"),
            axis.title.y = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            strip.text.x = element_text(size = fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            plot.title = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"))
ggsave(filename = file.path(plotdir,sprintf("ADOS_%s_traj_plot.pdf",y_var)))
p
```



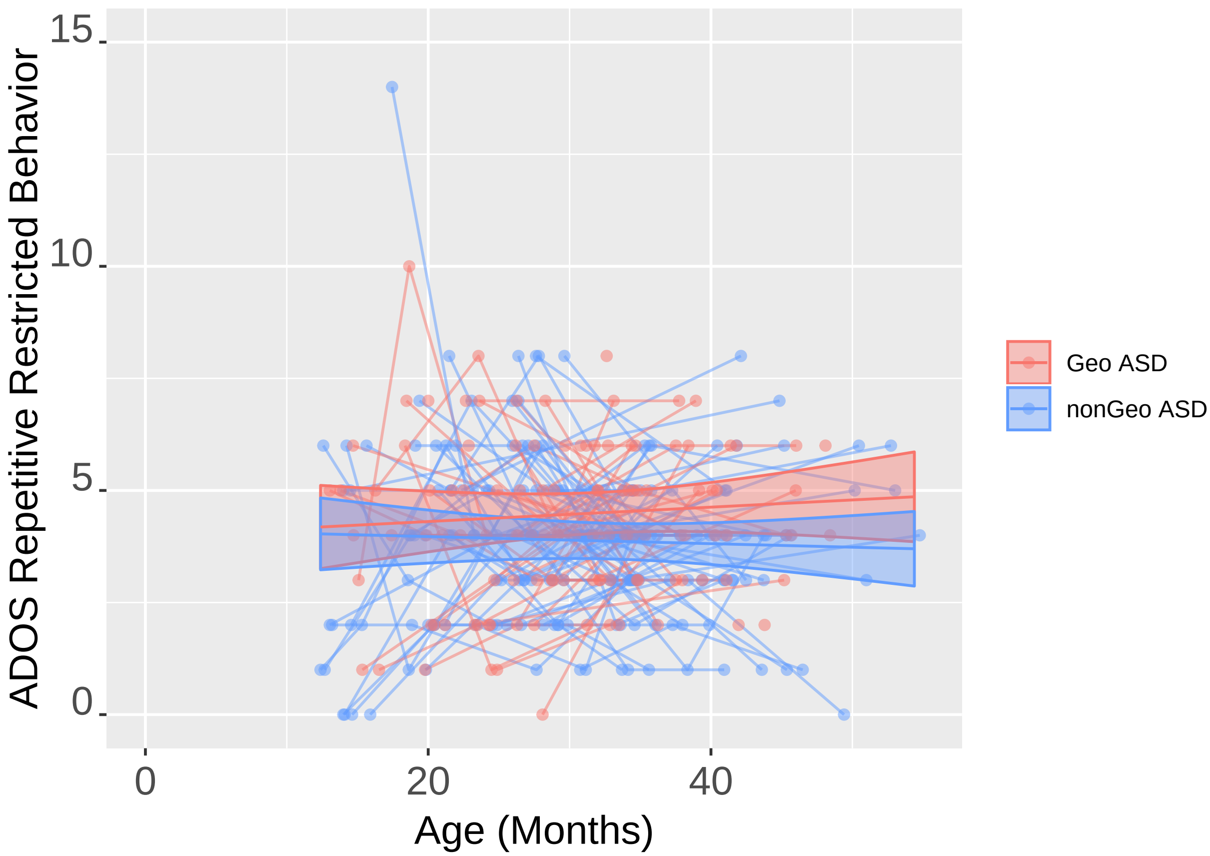
ADOS Repetitive Restricted Behavior

```
yLabel = "ADOS Repetitive Restricted Behavior"
y_var = "RRTot"
yLimits = c(0,15)

ADOSRRB_all_quad = spaghettiPlot(df = df2use, x_var = x_var, y_var = y_var,
    subgrp_var = subgrp_var, xLabel = xLabel, modelType = modelType,
    yLabel = yLabel, fname2save = NULL,
    plot_dots = plot_dots, plot_lines = plot_lines, ci_band = ci_band,
    dot_alpha = dot_alpha, line_alpha = line_alpha, band_alpha = band_alpha,
```

```
    xLimits = xLimits, yLimits = yLimits, standardize = standardize)
anova(ADOSRRB_all_quad$lme_model)
```

```
##                   numDF denDF  F-value p-value
## (Intercept)           1   181 0.211298  0.6463
## AageMo                1   181 0.131105  0.7177
## ETsubgrpDx            1   120 5.038428  0.0266
## AageMo:ETsubgrpDx     1   181 1.031324  0.3112
```

```
# change the coloring of the groups to match other figures
p = ADOSRRB_all_quad$p
p = p + scale_colour_manual(values = cols2use) + scale_fill_manual(values = cols2use) +
        theme(axis.text.x = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            axis.text.y = element_text(size=fontSize+5,hjust=1,vjust=0,face="plain"),
            axis.title.x = element_text(size=fontSize+5,hjust=0.5,vjust=0,face="plain"),
            axis.title.y = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            strip.text.x = element_text(size = fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            plot.title = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"))
ggsave(filename = file.path(plotdir,sprintf("ADOS_%s_traj_plot.pdf",y_var)))
p
```



Compute partial correlations in MATLAB and set up data to run tests

Use R to run MATLAB code to estimate partial correlations (Tikhonov regularisation) with FSLNets code. Relies on MATLAB function estimateConnectivity.m being placed in your code directory.

```r
if (RUNMATLAB){
  code2run = sprintf("cd %s; estimateConnectivity;",codedir)
  res = run_matlab_code(code2run)
}

dfname = file.path(tidydatadir,"partialCorDataASDTDLDDDSIB_ridge_lambda1.txt")
D = read.delim(dfname, na.strings = c("NA", "NaN", " "))
```

Split up data into subsets for further analyses.

```r
# make sure variables are factors
D$subjectId = factor(D$subjectId)
D$sex = factor(D$sex)
D$subgrp = factor(D$subgrp)
D$subgrp = factor(D$subgrp,levels(D$subgrp)[c(2,4,3,6,5,1)])
D$CaseControl = factor(D$CaseControl)

# find variables names for the IC-pairs
vars2use = colnames(D)[8:ncol(D)]

D$CaseControl2 = as.character(D$CaseControl)
D$CaseControl2[D$subgrp=="LD_DD"] = "LD_DD"
D$CaseControl2[D$subgrp=="TypSibASD"] = "TypSibASD"
D$CaseControl2 = factor(D$CaseControl2)

# grab subsets of data for expt 2
Dexp2_all = subset(D, is.element(D$subgrp,c("GeoASD","nonGeoASD","LD_DD","TypSibASD","TD")),
               select = 1:ncol(D))
Dexp2_all$subgrp = factor(Dexp2_all$subgrp)

colours2use = get_ggColorHue(6)
colours2use = c(colours2use[1],colours2use[5],colours2use[2:4])

Dtmp = merge(Dexp2_all,Dfmri, by = "subjectId")
Dtmp$subgrp = factor(Dtmp$subgrp.x)
Dtmp$sex = factor(Dtmp$sex.x)
Dtmp$scan_age = Dtmp$scan_age.x
```

Run ANOVAs on each IC-pair on CaseControl, Sex, Scan Age model

```r
colnames2use = c("df1_all","df2_all",
                 "Fstat_all","pval_all","etasq_all","fdr_all",
                 "df1_subtype","df2_subtype",
                 "fstat_subtype","pval_subtype","etasq_subtype","fdr_subtype",
                 "fstat_subtypeDVARScov","pval_subtypeDVARScov","fdr_subtypeDVARScov",
                 "AIC_nosubtype","AIC_subtype","AIC_delta")

aovres = data.frame(matrix(nrow = length(vars2use),ncol =length(colnames2use)))
colnames(aovres) = colnames2use
rownames(aovres) = vars2use

for (i in 1:length(vars2use)) {
    y_var = vars2use[i]
```

```r
  # construct linear model for ASD vs TD vs LD/DD vs TD ASDSib
  lm_formula = as.formula(sprintf("%s ~ %s + %s + %s",y_var,"CaseControl2","sex","scan_age"))
  mod2use = eval(substitute(lm(formula = lm_formula, data = D, na.action = na.omit)))

  # run ANOVA
  res = anova(mod2use)
  # extract F-stat and pvalue
  Fstat = res["CaseControl2","F value"]
  pval = res["CaseControl2","Pr(>F)"]
  etasq_res = etasq(mod2use)
  aovres$df1_all[i] = res["CaseControl2","Df"]
  aovres$df2_all[i] = res["Residuals","Df"]
  aovres$Fstat_all[i] = Fstat
  aovres$pval_all[i] = pval
  aovres$etasq_all[i] = etasq_res["CaseControl2","Partial eta^2"]


  # get residual for effect size
  # remove variation from covariate
  covname2use = c("sexM","scan_age")
  beta1 = mod2use$coefficients[covname2use, drop = FALSE]
  beta1[is.na(beta1)] = 0
  full_model = model.matrix(~0+as.factor(CaseControl2) + as.factor(sex) + scan_age, data=D)
  colnames(full_model) = c("ASD","LD_DD","TD","TypSibASD","sex","scan_age")
  covname2use = c("sex","scan_age")
  D$covadj = as.numeric(t(D[,y_var] - beta1 %*% t(full_model[,covname2use])))

  # specific ASD pairwise comparisons
  # TD vs ASD
  pw_comp_res = t.test(D[D$CaseControl2=="TD",y_var],D[D$CaseControl2=="ASD",y_var])
aovres$TD_vs_ASD_t[i] = pw_comp_res$statistic
aovres$TD_vs_ASD_p[i] = pw_comp_res$p.value
Dsubset = subset(D,D$CaseControl2=="TD" | D$CaseControl2=="ASD")
Dsubset$CaseControl2 = factor(Dsubset$CaseControl2)
dres = effsize::cohen.d(Dsubset$covadj,Dsubset[,"CaseControl2"])
aovres$TD_vs_ASD_d[i] = dres$estimate

# TypSibASD vs ASD
  pw_comp_res = t.test(D[D$CaseControl2=="TypSibASD",y_var],D[D$CaseControl2=="ASD",y_var])
aovres$TypSibASD_vs_ASD_t[i] = pw_comp_res$statistic
aovres$TypSibASD_vs_ASD_p[i] = pw_comp_res$p.value
Dsubset = subset(D,D$CaseControl2=="TypSibASD" | D$CaseControl2=="ASD")
Dsubset$CaseControl2 = factor(Dsubset$CaseControl2)
dres = effsize::cohen.d(Dsubset$covadj,Dsubset[,"CaseControl2"])
aovres$TypSibASD_vs_ASD_d[i] = dres$estimate

# LD_DD vs ASD
  pw_comp_res = t.test(D[D$CaseControl2=="LD_DD",y_var],D[D$CaseControl2=="ASD",y_var])
aovres$LDDD_vs_ASD_t[i] = pw_comp_res$statistic
aovres$LDDD_vs_ASD_p[i] = pw_comp_res$p.value
Dsubset = subset(D,D$CaseControl2=="LD_DD" | D$CaseControl2=="ASD")
Dsubset$CaseControl2 = factor(Dsubset$CaseControl2)
dres = effsize::cohen.d(Dsubset$covadj,Dsubset[,"CaseControl2"])
```

```
aovres$LDDD_vs_ASD_d[i] = dres$estimate

# TD vs TypSibASD
  pw_comp_res = t.test(D[D$CaseControl2=="TD",y_var],D[D$CaseControl2=="TypSibASD",y_var])
aovres$TD_vs_TypSibASD_t[i] = pw_comp_res$statistic
aovres$TD_vs_TypSibASD_p[i] = pw_comp_res$p.value
Dsubset = subset(D,D$CaseControl2=="TD" | D$CaseControl2=="TypSibASD")
Dsubset$CaseControl2 = factor(Dsubset$CaseControl2)
dres = effsize::cohen.d(Dsubset$covadj,Dsubset[,"CaseControl2"])
aovres$TD_vs_TypSibASD_d[i] = dres$estimate

# TD vs LD_DD
  pw_comp_res = t.test(D[D$CaseControl2=="TD",y_var],D[D$CaseControl2=="LD_DD",y_var])
aovres$TD_vs_LDDD_t[i] = pw_comp_res$statistic
aovres$TD_vs_LDDD_p[i] = pw_comp_res$p.value
Dsubset = subset(D,D$CaseControl2=="TD" | D$CaseControl2=="LD_DD")
Dsubset$CaseControl2 = factor(Dsubset$CaseControl2)
dres = effsize::cohen.d(Dsubset$covadj,Dsubset[,"CaseControl2"])
aovres$TD_vs_LDDD_d[i] = dres$estimate

# TypSibASD vs LD_DD
  pw_comp_res = t.test(D[D$CaseControl2=="TypSibASD",y_var],D[D$CaseControl2=="LD_DD",y_var])
aovres$TypSibASD_vs_LDDD_t[i] = pw_comp_res$statistic
aovres$TypSibASD_vs_LDDD_p[i] = pw_comp_res$p.value
Dsubset = subset(D,D$CaseControl2=="TypSibASD" | D$CaseControl2=="LD_DD")
Dsubset$CaseControl2 = factor(Dsubset$CaseControl2)
dres = effsize::cohen.d(Dsubset$covadj,Dsubset[,"CaseControl2"])
aovres$TypSibASD_vs_LDDD_d[i] = dres$estimate

  # subtype model with data from TD, TD ASDSib, LD/DD, GeoASD and nonGeoASD
  lm_formula = as.formula(sprintf("%s ~ %s + %s + %s",y_var,"subgrp","sex","scan_age"))
  subtype_model = eval(substitute(lm(formula = lm_formula, data = Dexp2_all, na.action = na.omit)))

  # run ANOVA
  res = anova(subtype_model)
  # extract F-stat and pvalue
  fstat = res["subgrp","F value"]
  pval = res["subgrp","Pr(>F)"]
  etasq_res = etasq(subtype_model)
  aovres$df1_subtype[i] = res["subgrp","Df"]
  aovres$df2_subtype[i] = res["Residuals","Df"]
  aovres$fstat_subtype[i] = fstat
  aovres$pval_subtype[i] = pval
  aovres$etasq_subtype[i] = etasq_res["subgrp","Partial eta^2"]

  # construct linear model for subtype model with meanDVARSwavelet as covariate
  lm_formula = as.formula(sprintf("%s ~ %s + %s + %s + %s",y_var,"subgrp","sex","scan_age","meanDVARS
  mod2use = eval(substitute(lm(formula = lm_formula, data = Dtmp, na.action = na.omit)))

  # run ANOVA
  res = anova(mod2use)
  # extract F-stat and pvalue
  fstat = res[1,4]
```

```
    pval = res[1,5]
    aovres$fstat_subtypeDVARScov[i] = fstat
    aovres$pval_subtypeDVARScov[i] = pval


    # compare model with no ASD subtyping to ASD subtype model
    lm_formula = as.formula(sprintf("%s ~ %s + %s +%s",y_var,"CaseControl2","sex","scan_age"))
    notsubtype_model = eval(substitute(lm(formula = lm_formula, data = Dexp2_all, na.action = na.omit)))
    lm_formula = as.formula(sprintf("%s ~ %s + %s +%s",y_var,"subgrp","sex","scan_age"))
subtype_model = eval(substitute(lm(formula = lm_formula, data = Dexp2_all, na.action = na.omit)))

    aovres$AIC_nosubtype[i] = AIC(notsubtype_model)
    aovres$AIC_subtype[i] = AIC(subtype_model)

        if (aovres$AIC_subtype[i]<aovres$AIC_nosubtype[i]){
    aovres$AIC_delta[i] = aovres$AIC_nosubtype[i] - aovres$AIC_subtype[i]
    } else {
      aovres$AIC_delta[i] = aovres$AIC_subtype[i] - aovres$AIC_nosubtype[i]
    }

