

Appendix B: Code

Install Packages

```
# install.packages(c("dplyr", "tidyr", "ggplot2", "plotly", "Hmisc", "corrplot", "dplyr", "corrplot", "factoextra"))

library(dplyr)

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

library(tidyr)
library(ggplot2)
library(plotly)

##
## Attaching package: 'plotly'
## The following object is masked from 'package:ggplot2':
##
##   last_plot
## The following object is masked from 'package:stats':
##
##   filter
## The following object is masked from 'package:graphics':
##
##   layout

library(corrplot)

## corrplot 0.95 loaded

library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(broom)
library(ggrepel)
library(purrr)
library(psych)

##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##      %+%, alpha
```

1. Prepare Data

1.1 Load ABIDE Dataset

```
dt <- load("/Users/macbook/Desktop/ASD/asd_data.RData")
```

1.2 Check Data

```
print(dt)
```

```
## [1] "YALE_fmri"      "YALE_demo_var"
```

“dt”: include 2 datasets, “YALE_fmri” & “YALE_demo_var”

```
summary(YALE_fmri)
```

```
##      Length Class      Mode
## [1,] 110    data.table list
## [2,] 110    data.table list
## [3,] 110    data.table list
## [4,] 110    data.table list
## [5,] 110    data.table list
## [6,] 110    data.table list
## [7,] 110    data.table list
## [8,] 110    data.table list
## [9,] 110    data.table list
## [10,] 110   data.table list
## [11,] 110   data.table list
## [12,] 110   data.table list
## [13,] 110   data.table list
## [14,] 110   data.table list
## [15,] 110   data.table list
## [16,] 110   data.table list
## [17,] 110   data.table list
## [18,] 110   data.table list
## [19,] 110   data.table list
## [20,] 110   data.table list
## [21,] 110   data.table list
## [22,] 110   data.table list
## [23,] 110   data.table list
## [24,] 110   data.table list
## [25,] 110   data.table list
## [26,] 110   data.table list
## [27,] 110   data.table list
## [28,] 110   data.table list
## [29,] 110   data.table list
## [30,] 110   data.table list
## [31,] 110   data.table list
## [32,] 110   data.table list
## [33,] 110   data.table list
```

```
## [34,] 110    data.table list
## [35,] 110    data.table list
## [36,] 110    data.table list
## [37,] 110    data.table list
## [38,] 110    data.table list
## [39,] 110    data.table list
## [40,] 110    data.table list
## [41,] 110    data.table list
## [42,] 110    data.table list
## [43,] 110    data.table list
## [44,] 110    data.table list
## [45,] 110    data.table list
## [46,] 110    data.table list
## [47,] 110    data.table list
```

```
str(YALE_fmri[1])
```

```
## List of 1
## $ :Classes 'data.table' and 'data.frame':  196 obs. of  110 variables:
## ..$ #10 : num [1:196] -4.5 -9.63 -9.44 -3.06 3.74 ...
## ..$ #11 : num [1:196] -4.4 -5.36 -1.24 5.6 8.77 ...
## ..$ #12 : num [1:196] -2.264 -5.716 -4.535 -0.211 2.399 ...
## ..$ #13 : num [1:196] 0.358 -5.676 -10.499 -8.833 -1.03 ...
## ..$ #17 : num [1:196] 1.44 -0.366 -2.541 -3.112 -2.322 ...
## ..$ #18 : num [1:196] 6.421 5.939 -0.181 -7.215 -9.444 ...
## ..$ #26 : num [1:196] -0.181 7.706 16.039 16.447 5.578 ...
## ..$ #49 : num [1:196] -4.873 -7.722 -6.487 -2.653 -0.721 ...
## ..$ #50 : num [1:196] 0.834 1.348 0.934 -1.304 -4.378 ...
## ..$ #51 : num [1:196] 0.38 -1.418 -1.956 -0.576 1.161 ...
## ..$ #52 : num [1:196] -6.5 -8.4 -4.16 2.91 6.48 ...
## ..$ #53 : num [1:196] 1.908 0.556 -2.898 -6.579 -8.034 ...
## ..$ #54 : num [1:196] 3.94 5.69 3.7 -3.01 -10.59 ...
## ..$ #58 : num [1:196] 2.14 10.45 12.12 1.21 -16.99 ...
## ..$ #101 : num [1:196] 0.238 2.316 6.629 9.745 9.953 ...
## ..$ #102 : num [1:196] 0.762 3.215 6.981 9.695 9.305 ...
## ..$ #201 : num [1:196] -2.18 -6.8 -7.11 -1.74 5.48 ...
## ..$ #202 : num [1:196] -0.846 -2.659 -2.585 -1.49 -1.12 ...
## ..$ #301 : num [1:196] -3.18 1.73 9.57 12.65 8.75 ...
## ..$ #302 : num [1:196] 1.1 1.3 -2.15 -6.38 -6.36 ...
## ..$ #401 : num [1:196] -2.63 -1.31 0.483 2.093 3.882 ...
## ..$ #402 : num [1:196] 0.487 4.239 7.314 9.068 8.762 ...
## ..$ #501 : num [1:196] -3.01 -9.37 -7.39 5.78 18.75 ...
## ..$ #502 : num [1:196] 1.578 1.4442 -0.0491 -0.8953 0.693 ...
## ..$ #601 : num [1:196] -4.84 -6.62 -3.62 1.77 2.55 ...
## ..$ #602 : num [1:196] 0.027 1.033 3.849 8.893 11.318 ...
## ..$ #701 : num [1:196] -2.07 -6.82 -9.271 -6.71 -0.718 ...
## ..$ #702 : num [1:196] -3.16 -7.25 -8.33 -4.74 1.63 ...
## ..$ #801 : num [1:196] -1.995 -2.501 -1.198 0.599 0.684 ...
## ..$ #802 : num [1:196] 2.597 2.469 1.499 0.232 -0.139 ...
## ..$ #901 : num [1:196] 3.28 9.04 8.47 -3.63 -23.4 ...
## ..$ #902 : num [1:196] -2.269 0.295 6.314 9.678 4.054 ...
## ..$ #1001 : num [1:196] -3.366 -1.028 1.488 0.748 -4.257 ...
## ..$ #1002 : num [1:196] -1.17 2.09 7.41 10.53 6.84 ...
## ..$ #1101 : num [1:196] -6.645 -6.715 -0.425 6.888 8.167 ...
## ..$ #1102 : num [1:196] -0.632 0.521 3.128 4.614 3.31 ...
```

```

## ..$ #1201: num [1:196] -2.43 -0.925 3.135 5.844 4.145 ...
## ..$ #1202: num [1:196] -1.44 1.6 5.26 7.38 8.08 ...
## ..$ #1301: num [1:196] -0.399 3.497 9.361 12.673 9.42 ...
## ..$ #1302: num [1:196] -0.0423 9.6493 18.4766 21.5258 17.0406 ...
## ..$ #1401: num [1:196] -1.71 -4.86 -3.94 1.34 5.69 ...
## ..$ #1402: num [1:196] -0.715 -1.587 -1.57 -1.066 -0.964 ...
## ..$ #1501: num [1:196] 0.0956 -1.7428 -2.3854 -1.1106 0.5894 ...
## ..$ #1502: num [1:196] 0.58 1.37 2.54 3.21 2.7 ...
## ..$ #1601: num [1:196] 0.993 1.329 -1.238 -5.985 -9.175 ...
## ..$ #1602: num [1:196] 0.917 7.128 11.066 7.97 -0.348 ...
## ..$ #1701: num [1:196] -1.616 -5.946 -9.205 -7.699 -0.828 ...
## ..$ #1702: num [1:196] -4.06 -8.09 -7.86 -3.21 2.59 ...
## ..$ #1801: num [1:196] -2.99 -6.79 -9.56 -9.76 -6.83 ...
## ..$ #1802: num [1:196] -3.438 -1.772 0.207 0.505 -0.666 ...
## ..$ #1901: num [1:196] 1.5 1.11 -4.31 -9.23 -7.34 ...
## ..$ #1902: num [1:196] 3.73 3.53 -2.13 -9.79 -12.42 ...
## ..$ #2001: num [1:196] -0.81 -0.408 -2.49 -5.482 -4.944 ...
## ..$ #2002: num [1:196] 1.88 4.86 5.62 2.11 -4.07 ...
## ..$ #2101: num [1:196] 0.597 2.44 2.775 2.444 4.343 ...
## ..$ #2102: num [1:196] -0.753 7.38 18.395 25.367 22.491 ...
## ..$ #2201: num [1:196] 3.75 2.53 -5.1 -12.6 -12.7 ...
## ..$ #2202: num [1:196] 7.58 10.43 3.49 -7.58 -13.1 ...
## ..$ #2301: num [1:196] 5.47 3.6 -3.63 -9.73 -9.48 ...
## ..$ #2302: num [1:196] 4.32 7.91 5.13 -2.7 -9.87 ...
## ..$ #2401: num [1:196] 4.987 -0.338 -8.34 -12.584 -8.319 ...
## ..$ #2402: num [1:196] 0.0692 -5.4758 -8.9185 -6.2886 1.4442 ...
## ..$ #2501: num [1:196] 4.79 9.42 8.07 -1.64 -13.67 ...
## ..$ #2502: num [1:196] 0.819 7.398 13.331 12.498 4.311 ...
## ..$ #2601: num [1:196] -4.81 -14.74 -19.07 -15.16 -5.97 ...
## ..$ #2602: num [1:196] -0.613 -3.596 -5.917 -6.312 -4.72 ...
## ..$ #2701: num [1:196] 2.995 4.587 0.952 -5.488 -9.211 ...
## ..$ #2702: num [1:196] 1.6 2.13 0.35 -2.65 -4.38 ...
## ..$ #2801: num [1:196] -2.08 2.34 6.31 7.72 8.11 ...
## ..$ #2802: num [1:196] -0.1665 0.0514 -0.369 -0.3922 0.8151 ...
## ..$ #2901: num [1:196] -3.73 -2.86 1.49 4.99 4.67 ...
## ..$ #2902: num [1:196] -4.01 -5.75 -2.55 3.26 6.49 ...
## ..$ #3001: num [1:196] -2.506 -2.895 -0.843 2.591 5.83 ...
## ..$ #3002: num [1:196] 0.938 0.617 -1.733 -4.018 -4.155 ...
## ..$ #3101: num [1:196] 0.767 1.567 -1.055 -6.685 -12.061 ...
## ..$ #3102: num [1:196] 1.4 2.15 -0.8 -5.69 -9.01 ...
## ..$ #3201: num [1:196] 1.563 -1.794 -3.755 -0.644 4.477 ...
## ..$ #3202: num [1:196] 3.73 4.066 0.824 -2.753 -4.089 ...
## ..$ #3301: num [1:196] 1.17 0.596 -0.795 -1.013 1.101 ...
## ..$ #3302: num [1:196] 6.72 7.27 1.62 -4.94 -5.97 ...
## ..$ #3401: num [1:196] -0.884 -1.032 -0.623 0.774 2.76 ...
## ..$ #3402: num [1:196] 2.209 2.53 0.434 -3.183 -5.846 ...
## ..$ #3501: num [1:196] 2.2 2.21 -1.56 -8.37 -13.11 ...
## ..$ #3502: num [1:196] 0.603 -0.878 -5.138 -9.2 -9.554 ...
## ..$ #3601: num [1:196] 0.493 -0.492 -1.723 -4.104 -7.531 ...
## ..$ #3602: num [1:196] 3.36 -1.13 -8.29 -11.39 -7.65 ...
## ..$ #3701: num [1:196] -0.0805 -0.7828 -0.6161 0.7277 1.9565 ...
## ..$ #3702: num [1:196] -1.82 -4.65 -6.23 -4.89 -1.81 ...
## ..$ #3801: num [1:196] -0.812 -1.239 -1.785 -2.605 -3.321 ...
## ..$ #3802: num [1:196] 1.3 0.89 -0.653 -2.347 -3.476 ...