}
aovres$fdr_all = p.adjust(aovres$pval_all,method="fdr")
aovres$fdr_subtype = p.adjust(aovres$pval_subtype,method="fdr")
aovres$fdr_subtypeDVARScov = p.adjust(aovres$pval_subtypeDVARScov,method="fdr")
write.csv(aovres,file = file.path(resultdir,"casectrl_subtype_allcomps_output.csv"))
aovres[order(aovres$fdr_all),]
```

```
##            df1_all df2_all  Fstat_all     pval_all   etasq_all    fdr_all
## IC02_IC10        3     189 5.98793605 0.0006411647 0.088751689 0.01030673
## IC05_IC10        3     189 6.62772145 0.0002793470 0.099384309 0.01030673
## IC09_IC10        3     189 5.93476452 0.0006871150 0.087604923 0.01030673
## IC04_IC28        3     189 3.59599439 0.0146332362 0.052651846 0.16462391
## IC02_IC26        3     189 3.18349890 0.0250784759 0.043636902 0.17123961
## IC05_IC21        3     189 3.21542015 0.0240560685 0.049726859 0.17123961
## IC05_IC26        3     189 3.13723380 0.0266372722 0.042710392 0.17123961
## IC09_IC21        3     189 2.52563127 0.0588964755 0.045368141 0.33129267
## IC02_IC05        3     189 1.85658005 0.1384172660 0.021808414 0.41475529
## IC05_IC06        3     189 2.19003321 0.0906344498 0.030689382 0.41475529
## IC05_IC09        3     189 1.84314499 0.1407807151 0.029416041 0.41475529
## IC06_IC11        3     189 1.93596121 0.1252125277 0.015507204 0.41475529
## IC06_IC26        3     189 1.84836746 0.1398574293 0.032805613 0.41475529
## IC06_IC28        3     189 1.87143126 0.1358490509 0.024799059 0.41475529
## IC09_IC28        3     189 1.80626863 0.1474685488 0.024874876 0.41475529
## IC11_IC21        3     189 1.82247379 0.1444929199 0.027690262 0.41475529
## IC04_IC06        3     189 1.74098571 0.1600608309 0.027834059 0.42237866
## IC04_IC11        3     189 1.69778394 0.1689514649 0.020529232 0.42237866
## IC09_IC11        3     189 1.51685457 0.2115341143 0.021710412 0.50100185
## IC21_IC26        3     189 1.44706767 0.2305108843 0.021374848 0.51864949
## IC02_IC21        3     189 1.29807422 0.2764377752 0.028022073 0.59236666
## IC02_IC06        3     189 1.03640431 0.3776502418 0.014383703 0.63158570
## IC04_IC26        3     189 1.01936383 0.3852538618 0.021416531 0.63158570
## IC05_IC11        3     189 1.19323168 0.3136269284 0.014316538 0.63158570
## IC05_IC28        3     189 1.00232924 0.3929866559 0.013045065 0.63158570
## IC06_IC10        3     189 1.05942362 0.3675859303 0.011644740 0.63158570
```

```
## IC10_IC11       3      189 1.13857985 0.3347504230 0.017730870 0.63158570
## IC10_IC28       3      189 1.11609063 0.3438048973 0.015417061 0.63158570
## IC10_IC26       3      189 0.89850588 0.4430328726 0.014796062 0.68746480
## IC26_IC28       3      189 0.82825125 0.4798131092 0.016240274 0.71971966
## IC02_IC11       3      189 0.65685439 0.5795915116 0.008983540 0.73694497
## IC02_IC28       3      189 0.77557776 0.5089620687 0.015806378 0.73694497
## IC04_IC09       3      189 0.71724663 0.5428175717 0.006851961 0.73694497
## IC06_IC21       3      189 0.65039133 0.5836287384 0.006872911 0.73694497
## IC11_IC28       3      189 0.64095814 0.5895559748 0.010929312 0.73694497
## IC21_IC28       3      189 0.67193953 0.5702442027 0.012166235 0.73694497
## IC04_IC10       3      189 0.43773028 0.7262613162 0.006929300 0.85798570
## IC09_IC26       3      189 0.41337327 0.7435876074 0.004132000 0.85798570
## IC11_IC26       3      189 0.42425285 0.7358308595 0.006405970 0.85798570
## IC02_IC09       3      189 0.37978526 0.7676849435 0.007936667 0.86364556
## IC10_IC21       3      189 0.29750106 0.8271765680 0.002378506 0.90787672
## IC02_IC04       3      189 0.24917578 0.8618547779 0.001682733 0.91181365
## IC04_IC21       3      189 0.20687689 0.8915511274 0.004277508 0.91181365
## IC06_IC09       3      189 0.21549452 0.8855740320 0.001069455 0.91181365
## IC04_IC05       3      189 0.07011502 0.9758300655 0.001092861 0.97583007
##           df1_subtype df2_subtype fstat_subtype pval_subtype etasq_subtype
## IC02_IC10           4         157     5.2464201 0.0005418251   0.118932212
## IC05_IC10           4         157     5.5304785 0.0003428665   0.126778502
## IC09_IC10           4         157     3.8809450 0.0049204069   0.088662416
## IC04_IC28           4         157     2.5034963 0.0445057263   0.060818378
## IC02_IC26           4         157     1.9292006 0.1081992108   0.043753148
## IC05_IC21           4         157     3.3536872 0.0115052541   0.077341312
## IC05_IC26           4         157     2.7126302 0.0320040841   0.060404262
## IC09_IC21           4         157     1.8141057 0.1287429395   0.047637578
## IC02_IC05           4         157     1.7458510 0.1426004865   0.030382788
## IC05_IC06           4         157     1.9131827 0.1108603245   0.043316172
## IC05_IC09           4         157     1.6319616 0.1688513862   0.041163129
## IC06_IC11           4         157     1.1265917 0.3459943087   0.015881961
## IC06_IC26           4         157     1.1999246 0.3130703643   0.034909283
## IC06_IC28           4         157     0.8208392 0.5136849112   0.017445395
## IC09_IC28           4         157     2.1311358 0.0794498576   0.048756673
## IC11_IC21           4         157     1.5115729 0.2013757077   0.038283935
## IC04_IC06           4         157     1.4016006 0.2359325823   0.035649341
## IC04_IC11           4         157     1.0359784 0.3905237257   0.021072844
## IC09_IC11           4         157     1.7578038 0.1400772618   0.043013462
## IC21_IC26           4         157     0.8969769 0.4673138878   0.022531226
## IC02_IC21           4         157     1.6080595 0.1748983510   0.038837899
## IC02_IC06           4         157     1.5217349 0.1984247461   0.033316960
## IC04_IC26           4         157     0.9068775 0.4615046722   0.028550142
## IC05_IC11           4         157     1.5935604 0.1786626285   0.037748710
## IC05_IC28           4         157     0.8854963 0.4741145299   0.018885306
## IC06_IC10           4         157     1.0853656 0.3657184250   0.022538450
## IC10_IC11           4         157     1.7382713 0.1442224454   0.042533110
## IC10_IC28           4         157     0.6966364 0.5953562714   0.016628342
## IC10_IC26           4         157     0.7333331 0.5705063578   0.019385136
## IC26_IC28           4         157     0.9858475 0.4170295825   0.026408372
## IC02_IC11           4         157     1.2798160 0.2802431531   0.028770090
## IC02_IC28           4         157     0.7576068 0.5543875447   0.024800645
## IC04_IC09           4         157     0.7444058 0.5631210148   0.009891132
## IC06_IC21           4         157     0.5585791 0.6930588063   0.010575431
```

```
## IC11_IC28           4         157     0.3923445 0.8139063826   0.011003304
## IC21_IC28           4         157     0.5311015 0.7130482019   0.015920087
## IC04_IC10           4         157     0.3050935 0.8742362886   0.007325977
## IC09_IC26           4         157     0.7120184 0.5848718872   0.011768388
## IC11_IC26           4         157     0.3430069 0.8485742004   0.008254619
## IC02_IC09           4         157     0.2612827 0.9023892472   0.008277410
## IC10_IC21           4         157     0.1634915 0.9565534730   0.004481284
## IC02_IC04           4         157     0.5549054 0.6957250661   0.010238600
## IC04_IC21           4         157     0.1650124 0.9558304150   0.003947705
## IC06_IC09           4         157     0.3525796 0.8419446820   0.006685714
## IC04_IC05           4         157     0.4283900 0.7879751850   0.009811804
##           fdr_subtype fstat_subtypeDVARScov pval_subtypeDVARScov
## IC02_IC10  0.01219106             5.2814873         0.0005136610
## IC05_IC10  0.01219106             5.5355006         0.0003412898
## IC09_IC10  0.07380610             4.3649282         0.0022554104
## IC04_IC28  0.33379295             2.5175334         0.0435626902
## IC02_IC26  0.49923154             1.9208336         0.1096178143
## IC05_IC21  0.12943411             3.3583690         0.0114329436
## IC05_IC26  0.28803676             2.7262390         0.0313458541
## IC09_IC21  0.49923154             1.8075170         0.1300603941
## IC02_IC05  0.49923154             1.7588058         0.1399046368
## IC05_IC06  0.49923154             1.9203919         0.1096912562
## IC05_IC09  0.50248864             1.6216400         0.1714754113
## IC06_IC11  0.70771563             1.1348350         0.3421831974
## IC06_IC26  0.67086507             1.2018144         0.3122869006
## IC06_IC28  0.78797154             0.8190008         0.5148542412
## IC09_IC28  0.49923154             2.1176564         0.0811498802
## IC11_IC21  0.50343927             1.5200899         0.1989356930
## IC04_IC06  0.55878770             1.3926800         0.2389890735
## IC04_IC11  0.73223199             1.0593019         0.3786723026
## IC09_IC11  0.49923154             1.7560695         0.1404777629
## IC21_IC26  0.76196978             0.8922855         0.4701012885
## IC02_IC21  0.50248864             1.6178789         0.1724271902
## IC02_IC06  0.50343927             1.5120826         0.2012625685
## IC04_IC26  0.76196978             0.9103165         0.4595164643
## IC05_IC11  0.50248864             1.5862200         0.1806330476
## IC05_IC28  0.76196978             0.9015612         0.4646347662
## IC06_IC10  0.71553605             1.0801808         0.3682868205
## IC10_IC11  0.49923154             1.7274929         0.1465954745
## IC10_IC28  0.78797154             0.6942824         0.5969766807
## IC10_IC26  0.78797154             0.7301442         0.5726523674
## IC26_IC28  0.75065325             0.9795703         0.4204638333
## IC02_IC11  0.63054709             1.2760942         0.2817355009
## IC02_IC28  0.78797154             0.7752356         0.5428601305
## IC04_IC09  0.78797154             0.7417156         0.5649200816
## IC06_IC21  0.86722079             0.5569532         0.6942404785
## IC11_IC28  0.93136193             0.4146984         0.7978669794
## IC21_IC28  0.86722079             0.5277312         0.7155072947
## IC04_IC10  0.93668174             0.3064107         0.8733595741
## IC09_IC26  0.78797154             0.7346309         0.5696472893
## IC11_IC26  0.93136193             0.3870390         0.8176849288
## IC02_IC09  0.94436084             0.2622887         0.9017614498
## IC10_IC21  0.95655347             0.1654180         0.9556346967
## IC02_IC04  0.86722079             0.5865714         0.6728285255
```

```
## IC04_IC21  0.95655347            0.1687559         0.9540297381
## IC06_IC09  0.93136193            0.3617354         0.8355532148
## IC04_IC05  0.93136193            0.4270739         0.7889264394
##           fdr_subtypeDVARScov AIC_nosubtype AIC_subtype AIC_delta
## IC02_IC10          0.01155737    -47.601616  -51.329649 3.7280333
## IC05_IC10          0.01155737    -89.543146  -87.727228 1.8159180
## IC09_IC10          0.03383116     19.207204   20.297832 1.0906282
## IC04_IC28          0.32672018   -106.458925 -104.742943 1.7159814
## IC02_IC26          0.50315642   -169.978482 -168.124064 1.8544183
## IC05_IC21          0.12862062    -78.592384  -79.526365 0.9339807
## IC05_IC26          0.28211269   -255.830825 -254.253969 1.5768561
## IC09_IC21          0.50315642    -60.658256  -58.686249 1.9720067
## IC02_IC05          0.50315642     82.308288   82.882327 0.5740396
## IC05_IC06          0.50315642     95.057050   97.054316 1.9972661
## IC05_IC09          0.50315642    -52.451869  -50.975913 1.4759564
## IC06_IC11          0.69992018    120.108604  122.103260 1.9946554
## IC06_IC26          0.66918622   -223.189912 -222.636215 0.5536971
## IC06_IC28          0.78088959    -85.724944  -83.838817 1.8861263
## IC09_IC28          0.50315642    -42.049236  -40.331381 1.7178552
## IC11_IC21          0.50315642    -57.601587  -56.778775 0.8228122
## IC04_IC06          0.56602675    -22.458550  -20.787183 1.6713667
## IC04_IC11          0.71001057      2.124502    2.899205 0.7747028
## IC09_IC11          0.50315642     72.441559   74.136379 1.6948199
## IC21_IC26          0.75551993   -254.287124 -252.355442 1.9316818
## IC02_IC21          0.50315642     39.354436   38.471250 0.8831861
## IC02_IC06          0.50315642    148.702901  147.494989 1.2079123
## IC04_IC26          0.75551993   -236.735500 -235.222713 1.5127872
## IC05_IC11          0.50315642     99.736313   99.372038 0.3642751
## IC05_IC28          0.75551993   -100.826639  -98.978566 1.8480723
## IC06_IC10          0.71001057    -84.117563  -83.221939 0.8956241
## IC10_IC11          0.50315642     -8.805833  -12.076756 3.2709235
## IC10_IC28          0.79011619    -22.210481  -20.739908 1.4705726
## IC10_IC26          0.78088959   -266.070825 -264.149009 1.9218155
## IC26_IC28          0.75551993   -217.477024 -216.549457 0.9275674
## IC02_IC11          0.63390488    106.754312  107.018529 0.2642172
## IC02_IC28          0.78088959      2.609779    4.339769 1.7299894
## IC04_IC09          0.78088959    -73.155964  -71.286644 1.8693194
## IC06_IC21          0.86780060    -65.711022  -63.886750 1.8242726
## IC11_IC28          0.91707060      4.601489    6.592426 1.9909366
## IC21_IC28          0.87021157    -52.118375  -50.233033 1.8853422
## IC04_IC10          0.93574240    -35.998229  -34.053934 1.9442944
## IC09_IC26          0.78088959   -137.025822 -136.298689 0.7271324
## IC11_IC26          0.91707060   -143.994361 -142.248185 1.7461754
## IC02_IC09          0.94370384    122.941268  124.940886 1.9996182
## IC10_IC21          0.95563470    -52.959008  -51.248201 1.7108066
## IC02_IC04          0.86506525      3.619161    4.291625 0.6724636
## IC04_IC21          0.95563470    -35.278770  -33.642334 1.6364354
## IC06_IC09          0.91707060     13.332289   14.511621 1.1793315
## IC04_IC05          0.91707060    -61.822138  -60.855373 0.9667645
##           TD_vs_ASD_t  TD_vs_ASD_p  TD_vs_ASD_d TypSibASD_vs_ASD_t
## IC02_IC10  4.08833950 7.685064e-05 -0.661803649         2.52891112
## IC05_IC10  2.95087394 3.934468e-03 -0.521973563         2.08056846
## IC09_IC10  3.60817859 4.926817e-04 -0.662503945         1.60514818
## IC04_IC28 -1.76606802 8.025688e-02  0.309871665        -1.24243412
```

```
## IC02_IC26  0.39331789 6.949485e-01 -0.048477894        3.25085384
## IC05_IC21  2.18160767 3.191754e-02 -0.426570460       -0.93556094
## IC05_IC26 -0.04949474 9.606415e-01  0.007903278       -1.02296194
## IC09_IC21  1.12775990 2.617336e-01 -0.209555217        0.89907726
## IC02_IC05 -0.76411785 4.466252e-01  0.188520525       -1.50390486
## IC05_IC06  1.90562181 5.906742e-02 -0.295264273       -0.36031634
## IC05_IC09 -0.69732069 4.872455e-01  0.106570575        1.86430163
## IC06_IC11 -1.87524568 6.332053e-02  0.264260958       -2.12850956
## IC06_IC26  1.90066040 6.022430e-02 -0.353510307        1.75405272
## IC06_IC28 -1.81547459 7.227499e-02  0.303244712       -0.47425973
## IC09_IC28 -2.34545668 2.045466e-02  0.358652502       -0.88269113
## IC11_IC21 -0.46396030 6.437510e-01  0.074677776        0.62854280
## IC04_IC06 -1.20258706 2.315526e-01  0.219546493       -1.98821636
## IC04_IC11  1.14516866 2.548468e-01 -0.203050073        1.08230532
## IC09_IC11  1.91670998 5.815231e-02 -0.318727711        1.08027707
## IC21_IC26 -0.44160253 6.597511e-01  0.055661163        0.55691110
## IC02_IC21  1.91073845 5.852409e-02 -0.393557606       -0.07518491
## IC02_IC06  0.68804259 4.929031e-01 -0.127182434       -0.67397697
## IC04_IC26  1.25194125 2.134916e-01 -0.250988823        1.11379352
## IC05_IC11 -1.62316358 1.073264e-01  0.241994646       -1.71089498
## IC05_IC28 -1.35122981 1.795440e-01  0.199554025       -0.07819236
## IC06_IC10  0.49466501 6.217695e-01 -0.033272182        1.51861052
## IC10_IC11  1.34484249 1.815175e-01 -0.228129641        0.88711047
## IC10_IC28  0.43487622 6.644434e-01 -0.078969602        1.65381179
## IC10_IC26  1.34616600 1.813330e-01 -0.230728690        0.22047367
## IC26_IC28  1.42631969 1.571030e-01 -0.283825039        0.45501387
## IC02_IC11 -0.51021690 6.110792e-01  0.058197487       -0.74006594
## IC02_IC28  0.91101944 3.642256e-01 -0.148038199       -0.15265949
## IC04_IC09 -0.38594411 7.004998e-01  0.035614619       -1.22668061
## IC06_IC21 -0.81767650 4.155345e-01  0.143788690       -0.18806540
## IC11_IC28  1.34252287 1.821538e-01 -0.232978904       -0.14946364
## IC21_IC28 -1.30704894 1.944823e-01  0.245191838       -0.46818004
## IC04_IC10  0.60462558 5.467810e-01 -0.145824608       -0.59591614
## IC09_IC26  0.54436635 5.871097e-01 -0.051419753        0.91273213
## IC11_IC26 -0.64063517 5.231966e-01  0.115805958        0.37952573
## IC02_IC09 -0.72167774 4.721889e-01  0.153966077       -0.53188793
## IC10_IC21 -0.73670077 4.628613e-01  0.090885144       -0.73907481
## IC02_IC04 -0.27227874 7.860045e-01 -0.002581072       -0.97387047
## IC04_IC21  0.37934618 7.051324e-01 -0.088797337        0.42029412
## IC06_IC09 -0.32474611 7.458428e-01  0.060457096        0.44989804
## IC04_IC05 -0.16933350 8.658639e-01  0.045355647        0.26692174
##           TypSibASD_vs_ASD_p TypSibASD_vs_ASD_d LDDD_vs_ASD_t
## IC02_IC10        0.018893459        -0.60549851   1.664981303
## IC05_IC10        0.051204745        -0.64407807   2.474492564
## IC09_IC10        0.122955211        -0.40768683   2.276207344
## IC04_IC28        0.230040294         0.40910237  -2.410397306
## IC02_IC26        0.004109426        -0.85183487   0.738127855
## IC05_IC21        0.361434072         0.27613886   1.727497912
## IC05_IC26        0.317428302         0.21264252  -2.444897337
## IC09_IC21        0.377664368        -0.28308118   2.504603484
## IC02_IC05        0.150099443         0.48781992  -1.340594150
## IC05_IC06        0.722658809         0.18702522   2.172674299
## IC05_IC09        0.077204675        -0.52389168  -1.273976165
## IC06_IC11        0.043376527         0.26219332  -1.010866622
```

```
## IC06_IC26       0.094077896       -0.47822136  0.064676883
## IC06_IC28       0.640219736        0.06354073 -1.750154604
## IC09_IC28       0.387638366        0.17989116 -1.102535802
## IC11_IC21       0.537709201       -0.22043316  1.822353970
## IC04_IC06       0.061297514        0.56384079 -0.070216971
## IC04_IC11       0.292688188       -0.27258581  1.957693543
## IC09_IC11       0.290988913       -0.19888308  0.034663185
## IC21_IC26       0.583504369       -0.19959086 -1.428512006
## IC02_IC21       0.940753947       -0.15054231  0.261587360
## IC02_IC06       0.507980049        0.19522240  1.278363544
## IC04_IC26       0.280217553       -0.44176580 -0.101391205
## IC05_IC11       0.098101467        0.27463639 -0.716777911
## IC05_IC28       0.938489913       -0.02498218  0.671441994
## IC06_IC10       0.145240410       -0.37994367 -0.334485807
## IC10_IC11       0.385761083       -0.24973969  1.674121074
## IC10_IC28       0.113873588       -0.43104971  0.889749018
## IC10_IC26       0.827123262       -0.06648209  0.933366138
## IC26_IC28       0.653949571       -0.15740136  0.428707567
## IC02_IC11       0.468298368        0.15798090 -1.402569634
## IC02_IC28       0.880346519        0.09191791 -1.141456408
## IC04_IC09       0.236284374        0.29590365  0.112568695
## IC06_IC21       0.852880588        0.13267460  1.328837538
## IC11_IC28       0.882769719        0.04765042  0.394027054
## IC21_IC28       0.644552527        0.09207453  0.012233928
## IC04_IC10       0.558359344        0.10393199  0.536532437
## IC09_IC26       0.372001463       -0.22115212 -0.295386495
## IC11_IC26       0.708822625       -0.15234046 -0.756973721
## IC02_IC09       0.601581155        0.23834872 -0.661078783
## IC10_IC21       0.468125748        0.11862669 -0.033595333
## IC02_IC04       0.341251748        0.16008024 -0.007866081
## IC04_IC21       0.679001987       -0.16532429  0.749491351
## IC06_IC09       0.656759540       -0.01121969  0.624423151
## IC04_IC05       0.792351858       -0.04112908  0.272581163
##           LDDD_vs_ASD_p LDDD_vs_ASD_d TD_vs_TypSibASD_t TD_vs_TypSibASD_p
## IC02_IC10    0.11421669  -0.513481887        0.21820131        0.82899786
## IC05_IC10    0.02550088  -1.055820607       -0.25473259        0.80105022
## IC09_IC10    0.03574124  -0.633627226        1.09186292        0.28280785
## IC04_IC28    0.02814014   0.815223789        0.31394233        0.75661920
## IC02_IC26    0.47160179  -0.308303306       -2.73761457        0.01085497
## IC05_IC21    0.10102532  -0.422175267        2.17545566        0.03785921
## IC05_IC26    0.02648656   0.848796230        0.80832898        0.42360484
## IC09_IC21    0.02219477  -0.769672822       -0.05378268        0.95747040
## IC02_IC05    0.19833317   0.326487706        1.03012554        0.31402864
## IC05_IC06    0.04145219  -0.400138990        1.30414213        0.20652086
## IC05_IC09    0.21551330   0.258303636       -2.13723402        0.04174990
## IC06_IC11    0.32440801   0.106926074        0.58562082        0.56218451
## IC06_IC26    0.94913575  -0.019537513       -0.34455705        0.73288266
## IC06_IC28    0.09738586   0.431471170       -0.72520080        0.47443052
## IC09_IC28    0.28509393   0.255145793       -0.43654949        0.66680186
## IC11_IC21    0.08665476  -0.610152495       -0.82499411        0.41802720
## IC04_IC06    0.94483857  -0.043010476        1.27589466        0.21537305
## IC04_IC11    0.06616865  -0.484584063       -0.35830142        0.72318772
## IC09_IC11    0.97268223   0.019557453        0.46136210        0.64733747
## IC21_IC26    0.17273049   0.501106258       -0.79442073        0.43315542
```

```
## IC02_IC21    0.79675794  -0.151059255       1.30704722        0.20229281
## IC02_IC06    0.21826748  -0.331443623       1.04788126        0.30450639
## IC04_IC26    0.92027720   0.003682787      -0.43704057        0.66641475
## IC05_IC11    0.48249312   0.180912221       0.25743042        0.79824681
## IC05_IC28    0.51123661  -0.220245614      -0.68881003        0.49755147
## IC06_IC10    0.74157201   0.050041350      -1.19681625        0.24410639
## IC10_IC11    0.10918760  -0.392026709      -0.04835369        0.96182447
## IC10_IC28    0.38570354  -0.231220996      -1.34257728        0.19251562
## IC10_IC26    0.36394525  -0.334417336       0.91530710        0.36496532
## IC26_IC28    0.67185522  -0.058327783       0.52254239        0.60512798
## IC02_IC11    0.17703191   0.369359246       0.37745854        0.70888506
## IC02_IC28    0.26734552   0.348597433       0.59589742        0.55765203
## IC04_IC09    0.91171295  -0.119072715       0.92423307        0.36459961
## IC06_IC21    0.19537991  -0.137007642      -0.27418605        0.78629469
## IC11_IC28    0.69862137  -0.091413704       0.86424838        0.39660802
## IC21_IC28    0.99039110  -0.064479421      -0.47610974        0.63718566
## IC04_IC10    0.59871224  -0.169574791       0.89441218        0.38001646
## IC09_IC26    0.77139183   0.050474737      -0.58794193        0.56257756
## IC11_IC26    0.45911224   0.160963317      -0.67218032        0.50873015
## IC02_IC09    0.51775885   0.227523961       0.17462513        0.86307716
## IC10_IC21    0.97352880  -0.012527263       0.23486745        0.81614577
## IC02_IC04    0.99381778   0.015509251       0.67900514        0.50205163
## IC04_IC21    0.46241192  -0.184969188      -0.20261049        0.84130434
## IC06_IC09    0.53874527  -0.037002235      -0.66544372        0.51211048
## IC04_IC05    0.78831999  -0.072349239      -0.34791064        0.73084647
##           TD_vs_TypSibASD_d TD_vs_LDDD_t TD_vs_LDDD_p TD_vs_LDDD_d
## IC02_IC10       0.045881596   0.27178664   0.78883297   0.09952879
## IC05_IC10      -0.111054478  -1.31878303   0.20469367  -0.40106368
## IC09_IC10       0.276270073  -0.05346514   0.95781677   0.04706909
## IC04_IC28       0.105710807   1.52486475   0.14402146   0.42206586
## IC02_IC26      -0.728931363  -0.56363663   0.58018409  -0.17938025
## IC05_IC21       0.576537989  -0.06304070   0.95015113   0.06043328
## IC05_IC26       0.158275176   2.23996165   0.03585833   0.62728392
## IC09_IC21      -0.082207199  -1.80919219   0.08509324  -0.57209348
## IC02_IC05       0.305421357   0.89779818   0.37992371   0.12077041
## IC05_IC06       0.518267195  -0.82760205   0.41582500  -0.13203292
## IC05_IC09      -0.583891361   0.57721033   0.56740562   0.17678799
## IC06_IC11      -0.001113726  -0.21215939   0.83376980  -0.16108141
## IC06_IC26      -0.109945115   1.07922666   0.29105832   0.33860106
## IC06_IC28      -0.241854456   0.64537580   0.52537973   0.12234669
## IC09_IC28      -0.194753260  -0.03805877   0.97005479  -0.07446247
## IC11_IC21      -0.256199894  -1.94384435   0.06544994  -0.56856622
## IC04_IC06       0.375046605  -0.51524959   0.61229707  -0.23110192
## IC04_IC11      -0.063536896  -1.16977868   0.25405227  -0.26577922
## IC09_IC11       0.130924733   1.30553348   0.20150639   0.38972677
## IC21_IC26      -0.239325538   1.18361533   0.25165068   0.33053421
## IC02_IC21       0.246002295   0.69657988   0.49428537   0.21781471
## IC02_IC06       0.333289844  -0.87225654   0.39331415  -0.19162534
## IC04_IC26      -0.176567701   0.90950072   0.37107779   0.28663189
## IC05_IC11       0.032269707  -0.24721816   0.80702658  -0.05705982
## IC05_IC28      -0.215253825  -1.28893756   0.21250174  -0.34860217
## IC06_IC10      -0.374887932   0.62230457   0.53972513   0.10223357
## IC10_IC11      -0.019624823  -0.62613797   0.53633179  -0.18262577
## IC10_IC28      -0.386489049  -0.64064222   0.52914984  -0.15244062
```

```
## IC10_IC26       0.172640750  -0.19441887   0.84773528  -0.08748031
## IC26_IC28       0.127646675   0.80196654   0.42698966   0.31078102
## IC02_IC11       0.087226429   0.93540737   0.35776728   0.31035875
## IC02_IC28       0.238549603   1.64824439   0.11205711   0.57484394
## IC04_IC09       0.209516600  -0.32021737   0.75155506  -0.13664995
## IC06_IC21      -0.012510941  -1.79098689   0.08027450  -0.40027637
## IC11_IC28       0.286156893   0.23243656   0.81873252   0.11263063
## IC21_IC28      -0.150328998  -0.62492373   0.53913084  -0.23156097
## IC04_IC10       0.242991514  -0.21032913   0.83552166  -0.02192483
## IC09_IC26      -0.196878466   0.51931913   0.61018418   0.08211436
## IC11_IC26      -0.248182072   0.35317004   0.72727723   0.04158209
## IC02_IC09       0.080519963   0.27920145   0.78297121   0.06309760
## IC10_IC21       0.028944918  -0.45082605   0.65585406  -0.12095542
## IC02_IC04       0.151720911  -0.13336492   0.89515493   0.01539938
## IC04_IC21      -0.083115172  -0.47120202   0.64172971  -0.10866480
## IC06_IC09      -0.091000010  -0.83399185   0.41332843  -0.12547637
## IC04_IC05      -0.083602806  -0.35136145   0.72859483  -0.11459359
##           TypSibASD_vs_LDDD_t TypSibASD_vs_LDDD_p TypSibASD_vs_LDDD_d
## IC02_IC10          0.10254260          0.91915506        -0.070864696
## IC05_IC10         -1.04240230          0.30805908         0.368247201
## IC09_IC10         -0.86341158          0.39568304         0.254291759
## IC04_IC28          0.99211915          0.32950883        -0.335748988
## IC02_IC26          1.06604454          0.29776164        -0.363695815
## IC05_IC21         -1.97382473          0.05802930         0.666614220
## IC05_IC26          1.65407849          0.11224080        -0.600685885
## IC09_IC21         -1.59587579          0.12285096         0.564350100
## IC02_IC05         -0.07231240          0.94285325         0.125122105
## IC05_IC06         -1.71092620          0.09864977         0.614781381
## IC05_IC09          2.46506755          0.02035081        -0.896836915
## IC06_IC11         -0.64069240          0.52710263         0.175754238
## IC06_IC26          1.20964896          0.23643065        -0.482539092
## IC06_IC28          1.10258667          0.27971819        -0.376384806
## IC09_IC28          0.27734350          0.78356546        -0.080653820
## IC11_IC21         -0.88267335          0.38468221         0.311173663
## IC04_IC06         -1.35300216          0.18673381         0.559659060
## IC04_IC11         -0.64947667          0.52114545         0.202520485
## IC09_IC11          0.79061721          0.43567808        -0.265532296
## IC21_IC26          1.57740024          0.12874581        -0.590307665
## IC02_IC21         -0.27186917          0.78789155         0.003514785
## IC02_IC06         -1.50222101          0.14453396         0.527827593
## IC04_IC26          1.01576740          0.31893831        -0.437606250
## IC05_IC11         -0.41292011          0.68338250         0.100603629
## IC05_IC28         -0.59446892          0.55706035         0.171861894
## IC06_IC10          1.47104798          0.15225810        -0.492020033
## IC10_IC11         -0.43344264          0.66798763         0.148244438
## IC10_IC28          0.46273367          0.64710360        -0.193950166
## IC10_IC26         -0.75753909          0.45728619         0.292284124
## IC26_IC28          0.12008332          0.90532677        -0.120941141
## IC02_IC11          0.41656078          0.68008795        -0.205375668
## IC02_IC28          0.60757373          0.54852963        -0.246786678
## IC04_IC09         -1.02081219          0.31581357         0.341797730
## IC06_IC21         -0.91276694          0.37089222         0.281821449
## IC11_IC28         -0.41738406          0.67962582         0.124762408
## IC21_IC28         -0.27599135          0.78497781         0.130884775
```