```

```
## ..$ #3901: num [1:196] 4.27 2.48 -4.02 -10.91 -13.85 ...
## ..$ #3902: num [1:196] 2.4 0.98 -3.74 -7.84 -7.63 ...
## ..$ #4001: num [1:196] 6.43 4.57 -3.83 -12.53 -15.41 ...
## ..$ #4002: num [1:196] 4.14 2.96 -3.61 -10.56 -12.83 ...
## ..$ #4101: num [1:196] -5.88 -12.09 -8.2 4.66 15.09 ...
## ..$ #4102: num [1:196] -0.688 -6.489 -8.035 -1.804 6.485 ...
## ..$ #4201: num [1:196] -0.839 -4.584 -5.06 -1.937 1.865 ...
## ..$ #4202: num [1:196] -2.78 -5.567 -5.631 -3.155 -0.526 ...
## ..$ #4301: num [1:196] -4.3639 -5.6637 -3.4183 -0.0868 0.4465 ...
## .. [list output truncated]
## ..- attr(*, ".internal.selfref")=<externalptr>
```

“YALE_fmri”: list of 47 matrices, each 196x110

```
missing_fmri <- sapply(YALE_fmri, function(mat) sum(is.na(mat)))
```

```
cat("\n=== fMRI Data Missing Value Summary ===\n")
```

```
##
## === fMRI Data Missing Value Summary ===
```

```
cat("Total missing values:", sum(missing_fmri), "\n")
```

```
## Total missing values: 0
```

```
cat("Number of subjects with missing values:", sum(missing_fmri > 0), "\n")
```

```
## Number of subjects with missing values: 0
```

```
str(YALE_demo_var)
```

```
## 'data.frame': 47 obs. of 3 variables:
## $ DX_GROUP : Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 2 2 ...
## $ AGE_AT_SCAN: num 15.92 12.75 9.75 8.67 14.42 ...
## $ SEX : Factor w/ 2 levels "1","2": 1 1 1 1 2 1 2 2 1 1 ...
```

“YALE_demo_var”: data frame with 47 rows, 3 variables

```
missing_demo <- colSums(is.na(YALE_demo_var))
```

```
cat("Census Table Missing Value Summary:\n")
```

```
## Census Table Missing Value Summary:
```

```
print(missing_demo)
```

```
## DX_GROUP AGE_AT_SCAN SEX
## 0 0 0
```

There is no missing data

1.3 Time series preprocessing

```
YALE_fmri_processed <- lapply(1:length(YALE_fmri), function(i) {
  subject_mat <- YALE_fmri[[i]]
  subject_mat <- as.matrix(subject_mat)
  storage.mode(subject_mat) <- "double"

  global_mean <- mean(subject_mat)
```

```

subject_mat <- subject_mat - global_mean

min_val <- min(subject_mat)
subject_mat_shifted <- subject_mat - min_val + 1e-3
log_mat <- log(subject_mat_shifted)

processed_mat <- apply(log_mat, 2, function(ts) {
  (ts - mean(ts)) / sd(ts)
})

if (i %in% sample(1:47, 3)) {
  sample_regions <- sample(1:110, 5)
  par(mfrow = c(2, 5), mar = c(3,3,2,1), oma = c(0,0,2,0))
  for (r in sample_regions) {
    hist(subject_mat[, r], breaks = 30, main = paste("Region", r, "-Original"),
          xlab = "", col = "skyblue")
    hist(processed_mat[, r], breaks = 30, main = paste("Region", r, "-After"),
          xlab = "", col = "salmon")
  }
  mtext(paste("Subject", i, "Pretreatment Quality Check "), outer = TRUE, cex = 1.2)
}

return(processed_mat)
})

```

2 Data Informations

```
head(YALE_demo_var)
```

```
##      DX_GROUP AGE_AT_SCAN SEX
## 442         2      15.92   1
## 443         2      12.75   1
## 444         2       9.75   1
## 445         2       8.67   1
## 446         2      14.42   2
## 447         2      10.67   1
```

```
summary(YALE_demo_var)
```

```
##  DX_GROUP  AGE_AT_SCAN    SEX
##  1:21    Min.   : 7.00    1:33
##  2:26    1st Qu.:10.88    2:14
##           Median :13.25
##           Mean   :12.80
##           3rd Qu.:14.42
##           Max.   :17.83
```

Note: DX_GROUP → Diagnosis (1 = Autism, 2 = Control). SEX → Gender (1 = Male, 2 = Female).

```

demo_summary <- YALE_demo_var %>%
  group_by(DX_GROUP = factor(DX_GROUP, labels = c("Autism", "Control"))) %>%
  summarise(
    N = n(),
    Age_Mean = mean(AGE_AT_SCAN, na.rm = TRUE),

```

```

Age_SD = sd(AGE_AT_SCAN, na.rm = TRUE),
Male = sum(SEX == 1),
Female = sum(SEX == 2)
)

print(demo_summary)

## # A tibble: 2 x 6
##   DX_GROUP      N Age_Mean Age_SD  Male Female
##   <fct>      <int>   <dbl> <dbl> <int> <int>
## 1 Autism      21    12.9   3.10    14     7
## 2 Control      26    12.8   2.84    19     7

subject_summaries <- lapply(YALE_fmri_processed, function(mat) {
  apply(mat, 2, function(ts) {
    c(
      Mean = mean(ts),
      SD = sd(ts),
      Q25 = quantile(ts, 0.25),
      Median = quantile(ts, 0.5),
      Q75 = quantile(ts, 0.75)
    )
  }) %>% t() %>% as.data.frame()
})

for (i in 1:length(subject_summaries)) {
  subject_summaries[[i]]$Subject_ID <- i
  subject_summaries[[i]]$Group <- YALE_demo_var$DX_GROUP[i]
}

summary_df <- do.call(rbind, subject_summaries)

summary_df$Group <- factor(summary_df$Group, levels = c(1,2), labels = c("Autism", "Control"))

group_summary <- summary_df %>%
  group_by(Group) %>%
  summarise(across(.cols = -Subject_ID, .fns = list(Mean = mean, SD = sd), .names = "{.col}_{.fn}"))

print(group_summary)

## # A tibble: 2 x 11
##   Group Mean_Mean Mean_SD SD_Mean SD_SD `Q25.25%_Mean` `Q25.25%_SD`
##   <fct>      <dbl>   <dbl>   <dbl>   <dbl>      <dbl>      <dbl>
## 1 Autism  1.12e-12 5.37e-11      1 6.20e-17    -0.604    0.0988
## 2 Control -3.08e-17 3.69e-15      1 6.39e-17    -0.608    0.0978
## # i 4 more variables: `Median.50%_Mean` <dbl>, `Median.50%_SD` <dbl>,
## #   `Q75.75%_Mean` <dbl>, `Q75.75%_SD` <dbl>