```
## IC04_IC10         -0.83592461          0.41024656        0.254810216
## IC09_IC26          0.82039589          0.41959711       -0.245659414
## IC11_IC26          0.81878745          0.41970588       -0.264243925
## IC02_IC09          0.07247327          0.94272303        0.012120946
## IC10_IC21         -0.56136811          0.57886401        0.151234570
## IC02_IC04         -0.59734043          0.55559652        0.138843795
## IC04_IC21         -0.16893059          0.86705838        0.015784334
## IC06_IC09         -0.15153229          0.88061039        0.036324839
## IC04_IC05         -0.01119008          0.99114883        0.031048945
```

Write out results for significant connections

```r
sig_res = vars2use[aovres$fdr_all<=fdr_thresh]
sig_res
```

```
## [1] "IC02_IC10" "IC05_IC10" "IC09_IC10"
```

```r
Dexp2 = subset(D, is.element(D$subgrp,c("GeoASD","nonGeoASD","LD_DD","TypSibASD","TD")),
               select = 1:ncol(D))
Dexp2$subgrp = factor(Dexp2$subgrp)
Dexp2$Case_vs_nonASD = NA
Dexp2$Case_vs_nonASD[is.element(Dexp2$subgrp,c("LD_DD","TypSibASD"))] = "nonASD"
Dexp2$Case_vs_nonASD[is.element(Dexp2$subgrp,c("GeoASD","nonGeoASD"))] = "ASD"
Dexp2$Case_vs_nonASD = factor(Dexp2$Case_vs_nonASD)

# set up comparisons to run
comp1 = c("GeoASD","nonGeoASD","LD_DD","TypSibASD")
comp2 = list(comp = c("nonGeoASD","LD_DD","TypSibASD","TD"),
             comp = c("LD_DD","TypSibASD","TD"),
             comp = c("TypSibASD","TD"),
             comp = c("TD"))

# set up data frame for storing multiple comparison results
colnames2use = c("compName","tstat","pvalue","fdr_q","cohensd","np.W","np.pvalue","np.fdr_q")
mcompres = data.frame(matrix(nrow = 10,ncol = length(colnames2use)))
colnames(mcompres) = colnames2use

# set up stuff to plotting effect size matrix
dmat_idx = cbind(c(1,1,1,1,2,2,2,3,3,4), c(2,3,4,5,3,4,5,4,5,5))
dMat_grpLabels = c("GeoASD","nonGeoASD","LD_DD","TDSibASD","TD")

# more stuff for plots
yLimits = list(ylim = c(-0.4,1),
               ylim = c(-0.6,1),
               ylim = c(-0.3,1.6))

# loop over number of significant connections
for (i in 1:length(sig_res)) {
    y_var = sig_res[i]

    # model using only subgrp
    lm_formula = as.formula(sprintf("%s ~ %s + %s + %s",y_var,"subgrp","sex","scan_age"))
    mod2use = eval(substitute(lm(formula = lm_formula, data = Dexp2, na.action = na.omit)))
```

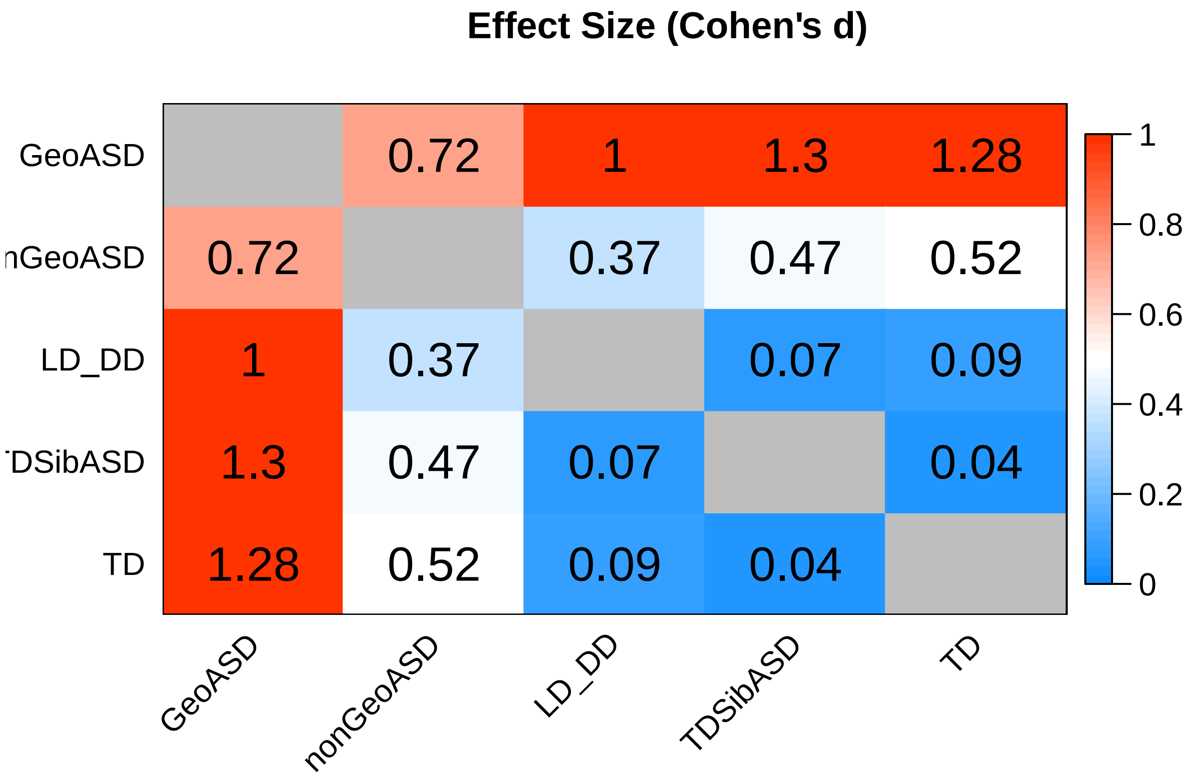
```
subtype_model = mod2use
subtype_formula = lm_formula

# anova on model using only subgrp
  sigaovres = anova(subtype_model)

  # model using case-control status
  lm_formula = as.formula(sprintf("%s ~ %s + %s + %s",y_var,"CaseControl2","sex","scan_age"))
  mod2use = eval(substitute(lm(formula = lm_formula, data = Dexp2, na.action = na.omit)))
cc_model = mod2use
cc_formula = lm_formula

  # compare subtype vs case-control models with AIC
  subtype_vs_cc_model_compeval = rbind(AIC(subtype_model),AIC(cc_model))
  rownames(subtype_vs_cc_model_compeval) = c(subtype_formula,cc_formula)
  colnames(subtype_vs_cc_model_compeval) = c("AIC")

  # remove sex and scan age for effect size computation
  covname2use = c("sexM","scan_age")
  beta1 = mod2use$coefficients[covname2use, drop = FALSE]
  beta1[is.na(beta1)] = 0
  full_model = model.matrix(~0+as.factor(subgrp) + as.factor(sex) + scan_age, data = Dexp2)
  colnames(full_model) = c("GeoASD","nonGeoASD","LD_DD","TypSibASD","TD","sex","scan_age")
  covname2use = c("sex","scan_age")
  Dexp2$covadj = as.numeric(t(Dexp2[,y_var] - beta1 %*% t(full_model[,covname2use])))

  # multiple comparisons on subgroup
  compCount = 0
  for (ic1 in 1:length(comp1)) {
      for (ic2 in 1:length(comp2[ic1]$comp)) {
          compCount = compCount + 1
          compname1 = comp1[ic1]
          compname2 = comp2[ic1]$comp[ic2]
          Dcomp = subset(Dexp2, Dexp2$subgrp==compname1 | Dexp2$subgrp==compname2,
                         select = c("subgrp",y_var,"covadj"))
          Dcomp$subgrp = factor(Dcomp$subgrp)

          tres = tres = t.test(Dcomp[Dcomp$subgrp==compname1,y_var],Dcomp[Dcomp$subgrp==compname2,y_va
          dres = effsize::cohen.d(Dcomp$covadj,Dcomp[,"subgrp"])

          # Added Mann-Whitney U test --------------------------
          npres = wilcox.test(Dcomp[Dcomp$subgrp==compname1,y_var],Dcomp[Dcomp$subgrp==compname2,y_va
          mcompres$np.W[compCount] = npres$statistic
          mcompres$np.pvalue[compCount] = npres$p.value
          # ---------------------------------------------------

          mcompres$compName[compCount] = sprintf("%s vs %s",compname1,compname2)
          mcompres$tstat[compCount] = tres$statistic
          mcompres$pvalue[compCount] = tres$p.value
          mcompres$cohensd[compCount] = dres$estimate
      }#for (ic2 in 1:length(comp2[ic1]$comp))
  }#for (ic1 in 1:length(comp1))
```

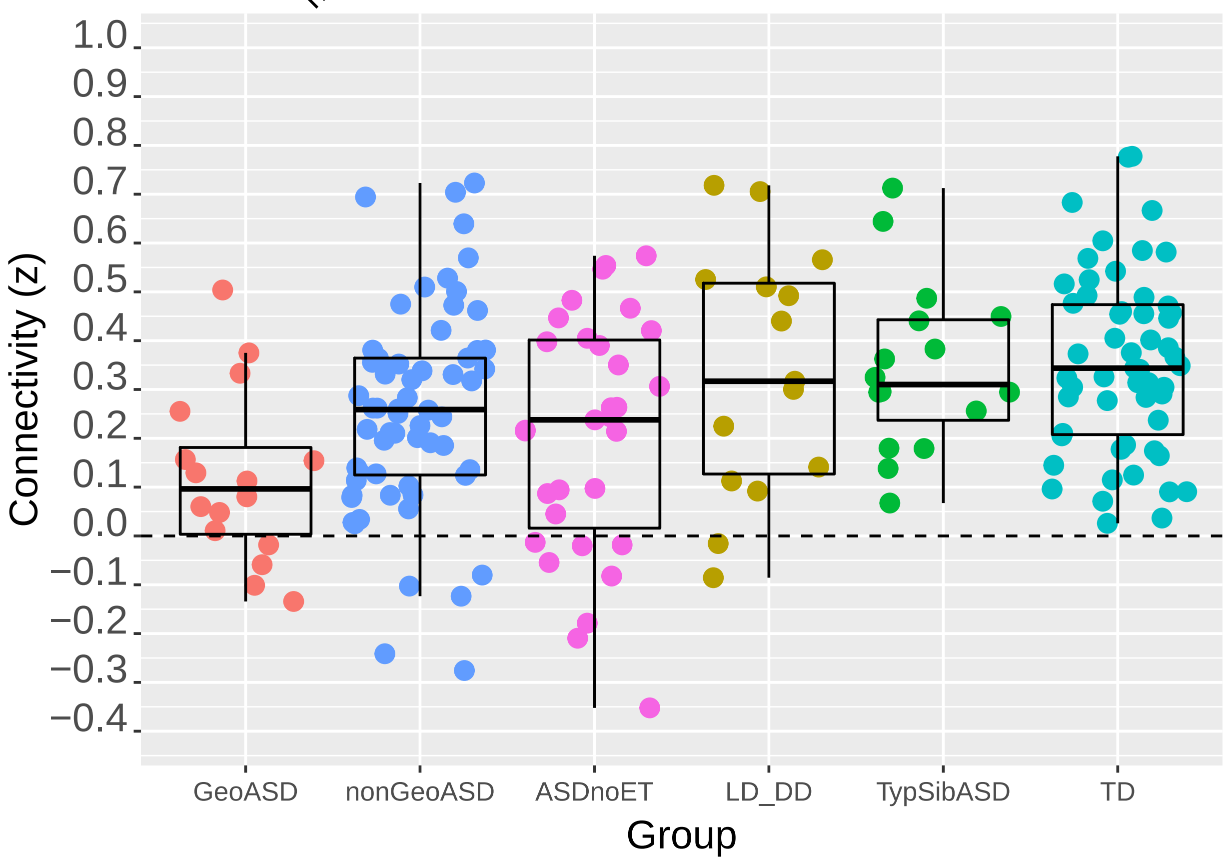
```
  # compute FDR
  mcompres$fdr_q = p.adjust(mcompres$pvalue,method = "fdr")
  # write.csv(mcompres, file = file.path(resultdir,sprintf("mcompres_%s.csv",y_var)))

  # compute FDR on non-parametric tests -------------------
  mcompres$np.fdr_q = p.adjust(mcompres$np.pvalue,method = "fdr")
  write.csv(mcompres, file = file.path(resultdir,sprintf("mcompres_%s.csv",y_var)))
# ------------------------------------------------------

  # RDOC model - Correlation with FixGeo across all groups
  rdoc_formula = as.formula(sprintf("%s ~ %s + %s + %s",y_var,"FixGeo","sex","scan_age"))
  rdoc_model = eval(substitute(lm(formula = rdoc_formula, data = Dexp2, na.action = na.omit)))
  subgrpNOGEOFIX_formula = as.formula(sprintf("%s ~ %s + %s + %s",y_var,"subgrp","sex","scan_age"))
  subgrpNOGEOFIX_model = eval(substitute(lm(formula = subgrpNOGEOFIX_formula, data = Dexp2, na.action

  # compare RDOC vs subgrp models with AIC
  model_compeval = rbind(AIC(subgrpNOGEOFIX_model),AIC(rdoc_model))
  rownames(model_compeval) = c(subgrpNOGEOFIX_formula,rdoc_formula)
  colnames(model_compeval) = c("AIC")

  # print results to screen
  print(sprintf("%s: ANOVA on stratified model", y_var))
  print(sigaovres)
  print(etasq(subtype_model))

  print(sprintf("%s: Statistics for each pairwise comparison", y_var))
  print(mcompres)

  print(sprintf("%s: Model comparison for subtype vs case-control models", y_var))
  print(subtype_vs_cc_model_compeval)

  print(sprintf("%s: Model comparison for RDOC vs subtype models", y_var))
  print(model_compeval)

  # make effect size matrix as heatmap figure
  ngrp = length(unique(Dexp2$subgrp))
  dMat = data.frame(matrix(nrow = ngrp, ncol=ngrp))
  dMat[diag(x = 1,nrow=ngrp,ncol=ngrp)==1] = NA
  for (ires in 1:dim(dmat_idx)[1]) {
      dMat[dmat_idx[ires,1],dmat_idx[ires,2]] = abs(mcompres$cohensd[ires])
      dMat[dmat_idx[ires,2],dmat_idx[ires,1]] = abs(mcompres$cohensd[ires])
  }# for (ires in 1:dim(dmat_idx)[1])
  rownames(dMat) = dMat_grpLabels
  colnames(dMat) = dMat_grpLabels