```

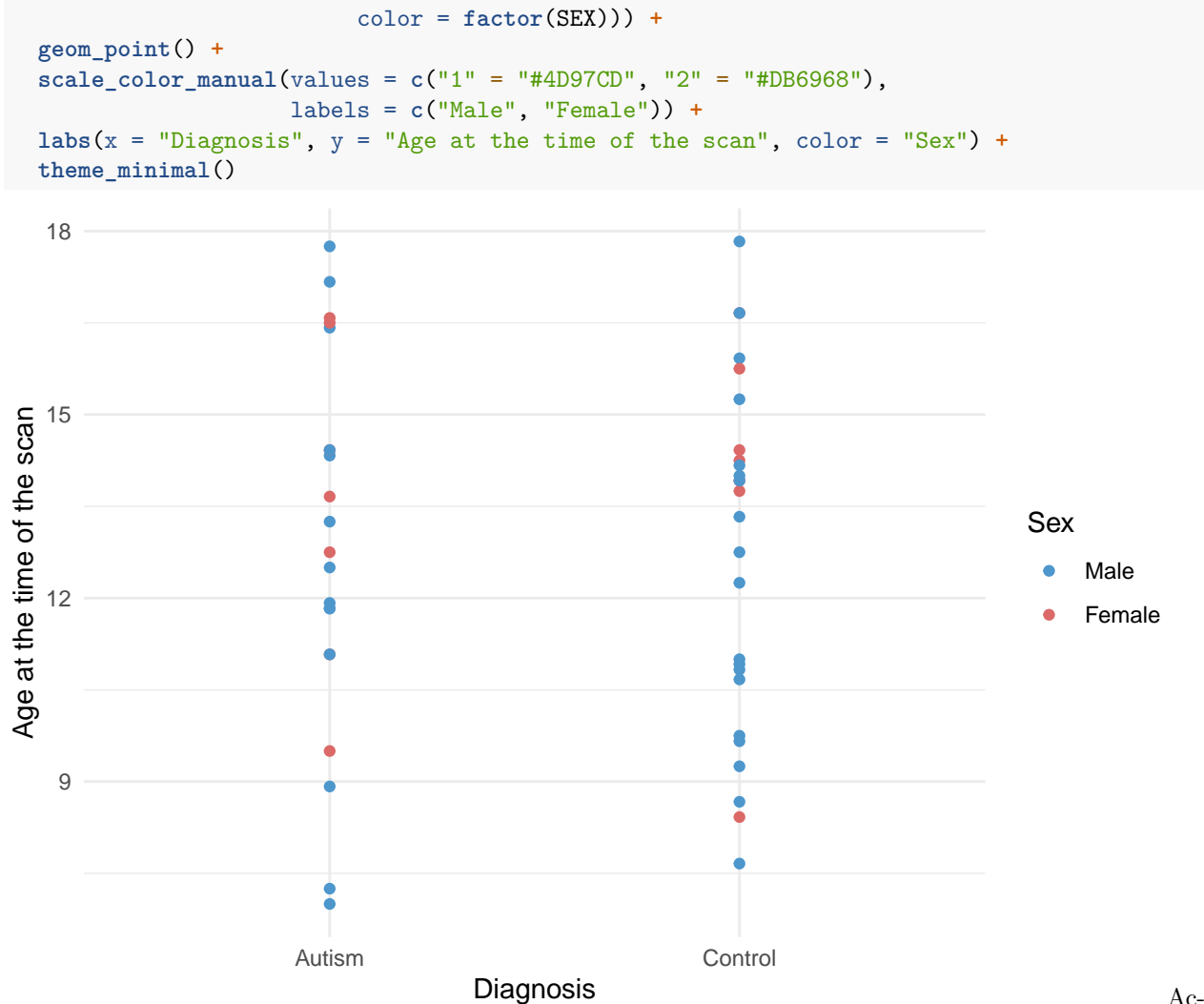
3 Data Analysis

3.1 Analyze by demographic variables

```

ggplot(YALE_demo_var, aes(x = factor(DX_GROUP, labels = c("Autism", "Control")),
  y = AGE_AT_SCAN,

```



According to this graph, it can be preliminarily determined that Autism has no strong association with gender and age. Further verification follows:

```

shapiro_autism <- shapiro.test(YALE_demo_var$AGE_AT_SCAN[YALE_demo_var$DX_GROUP == "1"])
shapiro_control <- shapiro.test(YALE_demo_var$AGE_AT_SCAN[YALE_demo_var$DX_GROUP == "2"])

cat("Autism group normality p-values:", shapiro_autism$p.value, "\n")

```

```
## Autism group normality p-values: 0.5453883
```

```
cat("Control group normality p-values:", shapiro_control$p.value, "\n")
```

```
## Control group normality p-values: 0.4201061
```

```

# If both groups are normal, the T-test is used. Otherwise, the Mann-Whitney U test is used
if (shapiro_autism$p.value > 0.05 & shapiro_control$p.value > 0.05) {
  t_test <- t.test(AGE_AT_SCAN ~ DX_GROUP, data = YALE_demo_var)
  cat("T-test results:\n")
  print(t_test)
} else {
  wilcox_test <- wilcox.test(AGE_AT_SCAN ~ DX_GROUP, data = YALE_demo_var)
  cat("Mann-Whitney U test results:\n")
}

```



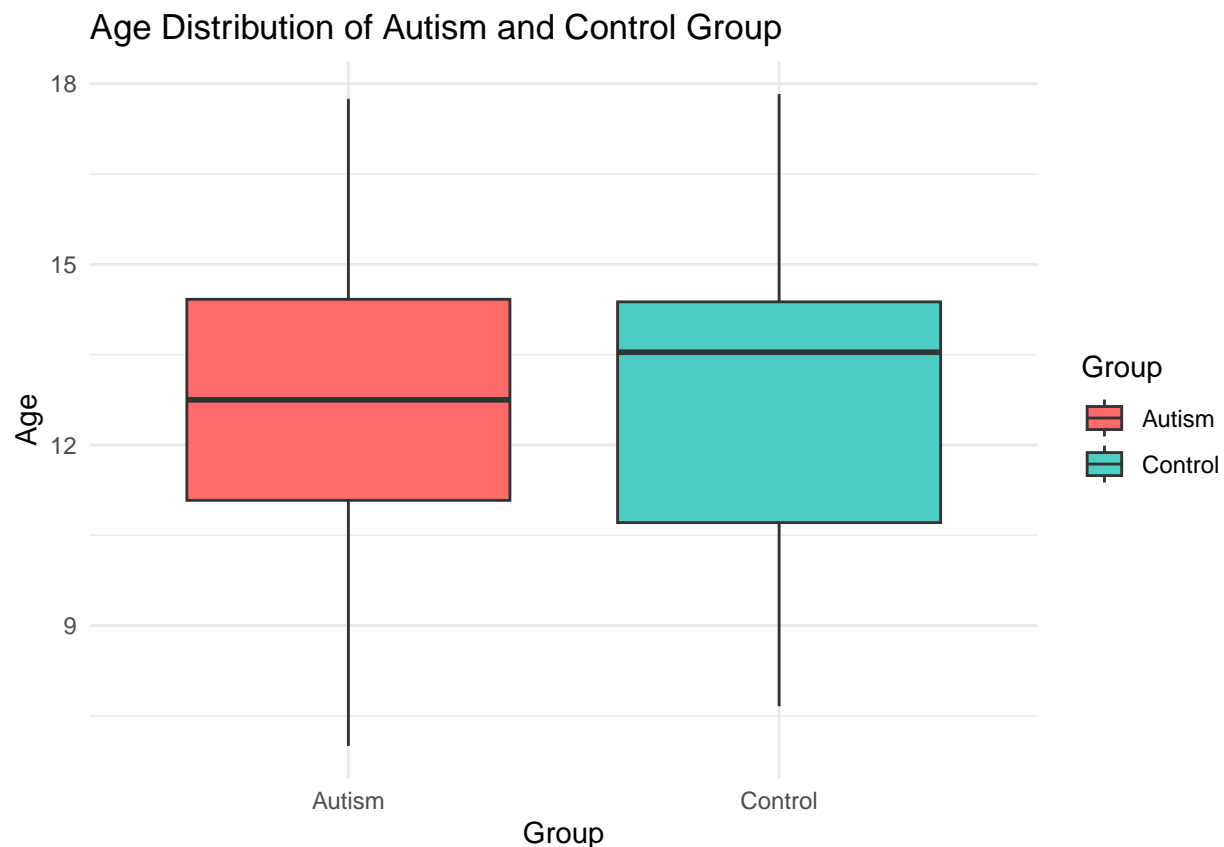
```

    print(wilcox_test)
}

## T-test results:
##
## Welch Two Sample t-test
##
## data: AGE_AT_SCAN by DX_GROUP
## t = 0.12401, df = 41.145, p-value = 0.9019
## alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0
## 95 percent confidence interval:
## -1.659918 1.877134
## sample estimates:
## mean in group 1 mean in group 2
## 12.86476 12.75615

ggplot(YALE_demo_var,
  aes(
    x = factor(DX_GROUP, labels = c("Autism", "Control")),
    y = AGE_AT_SCAN,
    fill = factor(DX_GROUP)
  )) +
  geom_boxplot() +
  scale_fill_manual(
    values = c("#FF6B6B", "#4ECDC4"),
    labels = c("Autism", "Control"),
    name = "Group"
  ) +
  labs(
    x = "Group",
    y = "Age",
    title = "Age Distribution of Autism and Control Group"
  ) +
  theme_minimal()

```



There was no significant difference in age.

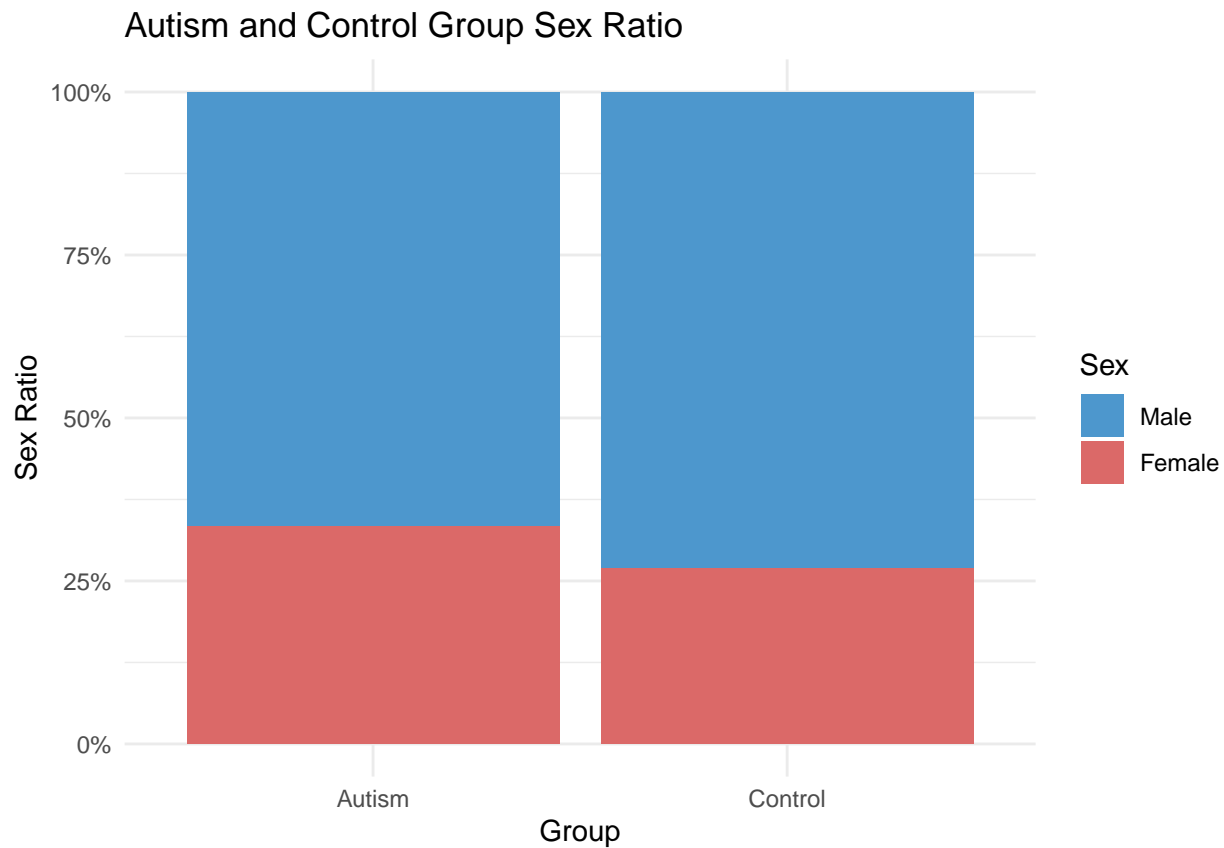
```
sex_table <- table(YALE_demo_var$SEX, YALE_demo_var$DX_GROUP)

# Chi Square test (if expected frequency >= 5)
if (all(chisq.test(sex_table)$expected >= 5)) {
  chisq_test <- chisq.test(sex_table)
  cat("Chi-square test results:\n")
  print(chisq_test)
} else {
  fisher_test <- fisher.test(sex_table)
  cat("Fisher test results:\n")
  print(fisher_test)
}

## Chi-square test results:
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: sex_table
## X-squared = 0.024641, df = 1, p-value = 0.8753

ggplot(YALE_demo_var, aes(x = factor(DX_GROUP, labels = c("Autism", "Control")),
  fill = factor(SEX))) +
  geom_bar(position = "fill") +
  scale_fill_manual(
    values = c("#4D97CD", "#DB6968"),
    labels = c("Male", "Female"),
    name = "Sex"
```

```
) +
scale_y_continuous(labels = scales::percent) +
labs(
  x = "Group",
  y = "Sex Ratio",
  title = "Autism and Control Group Sex Ratio"
) +
theme_minimal()
```



There was no significant difference in sex distribution.

3.2 Consider Brain Activities

3.2.1 Overall Comparison

```
autism_mats <- YALE_fmri[YALE_demo_var$DX_GROUP == 1]
autism_cor_mean <- apply(simplify2array(lapply(autism_mats, cor)), 1:2, mean)

control_mats <- YALE_fmri[YALE_demo_var$DX_GROUP == 2]
control_cor_mean <- apply(simplify2array(lapply(control_mats, cor)), 1:2, mean)

par(
  mfrow = c(1, 2),
  mar = c(0, 0, 5, 0),
  oma = c(0, 0, 2, 0)
)
```

```

add_title <- function(text, cex = 0.8, line = 3.5) {
  title(
    main = text,
    cex.main = cex,
    line = line
  )
}

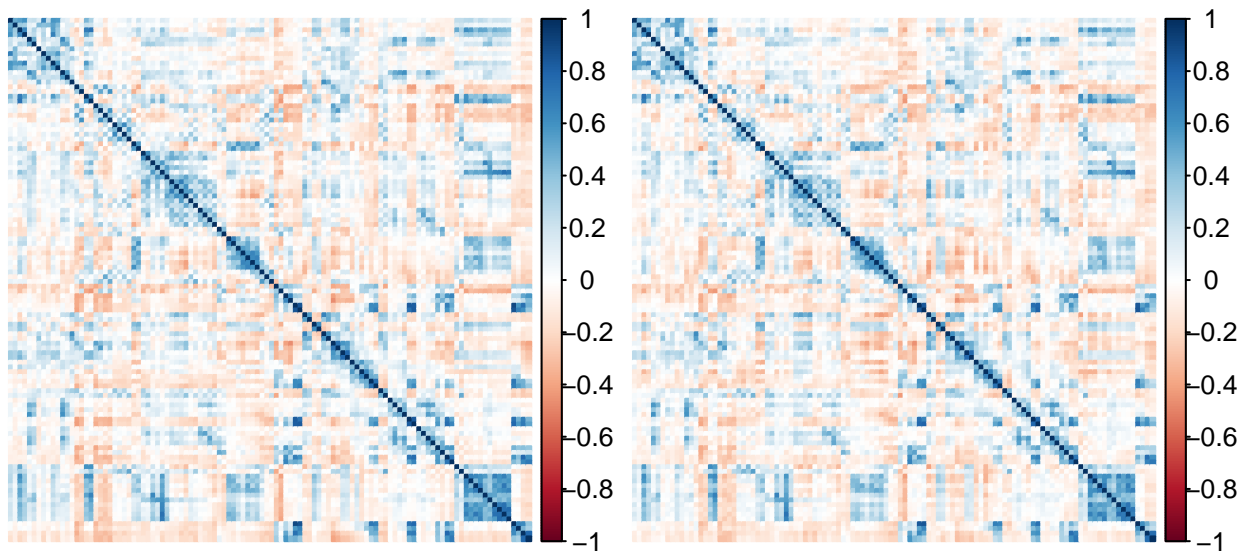
corrplot(
  autism_cor_mean,
  method = "color",
  tl.pos = "n"
)
add_title("Average Connection of Autism Group")

corrplot(
  control_cor_mean,
  method = "color",
  tl.pos = "n"
)
add_title("Average Connection of Control Group")