  #plot the matrix as a heatmap using the labeledHeatmap function from WGCNA
  # WGCNA::sizeGrWindow(10,10)
  # pdf(file = file.path(plotdir,sprintf("FC_%s_effectSize_plot.pdf",y_var)))
  # par(mar = c(6, 8.5, 3, 3))
  WGCNA::labeledHeatmap(Matrix = dMat,
      xLabels = rownames(dMat), yLabels = colnames(dMat),
      ySymbols = NULL, colorLabels = FALSE,
      colors = WGCNA::blueWhiteRed(50), textMatrix = round(dMat,digits=2),
```

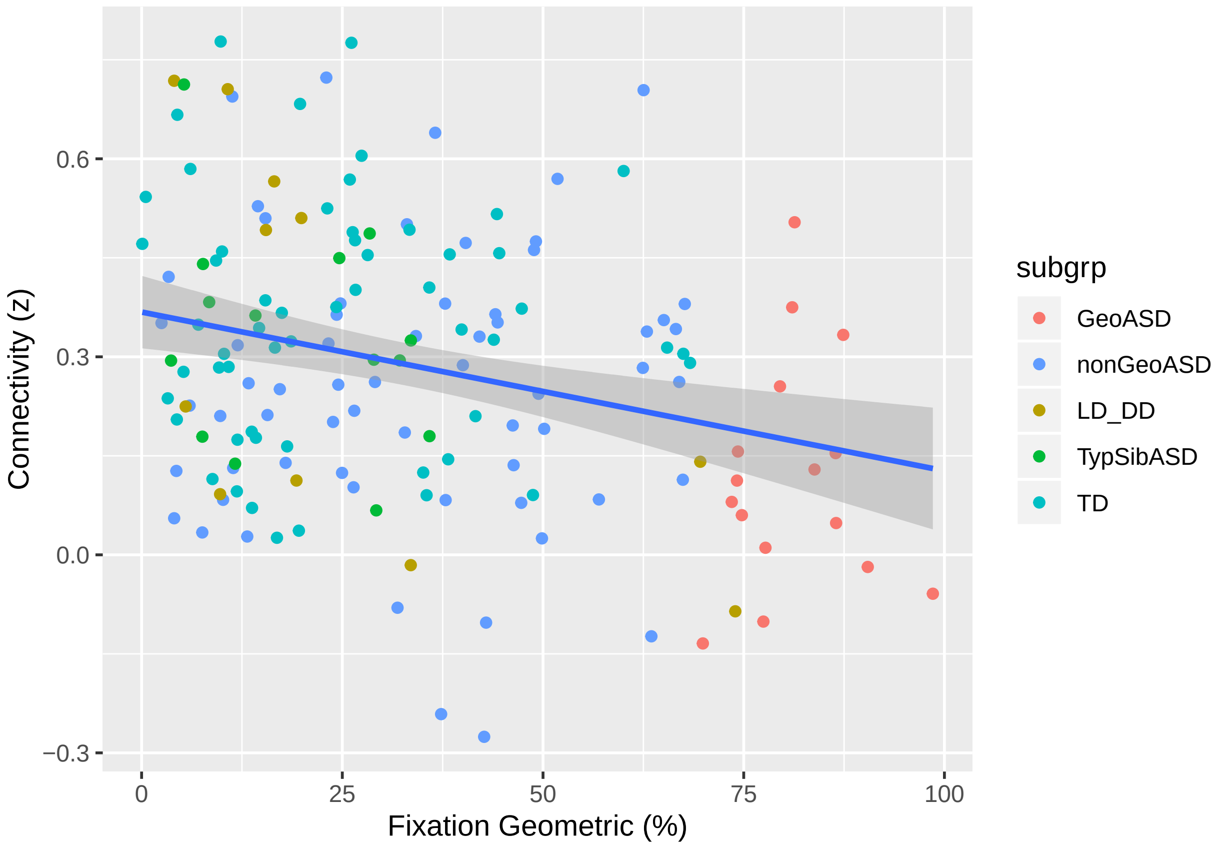
```r
        setStdMargins = FALSE, cex.text = 1.5, zlim = c(0,1),
        main = paste("Effect Size (Cohen's d)"))
    # dev.off()

    # make scatter-boxplots
    D$subgrp4plot = D$subgrp
    D$subgrp4plot = factor(D$subgrp4plot,levels(D$subgrp4plot)[c(1,2,6,3:5)])
  colours2use = get_ggColorHue(6)
    colours2use = c(colours2use[1],colours2use[5],colours2use[6],colours2use[2:4])

    xLabel = "Group"
    yLabel = "Connectivity (z)"
    p = ggplot(data = D, aes_string(x = "subgrp4plot", y = y_var, colour = "subgrp4plot"))
    p = p + geom_jitter(size = dotSize) +
        geom_boxplot(fill = NA, colour = "#000000", outlier.shape = NA) +
        guides(colour = FALSE)
    p = p + scale_y_continuous(limits = yLimits[i]$ylim,
                               breaks = round(seq(from = yLimits[i]$ylim[1],
                                                  to = yLimits[i]$ylim[2],
                                                  by = 0.1),digits=2)) +
      geom_hline(yintercept = 0, linetype = 2) +
      scale_colour_manual(values = colours2use) +
        xlab(xLabel) + ylab(yLabel) +
        theme(axis.text.x = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"),
            axis.text.y = element_text(size=fontSize+5,hjust=1,vjust=0,face="plain"),
            axis.title.x = element_text(size=fontSize+5,hjust=0.5,vjust=0,face="plain"),
            axis.title.y = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            strip.text.x = element_text(size = fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            plot.title = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"))
    ggsave(filename = file.path(plotdir,sprintf("FC_%s_subgrp_plot.pdf",y_var)))
    print(p)

  # Make RDOC plot
    colours2use = get_ggColorHue(6)
  colours2use = c(colours2use[1],colours2use[5],colours2use[2:4])
    p = ggplot(data = Dexp2, aes_string(x = "FixGeo", y = y_var)) +
      geom_point(data = Dexp2,aes(colour = subgrp)) + geom_smooth(method = lm)
    p = p + scale_colour_manual(values = colours2use) +
      xlab("Fixation Geometric (%)") + ylab("Connectivity (z)")
    ggsave(filename = file.path(plotdir,sprintf("FC_%s_RDOCgeoFix_plot.pdf",y_var)))
    print(p)
}# for (i in 1:length(sig_res))
```

```
## [1] "IC02_IC10: ANOVA on stratified model"
## Analysis of Variance Table
##
## Response: IC02_IC10
##            Df Sum Sq  Mean Sq F value    Pr(>F)
## subgrp      4 0.8513 0.212831  5.2464 0.0005418 ***
## sex         1 0.0277 0.027699  0.6828 0.4098801
## scan_age    1 0.0060 0.006044  0.1490 0.7000326
## Residuals 157 6.3690 0.040567
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##           Partial eta^2
## subgrp     0.1189322119
## sex        0.0040032808
## scan_age   0.0009480308
## Residuals            NA
## [1] "IC02_IC10: Statistics for each pairwise comparison"
##                  compName      tstat       pvalue        fdr_q     cohensd
## 1     GeoASD vs nonGeoASD -2.6543742 1.309552e-02 0.0261910354  0.72470597
## 2         GeoASD vs LD_DD -2.7516186 1.091693e-02 0.0261910354 -0.99527153
## 3     GeoASD vs TypSibASD -3.6308198 1.041578e-03 0.0052078906 -1.29532169
## 4            GeoASD vs TD -4.6631144 8.564567e-05 0.0008564567 -1.28028563
## 5      nonGeoASD vs LD_DD -1.1328690 2.713728e-01 0.3876753747 -0.37291200
## 6  nonGeoASD vs TypSibASD -1.7213301 9.640952e-02 0.1606825407 -0.47147648
## 7         nonGeoASD vs TD -2.7045624 7.878847e-03 0.0261910354 -0.51864838
## 8      LD_DD vs TypSibASD -0.1025426 9.191551e-01 0.9191550624 -0.06855427
## 9             LD_DD vs TD -0.2717866 7.888330e-01 0.9191550624  0.09283228
## 10        TypSibASD vs TD -0.2182013 8.289979e-01 0.9191550624  0.04117492
##    np.W    np.pvalue     np.fdr_q
## 1   291 1.138846e-02 0.0304180677
## 2    59 1.520903e-02 0.0304180677
## 3    44 1.059778e-03 0.0052988919
## 4   156 9.558253e-05 0.0009558253
## 5   379 2.714698e-01 0.3878139525
## 6   377 1.425512e-01 0.2375852823
## 7  1253 1.367676e-02 0.0304180677
## 8   122 9.534141e-01 0.9534140883
## 9   399 8.523894e-01 0.9470993045
## 10  413 7.153397e-01 0.8941745667
## [1] "IC02_IC10: Model comparison for subtype vs case-control models"
##                                             AIC
## IC02_IC10 ~ subgrp + sex + scan_age       -51.32965
## IC02_IC10 ~ CaseControl2 + sex + scan_age -47.60162
## [1] "IC02_IC10: Model comparison for RDOC vs subtype models"
##                                       AIC
## IC02_IC10 ~ subgrp + sex + scan_age -51.32965
## IC02_IC10 ~ FixGeo + sex + scan_age -46.13133
```

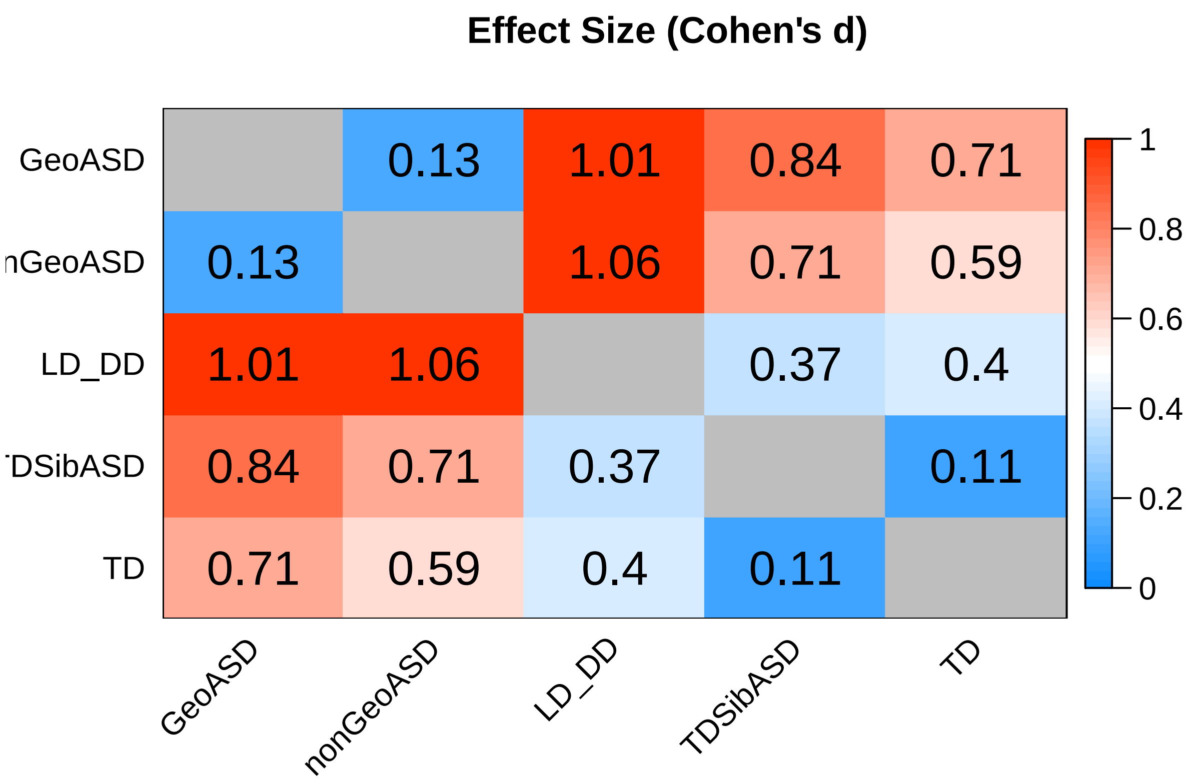

Effect Size (Cohen's d)
GeoASD
nGeoASD
LD_DD
TDSibASD
TD
0.72
1
1.3
1.28
0.37
0.47
0.52
0.07
0.09
0.04
GeoASD
nonGeoASD
LD_DD
TDSibASD
TD
1
0.8
0.6
0.4
0.2
0

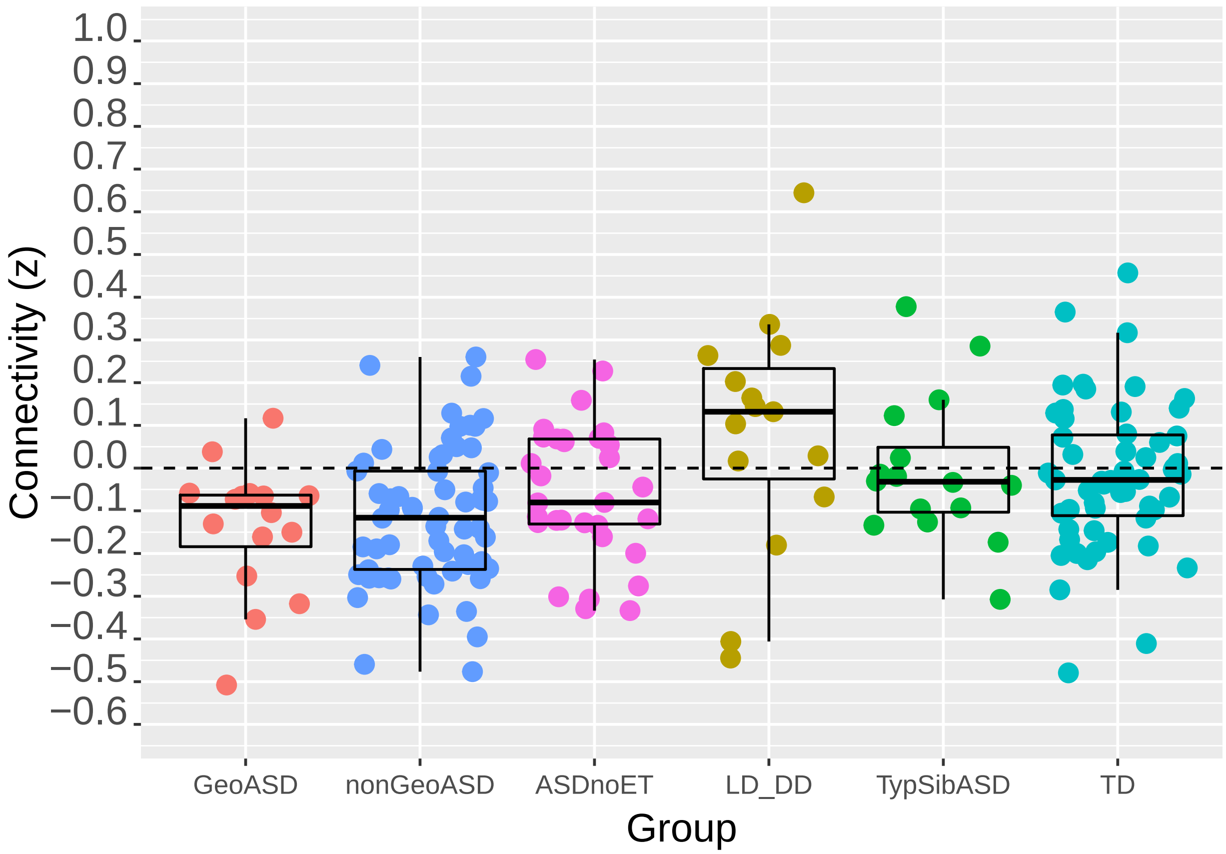

Connectivity (z)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
GeoASD
nonGeoASD
ASDnoET
LD_DD
TypSibASD
TD
Group

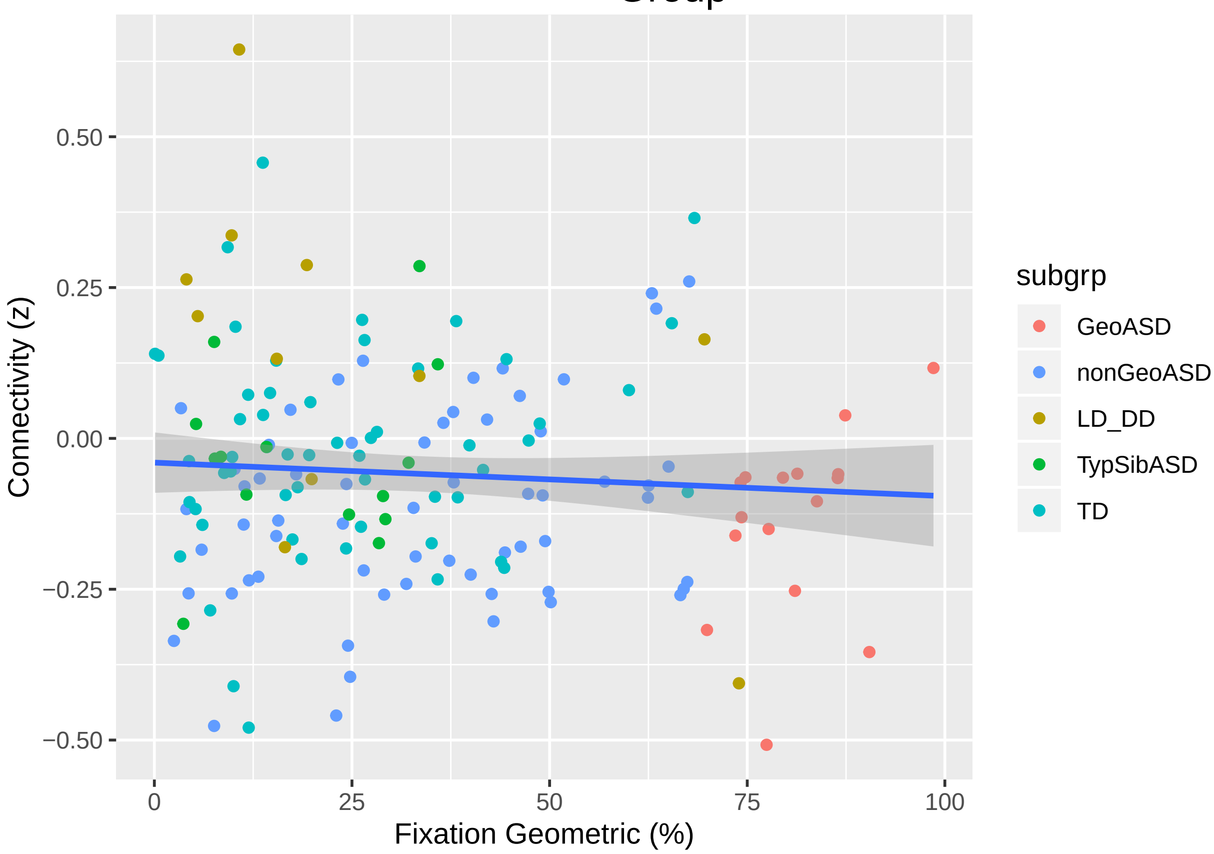



```
## [1] "IC05_IC10: ANOVA on stratified model"
## Analysis of Variance Table
##
## Response: IC05_IC10
##            Df Sum Sq  Mean Sq F value    Pr(>F)
## subgrp      4 0.7188 0.179700  5.5305 0.0003429 ***
## sex         1 0.0258 0.025780  0.7934 0.3744368
## scan_age    1 0.0024 0.002413  0.0743 0.7855749
## Residuals 157 5.1014 0.032493
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##            Partial eta^2
## subgrp       0.126778502
## sex          0.005234935
## scan_age     0.000472840
## Residuals             NA
## [1] "IC05_IC10: Statistics for each pairwise comparison"
##                  compName      tstat      pvalue      fdr_q    cohensd np.W
## 1     GeoASD vs nonGeoASD -0.5469707 0.58940781 0.65489757  0.1334773  490
## 2         GeoASD vs LD_DD -2.6798823 0.01384141 0.04613803 -1.0068856   54
## 3     GeoASD vs TypSibASD -2.2801524 0.02996097 0.05607706 -0.8434620   72
## 4            GeoASD vs TD -2.6438644 0.01348821 0.04613803 -0.7123278  262
## 5      nonGeoASD vs LD_DD -2.6046314 0.01893174 0.04732936 -1.0613451  224
## 6  nonGeoASD vs TypSibASD -2.2636323 0.03364624 0.05607706 -0.7090093  326
## 7         nonGeoASD vs TD -3.0678711 0.00270912 0.02709120 -0.5891286 1142
## 8      LD_DD vs TypSibASD  1.0424023 0.30805908 0.38507385  0.3688638  156
## 9             LD_DD vs TD  1.3187830 0.20469367 0.29241952 -0.4003244  550
```

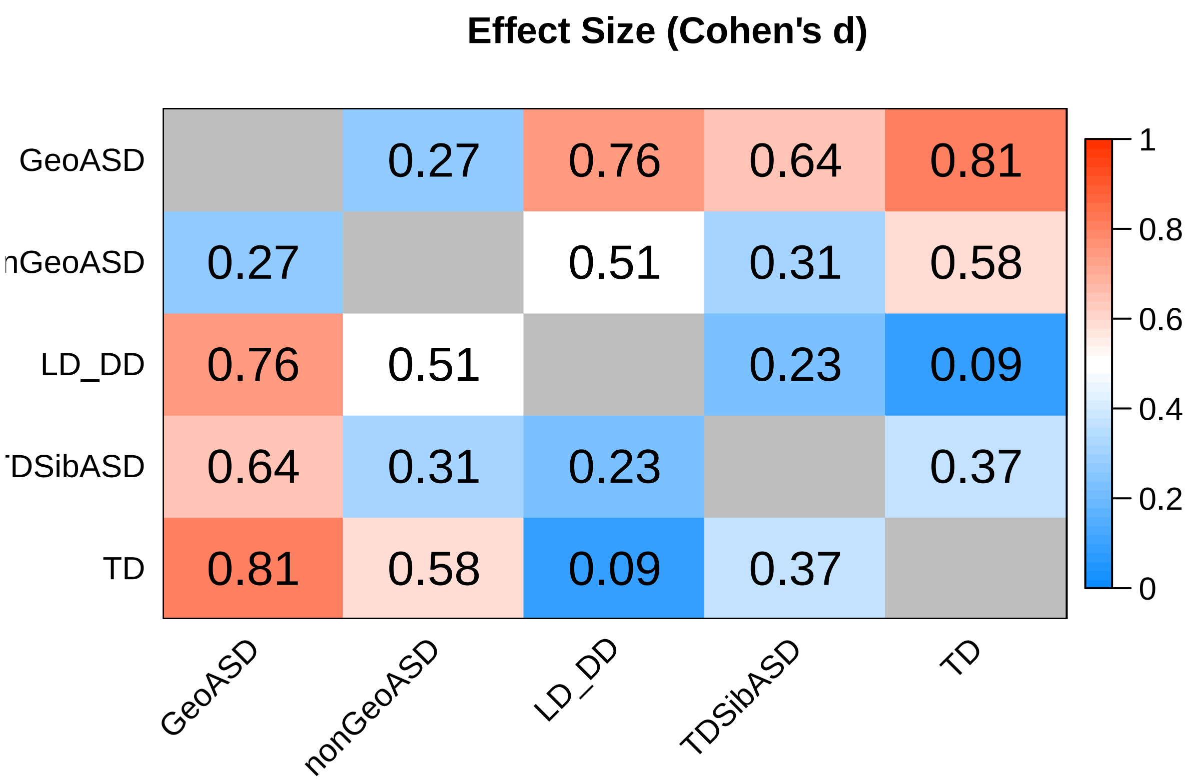
```
## 10        TypSibASD vs TD  0.2547326 0.80105022 0.80105022 -0.1092939  438
##      np.pvalue   np.fdr_q
## 1  0.945738872 0.98353038
## 2  0.008212685 0.02737562
## 3  0.035150037 0.05992341
## 4  0.014575291 0.03643823
## 5  0.001979682 0.01063810
## 6  0.035954046 0.05992341
## 7  0.002127620 0.01063810
## 8  0.162740408 0.20342551
## 9  0.049890903 0.07127272
## 10 0.983530382 0.98353038
## [1] "IC05_IC10: Model comparison for subtype vs case-control models"
##                                                AIC
## IC05_IC10 ~ subgrp + sex + scan_age       -87.72723
## IC05_IC10 ~ CaseControl2 + sex + scan_age -89.54315
## [1] "IC05_IC10: Model comparison for RDOC vs subtype models"
##                                          AIC
## IC05_IC10 ~ subgrp + sex + scan_age -87.72723
## IC05_IC10 ~ FixGeo + sex + scan_age -74.63179
```

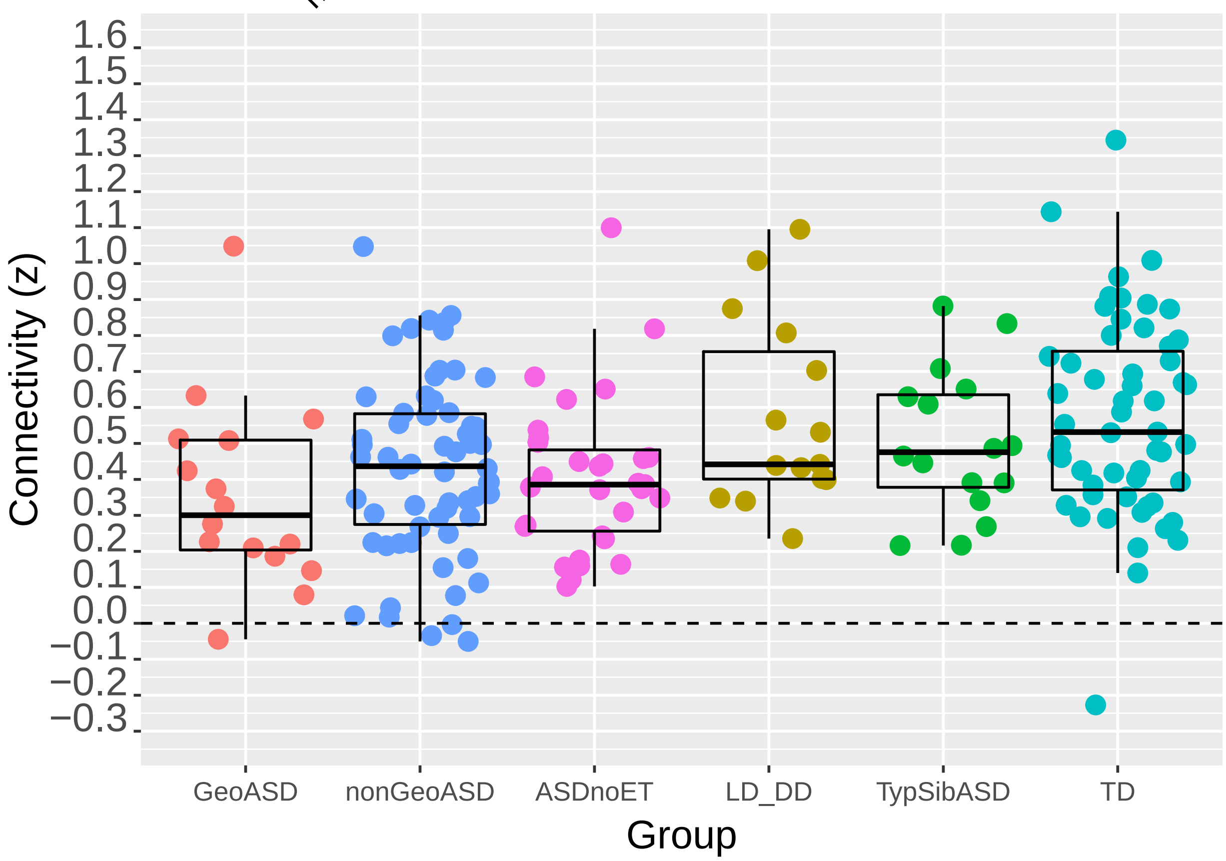


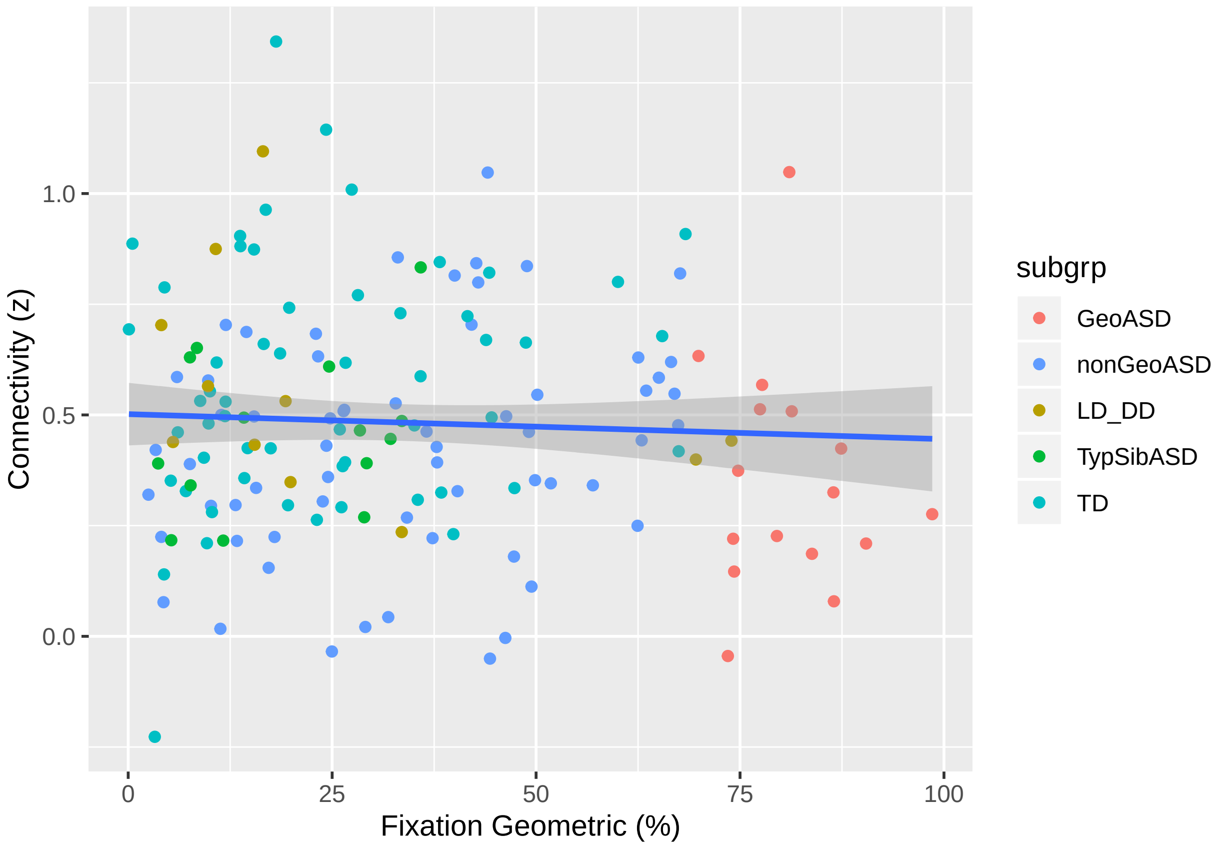

1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.6
Connectivity (z)
GeoASD
nonGeoASD
ASDnoET
LD_DD
TypSibASD
TD
Group


0.50
0.25
0.00
−0.25
−0.50
Connectivity (z)
0
25
50
75
100
Fixation Geometric (%)
subgrp
GeoASD
nonGeoASD
LD_DD
TypSibASD
TD