```

Average Connection of Autism Group

Average Connection of Control Group

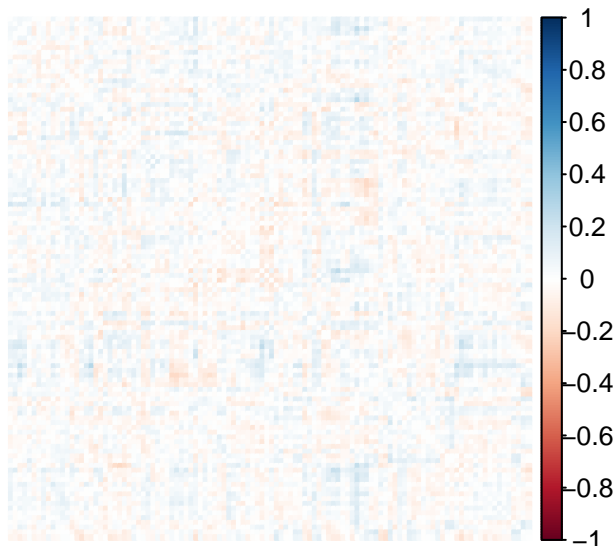


```

diff_matrix <- autism_cor_mean - control_cor_mean
corrplot(
  diff_matrix,
  method = "color",
  tl.pos = "n"
)
add_title("Connection Difference (Autism-Control)")

```

Connection Difference (Autism–Control)



Brain activity was similar between the two groups, however, Autistic group had slightly higher average activity intensity.

3.2.2 Brain Region Compare

```
get_region_mean <- function(mat) {  
  data.frame(  
    Region = 1:ncol(mat),  
    Mean = colMeans(mat, na.rm = TRUE)  
  )  
}  
  
mean_data <- lapply(YALE_fmri, get_region_mean) %>%  
  bind_rows(.id = "SubjectID") %>%  
  mutate(SubjectID = as.numeric(SubjectID))  
  
demo_clean <- YALE_demo_var %>%  
  mutate(  
    SubjectID = row_number(),  
    DX_GROUP = factor(DX_GROUP, labels = c("Autism", "Control"))  
  ) %>%  
  select(SubjectID, DX_GROUP)  
  
mean_data <- mean_data %>%  
  left_join(demo_clean, by = "SubjectID")  
  
region_stats <- mean_data %>%  
  group_by(Region, DX_GROUP) %>%  
  summarise(  
    Group_Mean = mean(Mean, na.rm = TRUE),  
    Group_SD = sd(Mean, na.rm = TRUE),  
    .groups = "drop"  
  ) %>%  
  pivot_wider(  
    id_vars = "Region",  
    names_from = "DX_GROUP",  
    values_from = "Group_Mean"  
  )
```

```

    names_from = DX_GROUP,
    values_from = c(Group_Mean, Group_SD),
    names_glue = "{.value}_{DX_GROUP}"
  ) %>%
  mutate(
    Diff_Mean = Group_Mean_Autism - Group_Mean_Control
  )

all_pvalues <- sapply(1:110, function(r) {
  dat <- mean_data %>% filter(Region == r)
  if (nrow(dat) < 2) return(NA)
  t.test(Mean ~ DX_GROUP, data = dat)$p.value
})

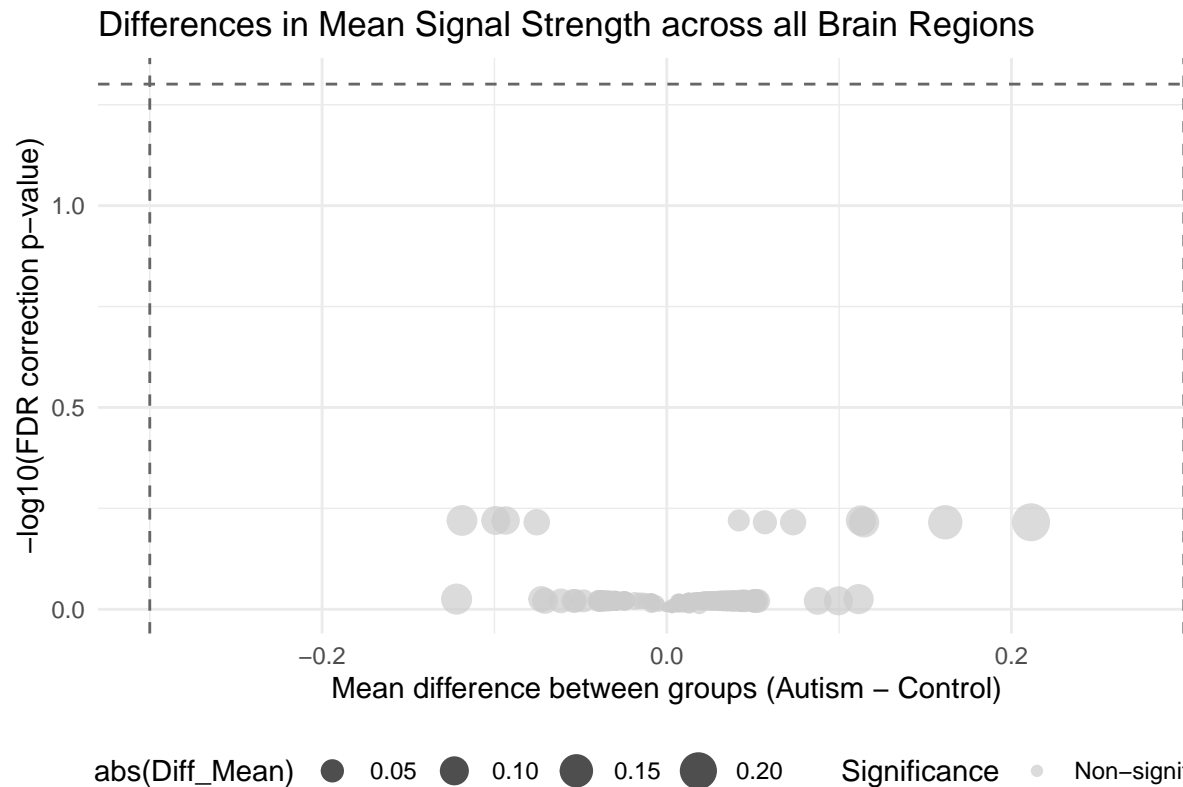
region_stats <- region_stats %>%
  mutate(
    p.value = all_pvalues,
    p.adj = p.adjust(p.value, method = "fdr")
  )

region_stats <- region_stats %>%
  mutate(
    Significance = case_when(
      p.adj < 0.05 & abs(Diff_Mean) > 0.3 ~ "FDR < 0.05 & |Diff| > 0.3",
      p.adj < 0.05 ~ "FDR < 0.05",
      TRUE ~ "Non-significant"
    )
  )

ggplot(region_stats, aes(x = Diff_Mean, y = -log10(p.adj))) +
  geom_point(aes(color = Significance, size = abs(Diff_Mean)), alpha = 0.7) +
  scale_color_manual(
    values = c(
      "FDR < 0.05 & |Diff| > 0.3" = "red",
      "FDR < 0.05" = "orange",
      "Non-significant" = "grey80"
    )
  ) +
  geom_vline(xintercept = c(-0.3, 0.3), linetype = "dashed", color = "grey40") +
  geom_hline(yintercept = -log10(0.05), linetype = "dashed", color = "grey40") +
  geom_text_repel(
    data = subset(region_stats, p.adj < 0.05 & abs(Diff_Mean) > 0.3),
    aes(label = Region),
    size = 3,
    max.overlaps = 20,
    box.padding = 0.5
  ) +
  labs(
    x = "Mean difference between groups (Autism - Control)",
    y = "-log10(FDR correction p-value)",
    title = "Differences in Mean Signal Strength across all Brain Regions ",
    caption = "Dashed line: | difference | > 0.3 and FDR < 0.05"
  ) +

```

```
theme_minimal() +
theme(legend.position = "bottom")
```



Compare by Mean

Dashed line: | difference | > 0.3 and FDR <

```
get_region_range <- function(mat) {
  data.frame(
    Region = 1:ncol(mat),
    Min = apply(mat, 2, min, na.rm = TRUE),
    Max = apply(mat, 2, max, na.rm = TRUE)
  )
}

range_data <- lapply(YALE_fmri, get_region_range) %>%
  bind_rows(.id = "SubjectID") %>%
  mutate(SubjectID = as.numeric(SubjectID))

demo_clean <- YALE_demo_var %>%
  mutate(
    SubjectID = row_number(),
    DX_GROUP = factor(DX_GROUP, labels = c("Autism", "Control"))
  ) %>%
  select(SubjectID, DX_GROUP)

range_data <- range_data %>%
  left_join(demo_clean, by = "SubjectID")
```

```

range_data <- range_data %>%
  mutate(Range = Max - Min)

region_diff <- range_data %>%
  group_by(Region, DX_GROUP) %>%
  summarise(
    Mean_Range = mean(Range, na.rm = TRUE),
    SD_Range = sd(Range, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  pivot_wider(
    names_from = DX_GROUP,
    values_from = c(Mean_Range, SD_Range),
    names_glue = "{.value}_{DX_GROUP}"
  ) %>%
  mutate(
    Diff_Mean = Mean_Range_Autism - Mean_Range_Control
  )

all_pvalues <- sapply(1:110, function(r) {
  dat <- range_data %>% filter(Region == r)
  if (nrow(dat) == 0) return(NA)
  wilcox.test(Range ~ DX_GROUP, data = dat)$p.value
})

region_diff <- region_diff %>%
  mutate(
    p.value = all_pvalues,
    p.adj = p.adjust(p.value, method = "fdr")
  )

region_diff <- region_diff %>%
  mutate(
    Significance = case_when(
      p.adj < 0.05 & abs(Diff_Mean) > 0.5 ~ "FDR < 0.05 & |Diff| > 0.5",
      p.adj < 0.05 ~ "FDR < 0.05",
      TRUE ~ "Non-significant"
    )
  )

ggplot(region_diff, aes(x = Diff_Mean, y = -log10(p.adj))) +
  geom_point(aes(color = Significance, size = abs(Diff_Mean)), alpha = 0.7) +
  scale_color_manual(
    values = c(
      "FDR < 0.05 & |Diff| > 0.5" = "red",
      "FDR < 0.05" = "orange",
      "Non-significant" = "grey80"
    )
  ) +
  geom_vline(xintercept = c(-0.5, 0.5), linetype = "dashed", color = "grey40") +
  geom_hline(yintercept = -log10(0.05), linetype = "dashed", color = "grey40") +
  geom_text_repel(
    data = subset(region_diff, p.adj < 0.05 & abs(Diff_Mean) > 0.5),

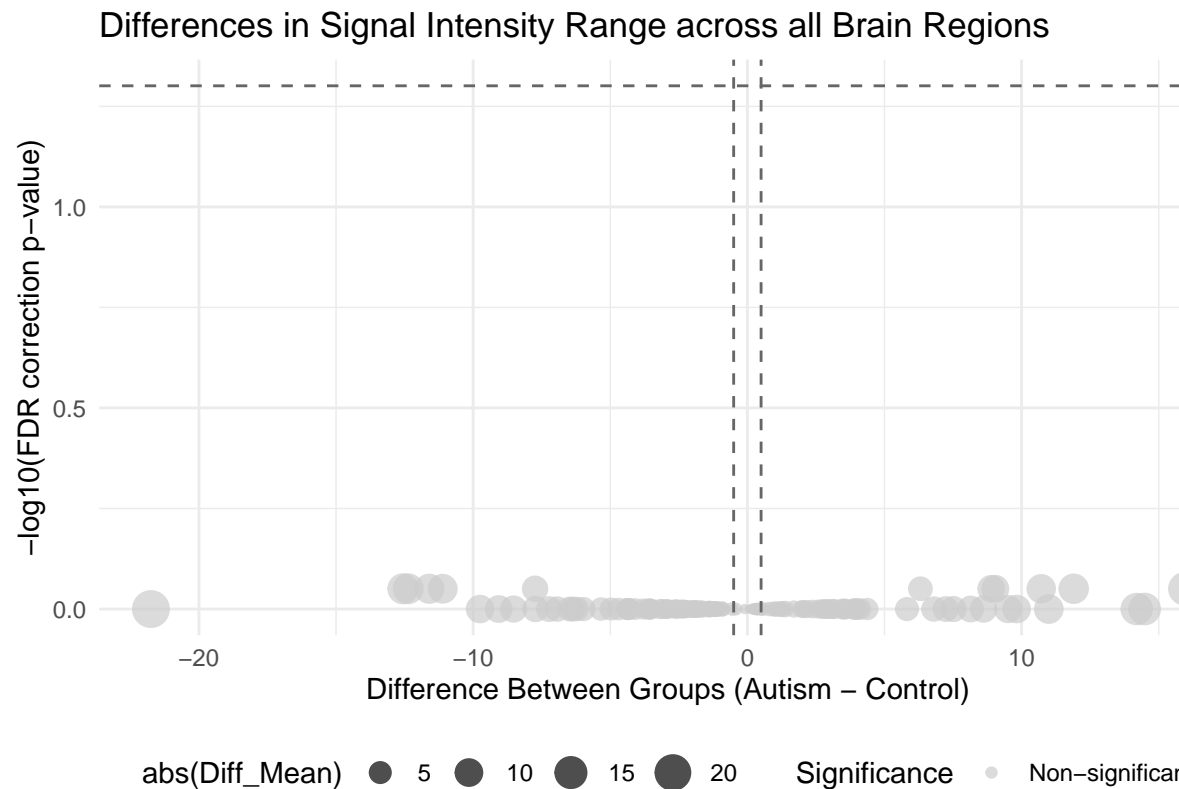
```



```

aes(label = Region),
size = 3,
max.overlaps = 20,
box.padding = 0.5
) +
labs(
  x = "Difference Between Groups (Autism - Control)",
  y = "-log10(FDR correction p-value)",
  title = "Differences in Signal Intensity Range across all Brain Regions",
  caption = "Dashed line: | difference | > 0.5 and FDR < 0.05"
) +
theme_minimal() +
theme(legend.position = "bottom")

```



Compare by Range

Dashed line: | difference | > 0.5 and FDR < 0.05

```

range_data <- range_data %>%
  mutate(Range = Max - Min)

region_stats <- range_data %>%
  group_by(Region, DX_GROUP) %>%
  summarise(
    Mean_Range = mean(Range, na.rm = TRUE),
    SD_Range = sd(Range, na.rm = TRUE)
  ) %>%
  pivot_wider(
    names_from = DX_GROUP,
    values_from = c(Mean_Range, SD_Range)
  )