```
## [1] "IC09_IC10: ANOVA on stratified model"
## Analysis of Variance Table
##
## Response: IC09_IC10
##            Df Sum Sq Mean Sq F value  Pr(>F)
## subgrp      4 0.9746 0.24366  3.8809 0.00492 **
## sex         1 0.1426 0.14260  2.2713 0.13380
## scan_age    1 0.4806 0.48064  7.6554 0.00634 **
## Residuals 157 9.8571 0.06278
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##           Partial eta^2
## subgrp       0.08866242
## sex          0.01050586
## scan_age     0.04649366
## Residuals            NA
## [1] "IC09_IC10: Statistics for each pairwise comparison"
##                  compName       tstat      pvalue      fdr_q     cohensd
## 1     GeoASD vs nonGeoASD -1.03272101 0.312762087 0.39095261  0.27318454
## 2         GeoASD vs LD_DD -2.32828059 0.027109839 0.09036613 -0.75648006
## 3     GeoASD vs TypSibASD -1.76278020 0.088748239 0.17749648 -0.64038849
## 4            GeoASD vs TD -2.85165854 0.008475410 0.04237705 -0.81099881
## 5      nonGeoASD vs LD_DD -1.93400349 0.067070406 0.16767601 -0.51051908
## 6  nonGeoASD vs TypSibASD -1.19094727 0.243811749 0.39095261 -0.31174329
## 7         nonGeoASD vs TD -2.86843486 0.004954703 0.04237705 -0.58167336
## 8      LD_DD vs TypSibASD  0.86341158 0.395683043 0.43964783  0.23353328
## 9             LD_DD vs TD  0.05346514 0.957816772 0.95781677  0.08578674
## 10        TypSibASD vs TD -1.09186292 0.282807850 0.39095261  0.36644480
##    np.W   np.pvalue   np.fdr_q
## 1   390 0.191725895 0.31954316
## 2    60 0.017094319 0.05698106
## 3    78 0.061458212 0.15364553
## 4   227 0.003450821 0.03450821
## 5   341 0.112188540 0.22437708
## 6   418 0.337553921 0.42194240
## 7  1227 0.009117170 0.04558585
## 8   133 0.625980888 0.69553432
## 9   410 0.977162676 0.97716268
## 10  368 0.325122217 0.42194240
## [1] "IC09_IC10: Model comparison for subtype vs case-control models"
##                                                AIC
## IC09_IC10 ~ subgrp + sex + scan_age       20.29783
## IC09_IC10 ~ CaseControl2 + sex + scan_age 19.20720
## [1] "IC09_IC10: Model comparison for RDOC vs subtype models"
##                                          AIC
## IC09_IC10 ~ subgrp + sex + scan_age 20.29783
## IC09_IC10 ~ FixGeo + sex + scan_age 25.42071
```


Effect Size (Cohen's d)
GeoASD
nGeoASD
LD_DD
TDSibASD
TD
0.27
0.76
0.64
0.81
0.51
0.31
0.58
0.23
0.09
0.37
1
0.8
0.6
0.4
0.2
0
GeoASD
nonGeoASD
LD_DD
TDSibASD
TD


Connectivity (z)
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
GeoASD
nonGeoASD
ASDnoET
LD_DD
TypSibASD
TD
Group



Cross-validation of models to find best model with lowest mean squared prediction error and mean absolute percentage error

```
sig_res = vars2use[aovres$fdr_all<=fdr_thresh]

Dexp2 = subset(D, is.element(D$subgrp,c("GeoASD","nonGeoASD","LD_DD","TypSibASD","TD")),
               select = 1:ncol(D))
Dexp2$subgrp = factor(Dexp2$subgrp)
Dexp2$Case_vs_nonASD = NA
Dexp2$Case_vs_nonASD[is.element(Dexp2$subgrp,c("LD_DD","TypSibASD"))] = "nonASD"
Dexp2$Case_vs_nonASD[is.element(Dexp2$subgrp,c("GeoASD","nonGeoASD"))] = "ASD"
Dexp2$Case_vs_nonASD = factor(Dexp2$Case_vs_nonASD)

kfolds = 5

cols2use = c("avg_mspe_subtype","avg_mspe_casectrl")
res = data.frame(matrix(nrow = length(sig_res), ncol = length(cols2use)))
colnames(res) = cols2use
rownames(res) = sig_res

subtype_mspe_res = data.frame(matrix(nrow = kfolds, ncol=length(sig_res)))
cc_mspe_res = data.frame(matrix(nrow = kfolds, ncol=length(sig_res)))
colnames(subtype_mspe_res) = sig_res
colnames(cc_mspe_res) = sig_res

subtype_mape_res = data.frame(matrix(nrow = kfolds, ncol=length(sig_res)))
cc_mape_res = data.frame(matrix(nrow = kfolds, ncol=length(sig_res)))
```

```r
colnames(subtype_mape_res) = sig_res
colnames(cc_mape_res) = sig_res

# loop over number of significant connections
for (i in 1:length(sig_res)) {
  y_var = sig_res[i]

  # make cross-validation indices
  set.seed(999)
  cvind = vfold_cv(data = Dexp2, v = 5, strata = "subgrp")

  for (k in 1:kfolds){
    training_mask = vector(length = dim(Dexp2)[1])
    training_mask[cvind$splits[[k]]$in_id] = TRUE
    test_mask = !training_mask

    training_data = Dexp2[training_mask,]
    test_data = Dexp2[test_mask,]

    # model using only subgrp
    lm_formula = as.formula(sprintf("%s ~ %s + %s + %s",y_var,"subgrp","sex","scan_age"))
    subtype_model = eval(substitute(lm(formula = lm_formula, data = training_data, na.action = na.omit)
    predres = predict.lm(subtype_model, newdata = test_data)

    # MSPE
    residual_data = test_data[,y_var] - predres
    sq_error = residual_data^2
    subtype_mspe_res[k,sig_res[i]] = mean(sq_error, na.rm = TRUE)

    # MAPE
    subtype_mape_res[k,sig_res[i]] = mean(abs((test_data[,y_var] - predres)/test_data[,y_var])*100)

    # model using case-control
    lm_formula = as.formula(sprintf("%s ~ %s + %s + %s",y_var,"CaseControl2","sex","scan_age"))
    cc_model = eval(substitute(lm(formula = lm_formula, data = training_data, na.action = na.omit)))
    predres = predict.lm(cc_model, newdata = test_data)

    # MSPE
    residual_data = test_data[,y_var] - predres
    sq_error = residual_data^2
    cc_mspe_res[k,sig_res[i]] = mean(sq_error, na.rm = TRUE)

    # MAPE
    cc_mape_res[k,sig_res[i]] = mean(abs((test_data[,y_var] - predres)/test_data[,y_var])*100)

  }#for (k in 1:kfolds){

}#for (i in 1:length(sig_res))

final_res = data.frame(rbind(colMeans(subtype_mspe_res),colMeans(cc_mspe_res)))
rownames(final_res) = c("subtype_model","casectrl_model")
write.csv(final_res,file = file.path(resultdir,"cv_mse_model_comparison.csv"))
final_res
```

```
##                 IC02_IC10  IC05_IC10 IC09_IC10
## subtype_model  0.04194458 0.03285194 0.0632078
## casectrl_model 0.04301352 0.03264858 0.0634412
```

```
final_res = data.frame(rbind(colMeans(subtype_mape_res),colMeans(cc_mape_res)))
rownames(final_res) = c("subtype_model","casectrl_model")
write.csv(final_res,file = file.path(resultdir,"cv_mape_model_comparison.csv"))
final_res
```

```
##                IC02_IC10 IC05_IC10 IC09_IC10
## subtype_model   125.5452  152.8099  156.0619
## casectrl_model  135.2241  152.1738  152.8081
```

Permutation tests on specific significant connections and looking for subtype differences

```
nperm = 10000
set.seed(1)

sig_res = vars2use[aovres$fdr_all<=fdr_thresh]
sig_res
```

```
## [1] "IC02_IC10" "IC05_IC10" "IC09_IC10"
```

```
Dexp2 = subset(D, is.element(D$subgrp,c("GeoASD","nonGeoASD","LD_DD","TypSibASD","TD")),
               select = 1:ncol(D))
Dexp2$subgrp = factor(Dexp2$subgrp)
Dexp2$Case_vs_nonASD = NA
Dexp2$Case_vs_nonASD[is.element(Dexp2$subgrp,c("LD_DD","TypSibASD"))] = "nonASD"
Dexp2$Case_vs_nonASD[is.element(Dexp2$subgrp,c("GeoASD","nonGeoASD"))] = "ASD"
Dexp2$Case_vs_nonASD = factor(Dexp2$Case_vs_nonASD)

# set up comparisons to run
comp1 = c("GeoASD","nonGeoASD","LD_DD","TypSibASD")
comp2 = list(comp = c("nonGeoASD","LD_DD","TypSibASD","TD"),
             comp = c("LD_DD","TypSibASD","TD"),
             comp = c("TypSibASD","TD"),
             comp = c("TD"))

# set up data frame for storing multiple comparison results
colnames2use = c("compName","tstat")
mcompres_perm = list()

# loop over number of significant connections
for (i in 1:length(sig_res)) {
  mcompres_perm[[i]] = data.frame(matrix(nrow = 10,ncol = length(colnames2use)))
  colnames(mcompres_perm[[i]]) = colnames2use
  mcompres_perm[[i]]$tstat = 1

  for (iperm in 1:nperm){

    # permute group label
    Dexp2$subgrp_perm = sample(Dexp2$subgrp)
```

```r
    y_var = sig_res[i]

    # multiple comparisons on subgroup
    compCount = 0
    for (ic1 in 1:length(comp1)) {
        for (ic2 in 1:length(comp2[ic1]$comp)) {
            compCount = compCount + 1
            compname1 = comp1[ic1]
            compname2 = comp2[ic1]$comp[ic2]

            # statistic on real data
            Dcomp_real = subset(Dexp2, Dexp2$subgrp==compname1 | Dexp2$subgrp==compname2,
                           select = c("subgrp",y_var))
            Dcomp_real$subgrp = factor(Dcomp_real$subgrp)

            tres_real = t.test(Dcomp_real[Dcomp_real$subgrp==compname1,y_var],
                               Dcomp_real[Dcomp_real$subgrp==compname2,y_var])

        # statistic on permuted data
            Dcomp_perm = subset(Dexp2, Dexp2$subgrp_perm==compname1 | Dexp2$subgrp_perm==compname2,
                           select = c("subgrp_perm",y_var))
            Dcomp_perm$subgrp = factor(Dcomp_perm$subgrp)

            tres_perm = t.test(Dcomp_perm[Dcomp_perm$subgrp_perm==compname1,y_var],
                               Dcomp_perm[Dcomp_perm$subgrp_perm==compname2,y_var])

            # fill in mcompres_perm
            mcompres_perm[[i]]$compName[compCount] = sprintf("%s vs %s",compname1,compname2)
            if (abs(tres_perm$statistic) >= abs(tres_real$statistic)){
              mcompres_perm[[i]]$tstat[compCount] = mcompres_perm[[i]]$tstat[compCount]+1
            } # if

        }#for (ic2 in 1:length(comp2[ic1]$comp))
    }#for (ic1 in 1:length(comp1))
  } # for (iperm in 1:nperm)

  mcompres_perm[[i]]$pval = mcompres_perm[[i]]$tstat/(nperm+1)
  mcompres_perm[[i]]$fdr = p.adjust(mcompres_perm[[i]]$pval, method = "fdr")
}# for (i in 1:length(sig_res))
sig_res[1]
```

```
## [1] "IC02_IC10"
```

```r
mcompres_perm[[1]]
```

```
##                  compName tstat       pval        fdr
## 1      GeoASD vs nonGeoASD   156 0.01559844 0.03119688
## 2          GeoASD vs LD_DD   106 0.01059894 0.02649735
## 3      GeoASD vs TypSibASD    17 0.00169983 0.00849915
## 4             GeoASD vs TD     4 0.00039996 0.00399960
## 5       nonGeoASD vs LD_DD  2665 0.26647335 0.38067622
## 6   nonGeoASD vs TypSibASD   939 0.09389061 0.15648435
## 7          nonGeoASD vs TD    82 0.00819918 0.02649735
## 8       LD_DD vs TypSibASD  9122 0.91210879 0.91210879
```

```
## 9             LD_DD vs TD  7878 0.78772123 0.91210879
## 10        TypSibASD vs TD  8239 0.82381762 0.91210879
```

```
sig_res[2]
```

```
## [1] "IC05_IC10"
```

```
mcompres_perm[[2]]
```

```
##                  compName tstat       pval        fdr
## 1     GeoASD vs nonGeoASD  5961 0.59604040 0.66226711
## 2         GeoASD vs LD_DD   118 0.01179882 0.03449655
## 3     GeoASD vs TypSibASD   284 0.02839716 0.04866180
## 4           GeoASD vs TD   121 0.01209879 0.03449655
## 5      nonGeoASD vs LD_DD   138 0.01379862 0.03449655
## 6  nonGeoASD vs TypSibASD   292 0.02919708 0.04866180
## 7         nonGeoASD vs TD    25 0.00249975 0.02499750
## 8      LD_DD vs TypSibASD  3102 0.31016898 0.38771123
## 9             LD_DD vs TD  2002 0.20017998 0.28597140
## 10        TypSibASD vs TD  8060 0.80591941 0.80591941
```

```
sig_res[3]
```

```
## [1] "IC09_IC10"
```

```
mcompres_perm[[3]]
```

```
##                  compName tstat       pval       fdr
## 1     GeoASD vs nonGeoASD  3148 0.31476852 0.3934607
## 2         GeoASD vs LD_DD   282 0.02819718 0.0939906
## 3     GeoASD vs TypSibASD   868 0.08679132 0.1735826
## 4           GeoASD vs TD    90 0.00899910 0.0449955
## 5      nonGeoASD vs LD_DD   631 0.06309369 0.1577342
## 6  nonGeoASD vs TypSibASD  2404 0.24037596 0.3934607
## 7         nonGeoASD vs TD    55 0.00549945 0.0449955
## 8      LD_DD vs TypSibASD  4016 0.40155984 0.4461776
## 9             LD_DD vs TD  9569 0.95680432 0.9568043
## 10        TypSibASD vs TD  2828 0.28277172 0.3934607
```

Connectivity-social affect correlation analyses

```
behav_data = subset(Dfmri, is.element(Dfmri$subgrp2,c("GeoASD","nonGeoASD")),
    select = c("subjectId","scan_age","sex","subgrp2","meanFD",
               "meanDVARSwavelet", "ADOSatscan_age", "ADOS_CoSoTot"))

conn_data = subset(D, is.element(D$subgrp,c("GeoASD","nonGeoASD")),
                   select = c("subjectId",sig_res))

data4analysis = merge(behav_data,conn_data)
write.table(data4analysis,
            file = file.path(tidydatadir,"data4connSocOrientcorr.txt"),
            sep = "\t", quote = FALSE, row.names = FALSE)

# Run correlation analyses and bootstrapping in MATLAB
if (RUNMATLAB){
```

```
  code2run = sprintf("cd %s; batch_connSocAffectcorr;",codedir)
  res = run_matlab_code(code2run)
}
```

Plot connectivity ADOS social affect relationships

IC02-IC10 ADOS Social Affect