```

```
## `summarise()` has grouped output by 'Region'. You can override using the
## `.groups` argument.
```

```
region_diff <- region_stats %>%
  mutate(
    Diff_Mean = Mean_Range_Autism - Mean_Range_Control,
    Diff_SD = SD_Range_Autism - SD_Range_Control
  ) %>%
  arrange(desc(abs(Diff_Mean)))

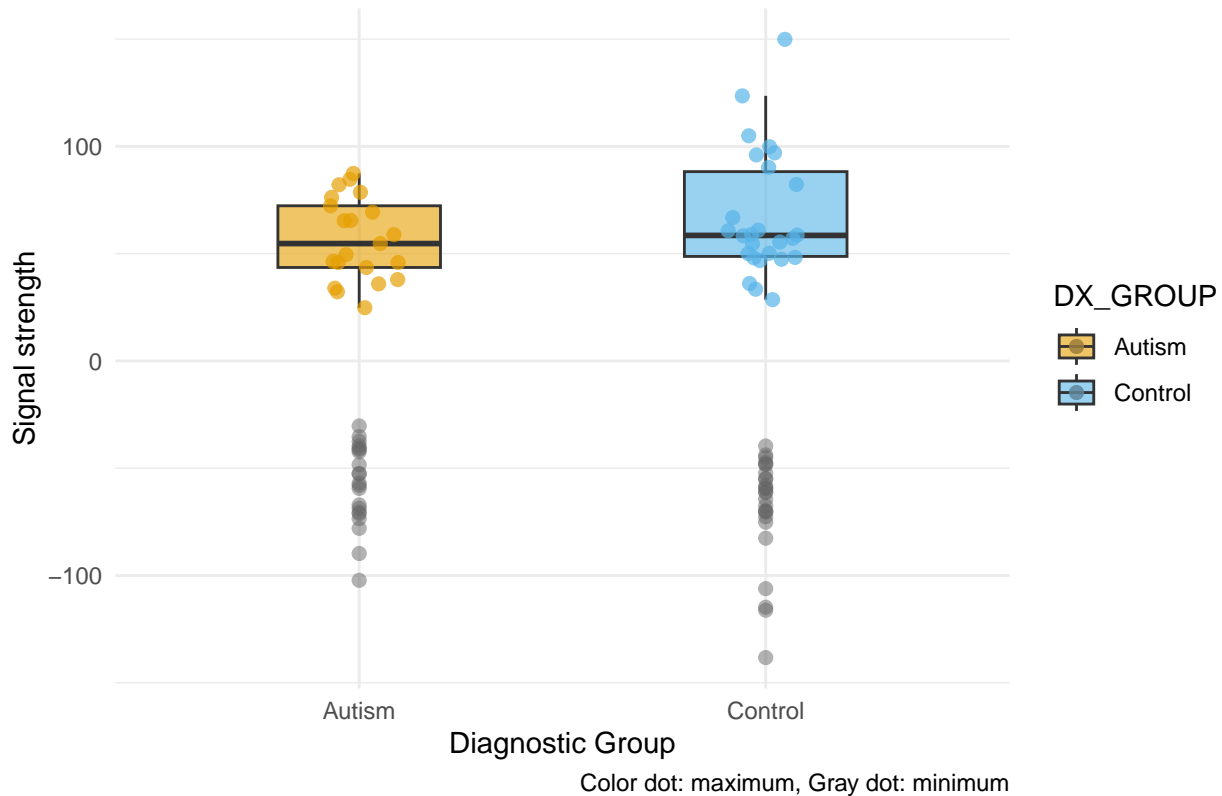
head(region_diff, )
```

```
## # A tibble: 6 x 7
## # Groups:   Region [6]
##   Region Mean_Range_Autism Mean_Range_Control SD_Range_Autism SD_Range_Control
##   <int>         <dbl>         <dbl>         <dbl>         <dbl>
## 1     31          115.          136.          36.7          52.1
## 2     64           70.4          54.5          36.8          32.7
## 3    109          136.          122.          29.7          36.0
## 4    107          150.          136.          47.7          44.0
## 5    102           66.5          79.0          15.5          30.5
## 6      6           64.8          77.2          20.2          26.4
## # i 2 more variables: Diff_Mean <dbl>, Diff_SD <dbl>
```

```
region31 <- range_data %>% filter(Region == 31)
```

```
ggplot(region31, aes(x = DX_GROUP, y = Max, fill = DX_GROUP)) +
  geom_boxplot(width = 0.4, alpha = 0.6, outlier.shape = NA) +
  geom_jitter(aes(color = DX_GROUP), width = 0.1, size = 2, alpha = 0.7) +
  geom_point(aes(y = Min), color = "grey40", size = 2, alpha = 0.5) +
  scale_fill_manual(values = c("Autism" = "#E69F00", "Control" = "#56B4E9")) +
  scale_color_manual(values = c("Autism" = "#E69F00", "Control" = "#56B4E9")) +
  labs(
    title = "Comparison of Signal Intensity Range in Brain Region with the Largest Between-Group Difference",
    x = "Diagnostic Group",
    y = "Signal strength",
    caption = "Color dot: maximum, Gray dot: minimum"
  ) +
  theme_minimal()
```

Comparison of Signal Intensity Range in Brain Region with the Largest Be



```
top_region <- region_diff$Region[1]

target_data <- range_data %>% filter(Region == top_region)

wilcox_result <- wilcox.test(Range ~ DX_GROUP, data = target_data)

cat("Statistical Test Results for the Brain Region with the Largest Between-Group Difference\n")

## Statistical Test Results for the Brain Region with the Largest Between-Group Difference
print(wilcox_result)

##
## Wilcoxon rank sum exact test
##
## data: Range by DX_GROUP
## W = 206, p-value = 0.1561
## alternative hypothesis: true location shift is not equal to 0
```

3.2.3 Brain Regions Connection Compare

```
compute_fc <- function(processed_mat) {
  stopifnot(all(abs(colMeans(processed_mat)) < 1e-6))
  stopifnot(all(abs(apply(processed_mat, 2, sd) - 1) < 1e-3))

  cor_mat <- cor(processed_mat)
  z_mat <- 0.5 * log((1 + cor_mat) / (1 - cor_mat))
  diag(z_mat) <- 0
}
```

```

    return(z_mat)
}

fc_mats <- lapply(YALE_fmri_processed, compute_fc)

demo_clean <- YALE_demo_var %>%
  mutate(
    SubjectID = row_number(),
    DX_GROUP = factor(DX_GROUP, labels = c("Autism", "Control"))
  ) %>%
  select(SubjectID, DX_GROUP)

autism_mats <- fc_mats[demo_clean$DX_GROUP == "Autism"]
control_mats <- fc_mats[demo_clean$DX_GROUP == "Control"]

autism_mean <- apply(simplify2array(autism_mats), 1:2, mean)
control_mean <- apply(simplify2array(control_mats), 1:2, mean)

epsilon <- 1e-6
relative_diff <- (autism_mean - control_mean) /
  ((abs(autism_mean) + abs(control_mean))/2 + epsilon)

relative_diff[lower.tri(relative_diff, diag = TRUE)] <- NA

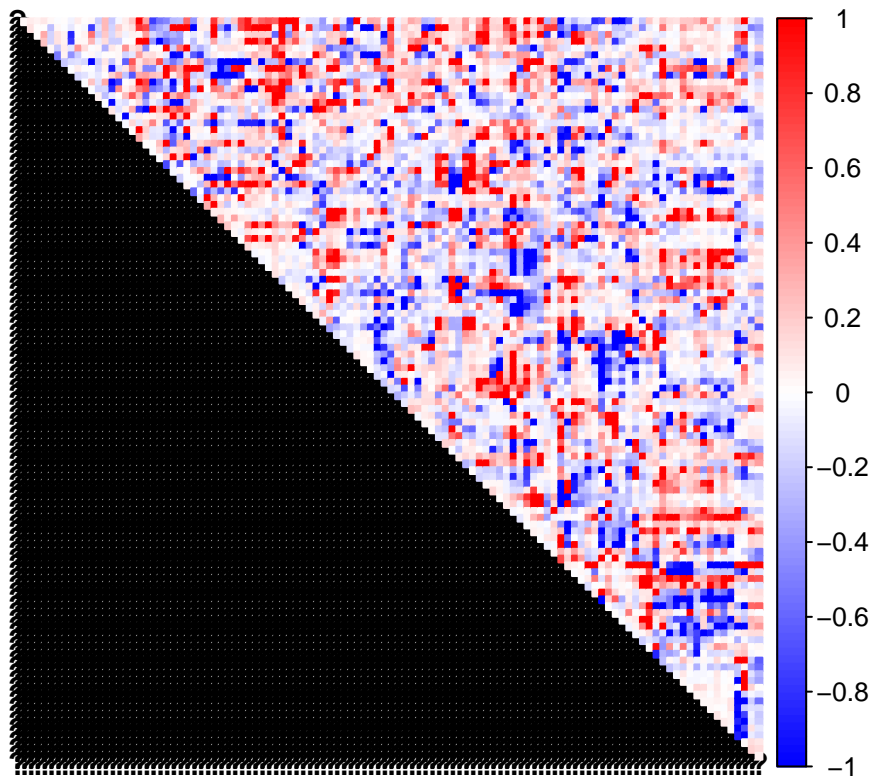
actual_min <- floor(min(relative_diff, na.rm = TRUE))
actual_max <- ceiling(max(relative_diff, na.rm = TRUE))
scale_factor <- max(abs(actual_min), abs(actual_max))
relative_diff_scaled <- relative_diff / scale_factor

col_palette <- colorRampPalette(c("blue", "white", "red"))(100)

corrplot(
  relative_diff_scaled,
  method = "color",
  col = col_palette,
  tl.pos = "n",
  col.lim = c(-1, 1),
  title = "Standardized Differences between Brain Regions Connection",
  mar = c(0, 0, 2, 0))

```

Standardized Differences between Brain Regions Connection



```
compute_fc <- function(mat) {  
  cor(mat)  
}  
  
fc_mats <- lapply(YALE_fmri, compute_fc)  
  
demo_clean <- YALE_demo_var %>%  
  mutate(  
    SubjectID = row_number(),  
    DX_GROUP = factor(DX_GROUP, labels = c("Autism", "Control"))  
  ) %>%  
  select(SubjectID, DX_GROUP)  
  
extract_connections <- function(mat) {  
  data.frame(  
    from = rep(1:110, each = 110),  
    to = rep(1:110, times = 110),  
    strength = as.vector(mat)  
  ) %>%  
  filter(from < to)  
}  
  
all_connections <- lapply(fc_mats, extract_connections) %>%  
  bind_rows(.id = "SubjectID") %>%  
  mutate(SubjectID = as.numeric(SubjectID)) %>%
```

```

left_join(demo_clean, by = "SubjectID")

connection_stats <- all_connections %>%
  group_by(from, to, DX_GROUP) %>%
  summarise(
    mean_strength = mean(strength, na.rm = TRUE),
    sd_strength = sd(strength, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  pivot_wider(
    names_from = DX_GROUP,
    values_from = c(mean_strength, sd_strength),
    names_glue = "{.value}_{DX_GROUP}"
  ) %>%
  mutate(
    diff_strength = mean_strength_Autism - mean_strength_Control
  )

p_values <- sapply(1:nrow(connection_stats), function(i) {
  conn <- connection_stats[i, ]
  autism_vals <- all_connections %>%
    filter(from == conn$from, to == conn$to, DX_GROUP == "Autism") %>%
    pull(strength)
  control_vals <- all_connections %>%
    filter(from == conn$from, to == conn$to, DX_GROUP == "Control") %>%
    pull(strength)
  if (length(autism_vals) < 2 || length(control_vals) < 2) return(NA)
  t.test(autism_vals, control_vals)$p.value
})

connection_stats <- connection_stats %>%
  mutate(
    p.value = p_values,
    p.adj = p.adjust(p.value, method = "fdr")
  )

connection_stats <- connection_stats %>%
  mutate(
    pooled_sd = sqrt(
      (sd_strength_Autism^2 * (sum(demo_clean$DX_GROUP == "Autism") - 1) +
       sd_strength_Control^2 * (sum(demo_clean$DX_GROUP == "Control") - 1)) /
      (sum(demo_clean$DX_GROUP == "Autism") + sum(demo_clean$DX_GROUP == "Control") - 2)
    ),
    cohen_d = diff_strength / pooled_sd
  )

top_connections <- connection_stats %>%
  arrange(p.value, desc(abs(cohen_d))) %>%
  slice_head(n = 10)

print(top_connections %>% select(from, to, diff_strength, cohen_d, p.value))

```

Sorted by uncorrected p-value and effect size

```
## # A tibble: 10 x 5
##   from to diff_strength cohen_d p.value
##   <int> <int>      <dbl>   <dbl>   <dbl>
## 1    18   74      0.283     1.44 0.0000111
## 2     3   74      0.246     1.31 0.0000325
## 3    74   95      0.235     1.29 0.0000438
## 4    74   98      0.209     1.20 0.000131
## 5    45   56     -0.189    -1.22 0.000182
## 6    35   74     -0.156    -1.15 0.000234
## 7    81   94      0.183     1.14 0.000253
## 8    40   72      0.213     1.10 0.000446
## 9    12   39      0.141     1.09 0.000498
## 10   74   97      0.198     1.07 0.000572
```

```
top_connections <- connection_stats %>%
  arrange(desc(abs(cohen_d))) %>%
  slice_head(n = 10)

print(top_connections %>% select(from, to, cohen_d, p.value, p.adj))
```

Sorted by effect size

```
## # A tibble: 10 x 5
##   from to cohen_d p.value p.adj
##   <int> <int>   <dbl>   <dbl> <dbl>
## 1    18   74     1.44 0.0000111 0.0664
## 2     3   74     1.31 0.0000325 0.0875
## 3    74   95     1.29 0.0000438 0.0875
## 4    45   56    -1.22 0.000182 0.216
## 5    74   98     1.20 0.000131 0.196
## 6    35   74    -1.15 0.000234 0.216
## 7     1   40     1.15 0.000588 0.260
## 8    81   94     1.14 0.000253 0.216
## 9    40   72     1.10 0.000446 0.260
## 10     8   26     1.10 0.000645 0.260
```

```
p_values <- sapply(1:nrow(connection_stats), function(i) {
  conn <- connection_stats[i, ]
  autism_vals <- all_connections %>%
    filter(from == conn$from, to == conn$to, DX_GROUP == "Autism") %>%
    pull(strength)
  control_vals <- all_connections %>%
    filter(from == conn$from, to == conn$to, DX_GROUP == "Control") %>%
    pull(strength)
  if (length(autism_vals) < 2 || length(control_vals) < 2) return(NA)
  wilcox.test(autism_vals, control_vals)$p.value
})

connection_stats <- connection_stats %>%
  mutate(
    p.value = p_values,
    p.adj = p.adjust(p.value, method = "fdr")
  )
```

```

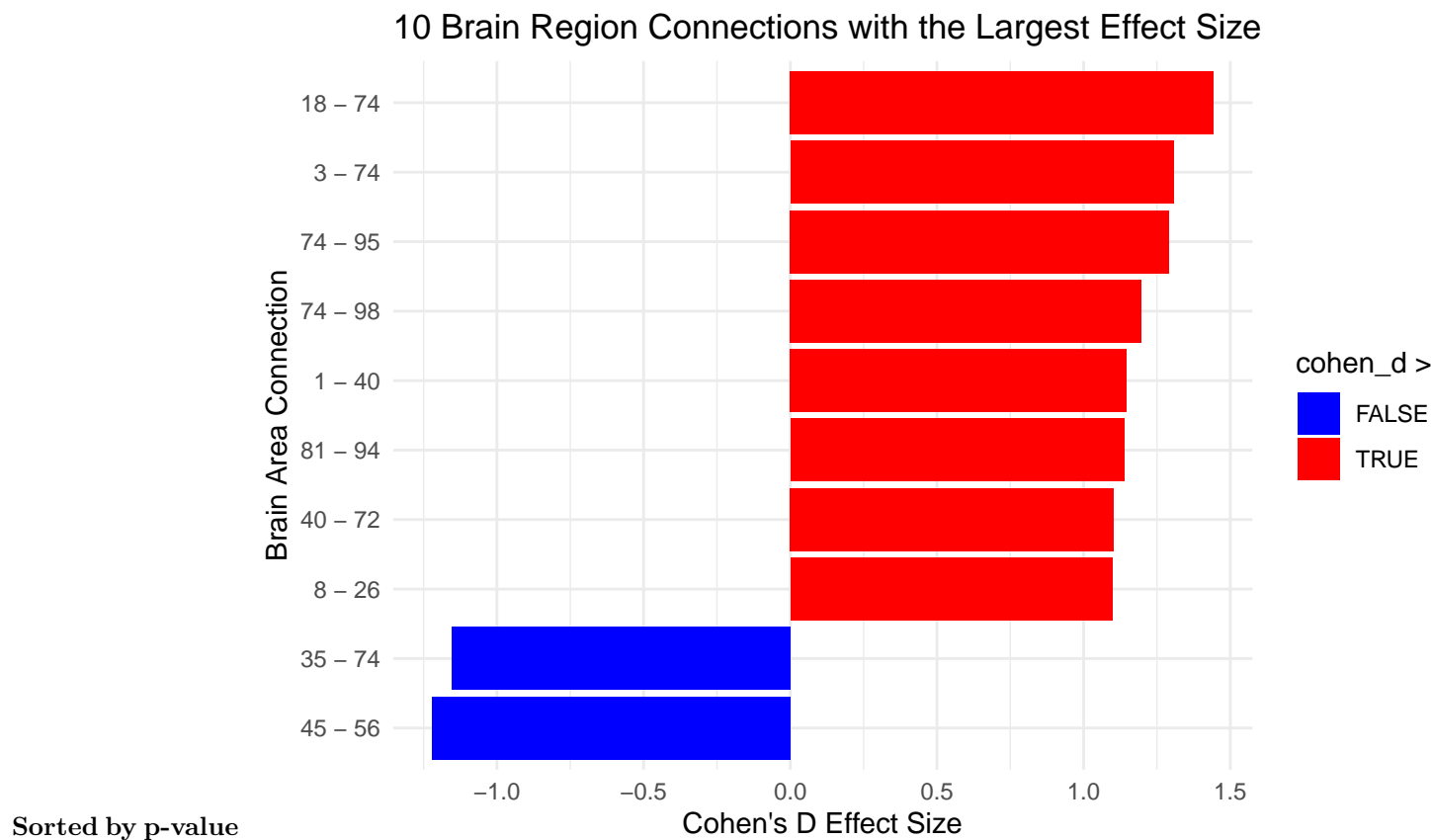
)

top_connections <- connection_stats %>%
  filter(p.adj < 0.1) %>%
  arrange(p.adj, desc(abs(cohen_d))) %>%
  slice_head(n = 10)

top_effects <- connection_stats %>%
  arrange(desc(abs(cohen_d))) %>%
  slice_head(n = 10)

ggplot(top_effects, aes(x = reorder(paste(from, "-", to), cohen_d),
  y = cohen_d, fill = cohen_d > 0)) +
  geom_col() +
  scale_fill_manual(values = c("blue", "red")) +
  labs(
    x = "Brain Area Connection",
    y = "Cohen's D Effect Size",
    title = "10 Brain Region Connections with the Largest Effect Size"
  ) +
  coord_flip() +
  theme_minimal()

```



4 Verification

```
all_z <- unlist(lapply(fc_mats, function(mat) mat[upper.tri(mat)]))

ggplot(data.frame(Z = all_z), aes(x = Z)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "skyblue") +
  stat_function(fun = dnorm, args = list(mean = mean(all_z), sd = sd(all_z)),
               color = "red", size = 1) +
  labs(title = "Functional Connection Fisher Z value distribution",
       subtitle = "Red curve is normal distribution reference line") +
  theme_minimal()
```

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
generated.

Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
i Please use `after_stat(density)` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
generated.