```
yLimits = c(0,25)
comp2plot = "IC02_IC10"
geo_res = read.delim(file.path(tidydatadir,
                               sprintf("GeoASD_%s_ADOS_CoSoTot_corrRes.txt",comp2plot)))
nongeo_res = read.delim(file.path(tidydatadir,
                                  sprintf("nonGeoASD_%s_ADOS_CoSoTot_corrRes.txt",comp2plot)))
corr_res = rbind(geo_res,nongeo_res)
rownames(corr_res) = c("GeoASD","nonGeoASD")
corr_res
```

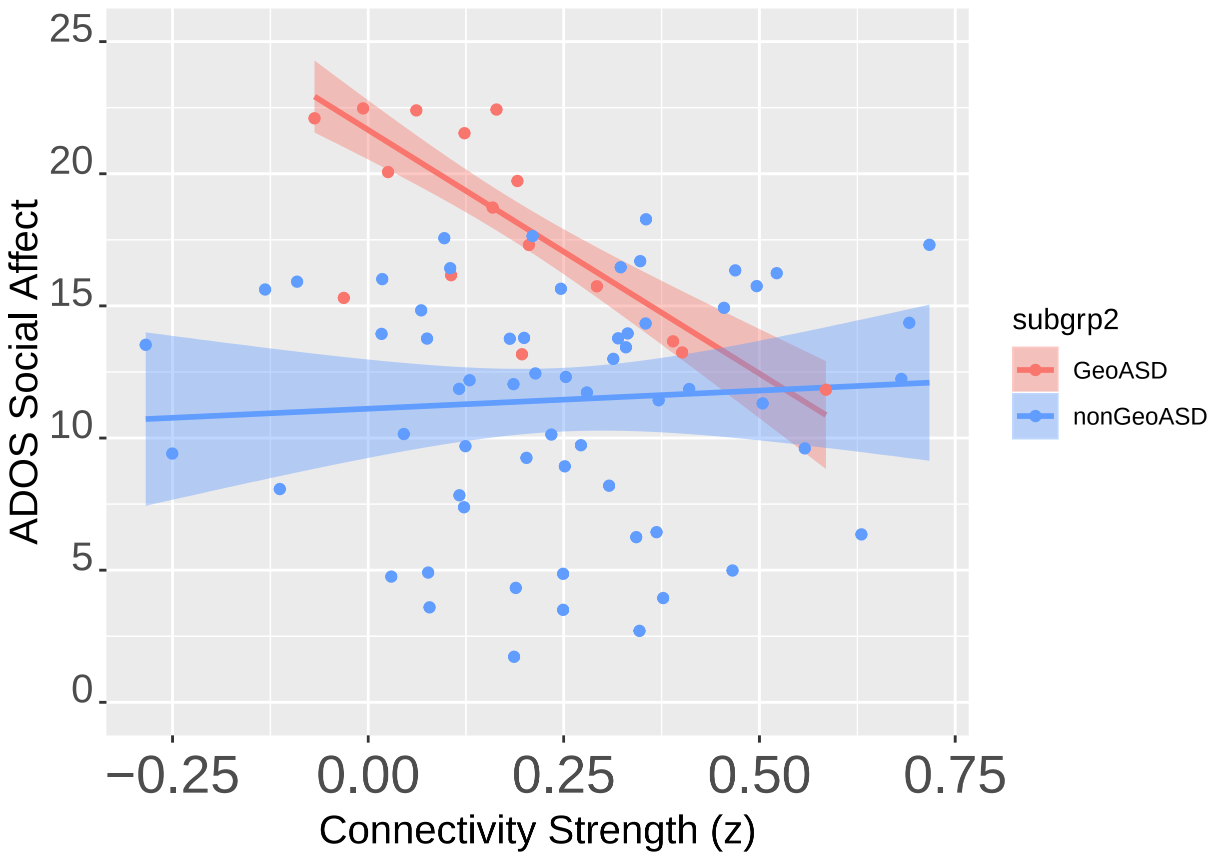
```
##                     r          p     ci95lo     ci95hi
## GeoASD    -0.78979061 0.00196584 -0.9958642 -0.3516584
## nonGeoASD  0.06866816 0.64455412 -0.2024230  0.3311468
```

```
n = sum(Dexp2$subgrp=="GeoASD")
n2 = sum(Dexp2$subgrp=="nonGeoASD")
r_comp_res = paired.r(geo_res$r,nongeo_res$r,n=n,n2=n2)
comp_res = data.frame(matrix(nrow = 1, ncol = 2))
colnames(comp_res) = c("z","p")
rownames(comp_res) = sprintf("GeoASD vs nonGeoASD: %s, ADOS_CoSoTot",comp2plot)
comp_res$z = r_comp_res$z
comp_res$p = r_comp_res$p
comp_res
```

```
##                                                      z            p
## GeoASD vs nonGeoASD: IC02_IC10, ADOS_CoSoTot 3.719662 0.0001994899
```

```
geo_adj = read.delim(file.path(tidydatadir,
                               sprintf("GeoASD_%s_ADOS_CoSoTot_xy_adjdata.txt",comp2plot)))
nongeo_adj = read.delim(file.path(tidydatadir,
                                  sprintf("nonGeoASD_%s_ADOS_CoSoTot_xy_adjdata.txt",comp2plot)))
Dtmp = rbind(geo_adj,nongeo_adj)

Dconn = merge(Dfmri, Dtmp, by = "subjectId")

cols2use = get_ggColorHue(6)
cols2use = c(cols2use[1],cols2use[5])
g = ggplot(data = Dconn, aes(y = y_adj, x = x_adj, colour=subgrp2, fill = subgrp2))
g = g + geom_smooth(method = rlm, method.args = list(psi = psi.bisquare)) + geom_point() +
    scale_colour_manual(values = cols2use) + scale_fill_manual(values = cols2use)
yLabel = "ADOS Social Affect"
xLabel = "Connectivity Strength (z)"
g = g + xlab(xLabel) + ylab(yLabel) + theme(axis.text.x = element_text(size=fontSize+10,hjust=0.5,vjust=
            axis.text.y = element_text(size=fontSize+5,hjust=1,vjust=0,face="plain"),
```

```
            axis.title.x = element_text(size=fontSize+5,hjust=0.5,vjust=0,face="plain"),
            axis.title.y = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            strip.text.x = element_text(size = fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            plot.title = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"))
g = g + scale_y_continuous(limits = yLimits)
ggsave(filename = file.path(plotdir,sprintf("FC_%s_ADOSsoceng_corr_plot.pdf",comp2plot)))
g
```



IC05-IC10 ADOS Social Affect

```
comp2plot = "IC05_IC10"
geo_res = read.delim(file.path(tidydatadir,
                              sprintf("GeoASD_%s_ADOS_CoSoTot_corrRes.txt",comp2plot)))
nongeo_res = read.delim(file.path(tidydatadir,
                                 sprintf("nonGeoASD_%s_ADOS_CoSoTot_corrRes.txt",comp2plot)))
corr_res = rbind(geo_res,nongeo_res)
rownames(corr_res) = c("GeoASD","nonGeoASD")
corr_res
```

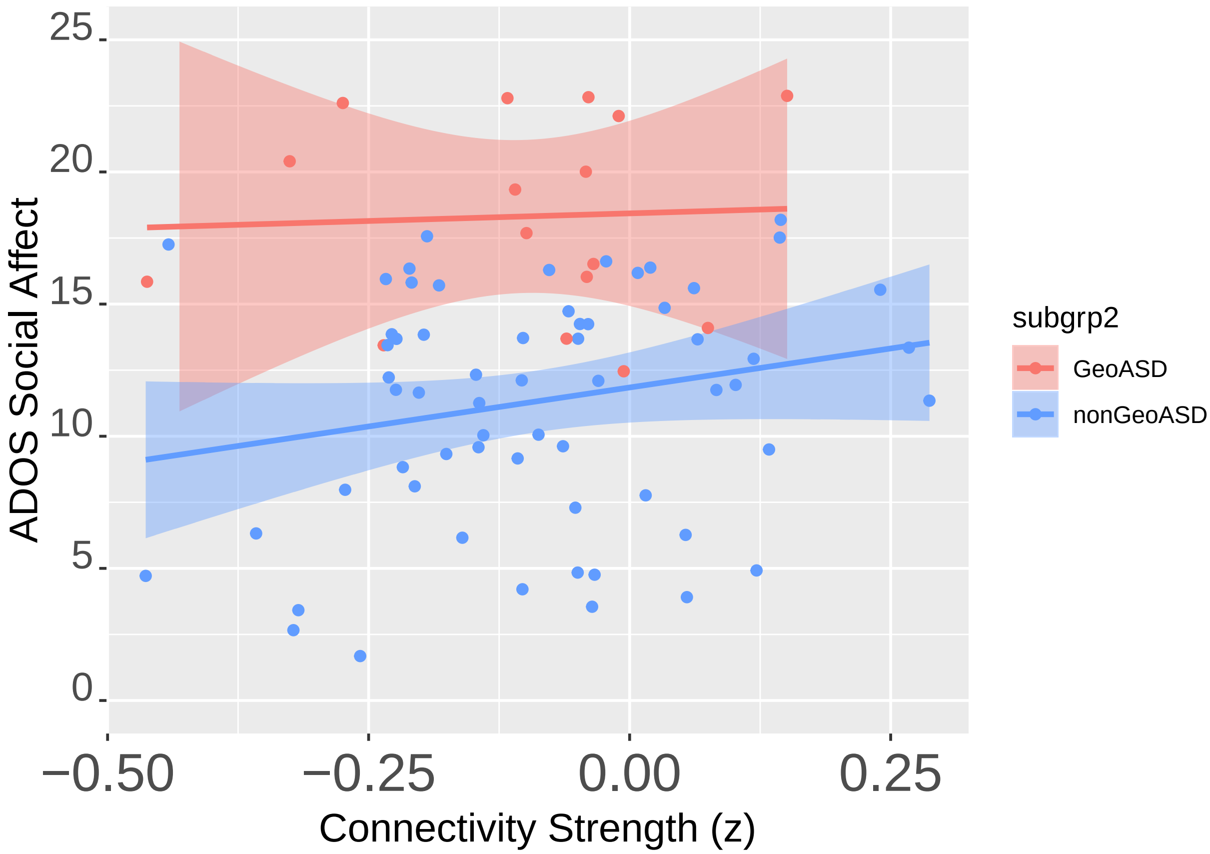
```
##                    r         p     ci95lo    ci95hi
## GeoASD    0.04567706 0.8810737 -0.8157176 0.7518075
## nonGeoASD 0.22399054 0.1260338 -0.1023718 0.4920886
```

```
n = sum(Dexp2$subgrp=="GeoASD")
n2 = sum(Dexp2$subgrp=="nonGeoASD")
```

```r
r_comp_res = paired.r(geo_res$r,nongeo_res$r,n=n,n2=n2)
comp_res = data.frame(matrix(nrow = 1, ncol = 2))
colnames(comp_res) = c("z","p")
rownames(comp_res) = sprintf("GeoASD vs nonGeoASD: %s, ADOS_CoSoTot",comp2plot)
comp_res$z = r_comp_res$z
comp_res$p = r_comp_res$p
comp_res
```

```
##                                                       z         p
## GeoASD vs nonGeoASD: IC05_IC10, ADOS_CoSoTot 0.5944945 0.5521814
```

```r
geo_adj = read.delim(file.path(tidydatadir,
                               sprintf("GeoASD_%s_ADOS_CoSoTot_xy_adjdata.txt",comp2plot)))
nongeo_adj = read.delim(file.path(tidydatadir,
                                  sprintf("nonGeoASD_%s_ADOS_CoSoTot_xy_adjdata.txt",comp2plot)))
Dtmp = rbind(geo_adj,nongeo_adj)

Dconn = merge(Dfmri, Dtmp, by = "subjectId")

cols2use = get_ggColorHue(6)
cols2use = c(cols2use[1],cols2use[5])
g = ggplot(data = Dconn, aes(y = y_adj, x = x_adj, colour=subgrp2, fill = subgrp2))
g = g + geom_smooth(method = rlm, method.args = list(psi = psi.bisquare)) + geom_point() +
    scale_colour_manual(values = cols2use) + scale_fill_manual(values = cols2use)
yLabel = "ADOS Social Affect"
xLabel = "Connectivity Strength (z)"
g = g + xlab(xLabel) + ylab(yLabel) + theme(axis.text.x = element_text(size=fontSize+10,hjust=0.5,vjust
            axis.text.y = element_text(size=fontSize+5,hjust=1,vjust=0,face="plain"),
            axis.title.x = element_text(size=fontSize+5,hjust=0.5,vjust=0,face="plain"),
            axis.title.y = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            strip.text.x = element_text(size = fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            plot.title = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"))
g = g + scale_y_continuous(limits = yLimits)
ggsave(filename = file.path(plotdir,sprintf("FC_%s_ADOSsoceng_corr_plot.pdf",comp2plot)))
g
```



IC09-IC10 ADOS Social Affect

```
comp2plot = "IC09_IC10"
geo_res = read.delim(file.path(tidydatadir,
                               sprintf("GeoASD_%s_ADOS_CoSoTot_corrRes.txt",comp2plot)))
nongeo_res = read.delim(file.path(tidydatadir,
                                  sprintf("nonGeoASD_%s_ADOS_CoSoTot_corrRes.txt",comp2plot)))
corr_res = rbind(geo_res,nongeo_res)
rownames(corr_res) = c("GeoASD","nonGeoASD")
corr_res
```

```
##                    r         p      ci95lo    ci95hi
## GeoASD    -0.3389316 0.2563662 -0.83704174 0.5817597
## nonGeoASD  0.2950895 0.0452685  0.05118179 0.4937473
```

```
n = sum(Dexp2$subgrp=="GeoASD")
n2 = sum(Dexp2$subgrp=="nonGeoASD")
r_comp_res = paired.r(geo_res$r,nongeo_res$r,n=n,n2=n2)
comp_res = data.frame(matrix(nrow = 1, ncol = 2))
colnames(comp_res) = c("z","p")
rownames(comp_res) = sprintf("GeoASD vs nonGeoASD: %s, ADOS_CoSoTot",comp2plot)
comp_res$z = r_comp_res$z
comp_res$p = r_comp_res$p
comp_res
```

```
##                                                     z          p
```

```
## GeoASD vs nonGeoASD: IC09_IC10, ADOS_CoSoTot 2.144411 0.03199995
```

```
geo_adj = read.delim(file.path(tidydatadir,
                               sprintf("GeoASD_%s_ADOS_CoSoTot_xy_adjdata.txt",comp2plot)))
nongeo_adj = read.delim(file.path(tidydatadir,
                                  sprintf("nonGeoASD_%s_ADOS_CoSoTot_xy_adjdata.txt",comp2plot)))
Dtmp = rbind(geo_adj,nongeo_adj)

Dconn = merge(Dfmri, Dtmp, by = "subjectId")

cols2use = get_ggColorHue(6)
cols2use = c(cols2use[1],cols2use[5])
g = ggplot(data = Dconn, aes(y = y_adj, x = x_adj, colour=subgrp2, fill = subgrp2))
g = g + geom_smooth(method = rlm, method.args = list(psi = psi.bisquare)) + geom_point() +
    scale_colour_manual(values = cols2use) + scale_fill_manual(values = cols2use)
yLabel = "ADOS Social Affect"
xLabel = "Connectivity Strength (z)"
g = g + xlab(xLabel) + ylab(yLabel) + theme(axis.text.x = element_text(size=fontSize+10,hjust=0.5,vjust=
            axis.text.y = element_text(size=fontSize+5,hjust=1,vjust=0,face="plain"),
            axis.title.x = element_text(size=fontSize+5,hjust=0.5,vjust=0,face="plain"),
            axis.title.y = element_text(size=fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            strip.text.x = element_text(size = fontSize+5,hjust=0.5,vjust=0.5,face="plain"),
            plot.title = element_text(size=fontSize,hjust=0.5,vjust=0.5,face="plain"))
g = g + scale_y_continuous(limits = yLimits)
ggsave(filename = file.path(plotdir,sprintf("FC_%s_ADOSsoceng_corr_plot.pdf",comp2plot)))
g
```