MANOVA III

Mike Strube

November 28, 2018

1 Preliminaries

The RStudio workspace and console panes are cleared of old output, variables, and other miscellaneous debris. Then some packages are loaded and the required data files are input.

1.1 Clear the Console Panes and Load Packages

```
options(replace.assign = TRUE, width = 65, digits = 4, scipen = 4, fig.width = 4,
    fig.height = 4)
# Clear the workspace and console.
rm(list = ls(all = TRUE))
cat("\f")
```

```
# Turn off showing of significance asterisks.
options(show.signif.stars = F)
# Set the contrast option; important for ANOVAs.
options(contrasts = c("contr.sum", "contr.poly"))
how_long <- Sys.time()
set.seed(123)
library(knitr)
```

```
library(psych)

## Warning:  package 'psych' was built under R version 3.5.1

library(ggplot2)

## Warning:  package 'ggplot2' was built under R version 3.5.1
##
## Attaching package:  'ggplot2'
## The following objects are masked from 'package:psych':
##
##     %+%, alpha

library(MASS)
library(sciplot)
library(dplyr)

## Warning:  package 'dplyr' was built under R version 3.5.1
##
## Attaching package:  'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(aod)
library(MVN)
```

```
## sROC 0.1-2 loaded
```

```
library(boot)
```

```
##
## Attaching package: 'boot'
## The following object is masked from 'package:psych':
##
##     logit
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 3.5.1
## Loading required package: carData
##
## Attaching package: 'car'
## The following object is masked from 'package:boot':
##
##     logit
## The following object is masked from 'package:dplyr':
##
##     recode
## The following object is masked from 'package:psych':
##
##     logit
```

```
library(LogisticDx)
library(biotools)
```

```
## Loading required package: rpanel
## Loading required package: tcltk
## Package 'rpanel', version 1.1-4: type help(rpanel) for summary information
##
## Attaching package: 'rpanel'
## The following object is masked from 'package:boot':
##
##     poisons
## Loading required package: tkrplot
## Loading required package: lattice
##
## Attaching package: 'lattice'
## The following object is masked from 'package:boot':
##
##     melanoma
```

```
## Loading required package:  SpatialEpi
## Loading required package:  sp

## ---
## biotools version 3.1

##

library(multcomp)

## Loading required package:  mvtnorm
## Loading required package:  survival
##
## Attaching package:  'survival'
## The following object is masked from 'package:boot':
##
##     aml
## The following object is masked from 'package:aod':
##
##     rats
## Loading required package:  TH.data
##
## Attaching package:  'TH.data'
## The following object is masked from 'package:MASS':
##
##     geyser

library(candisc)

## Loading required package:  heplots
##
## Attaching package:  'heplots'
## The following object is masked from 'package:biotools':
##
##     boxM
##
## Attaching package:  'candisc'
## The following object is masked from 'package:stats':
##
##     cancor

library(ez)
library(GGally)

##
## Attaching package:  'GGally'
## The following object is masked from 'package:dplyr':
##
##     nasa

library(qqplotr)

##
## Attaching package:  'qqplotr'
## The following objects are masked from 'package:ggplot2':
##
##     stat_qq_line, StatQqLine
```

```r
library(gridExtra)

##
## Attaching package:  'gridExtra'
## The following object is masked from 'package:dplyr':
##
##     combine

library(reshape)

##
## Attaching package:  'reshape'
## The following object is masked from 'package:dplyr':
##
##     rename

library(emmeans)

## Warning:  package 'emmeans' was built under R version 3.5.1
## NOTE: As of emmeans versions > 1.2.3,
##       The 'cld' function will be deprecated in favor of 'CLD'.
##       You may use 'cld' only if you have package:multcomp attached.
##
## Attaching package:  'emmeans'
## The following object is masked from 'package:GGally':
##
##     pigs
## The following object is masked from 'package:multcomp':
##
##     cld

library(profileR)

## Warning:  package 'profileR' was built under R version 3.5.1
## Loading required package:  RColorBrewer
## Loading required package:  lavaan
## This is lavaan 0.6-1
## lavaan is BETA software!  Please report any bugs.
##
## Attaching package:  'lavaan'
## The following object is masked from 'package:psych':
##
##     cor2cov

library(Rmisc)

## Loading required package:  plyr
## --------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## --------------------------------
##
## Attaching package:  'plyr'
## The following objects are masked from 'package:reshape':
##
##     rename, round_any
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename,
##     summarise, summarize
```

1.2 Data

```
setwd("C:\\Courses\\Psychology 516\\PowerPoint\\2018")

# Get the data for the main MANOVA examples.
Skills <- read.table("manova.csv", sep = ",", header = TRUE)
Skills <- as.data.frame(Skills)

# Get the data for profile analyses.
Profiles <- read.table("profile.csv", sep = ",", header = TRUE)
Profiles <- as.data.frame(Profiles)
Profiles <- Profiles[which(Profiles$agemate != 6), ]
Profiles <- Profiles[order(Profiles$agemate), ]
Profiles$AgeMate[Profiles$agemate == "1"] <- "Younger"
Profiles$AgeMate[Profiles$agemate == "2"] <- "Older"
Profiles$AgeMate[Profiles$agemate == "3"] <- "Same"
Profiles$AgeMate <- as.factor(Profiles$AgeMate)

# Get the data for the doubly multivariate example.
Double <- read.table("doubly_multivariate.csv", sep = ",", header = TRUE)
Double <- as.data.frame(Double)
Double$Group <- as.factor(Double$Group)
Double$Group2[Double$Group == "1"] <- "Healthy Controls"
Double$Group2[Double$Group == "2"] <- "Patients"
Double$Group2 <- as.factor(Double$Group2)
```

1.3 Data Modifications

Residualized versions of continuous predictors are created so that preliminary analyses are not contaminated by outcome differences. Labeled variables are created to assist in creation of some tables and graphs. Dummy codes and linear combinations are created for specialized analyses (not all used here).

```
# Residuals
Skills$P_Verbal_R <- lm(P_Verbal ~ as.factor(Group), data = Skills)$residuals
Skills$P_Quant_R <- lm(P_Quant ~ as.factor(Group), data = Skills)$residuals
Skills$C_Verbal_R <- lm(C_Verbal ~ as.factor(Group), data = Skills)$residuals
Skills$C_Quant_R <- lm(C_Quant ~ as.factor(Group), data = Skills)$residuals

# Labels
Skills$Tx_P2[Skills$Tx_P == "1"] <- "No Paper Tx"
Skills$Tx_P2[Skills$Tx_P == "2"] <- "Paper Tx"

Skills$Tx_C2[Skills$Tx_C == "1"] <- "No Computer Tx"
Skills$Tx_C2[Skills$Tx_C == "2"] <- "Computer Tx"
```

```r
Skills$Group2[Skills$Group == "1"] <- "No Paper Tx and No Computer Tx"
Skills$Group2[Skills$Group == "2"] <- "Paper Tx and No Computer Tx"
Skills$Group2[Skills$Group == "3"] <- "No Paper Tx and Computer Tx"
Skills$Group2[Skills$Group == "4"] <- "Paper Tx and Computer Tx"

Skills$Group3[Skills$Group == "1"] <- "No P, No C"
Skills$Group3[Skills$Group == "2"] <- "P, No C"
Skills$Group3[Skills$Group == "3"] <- "No P, C"
Skills$Group3[Skills$Group == "4"] <- "P, C"

# Dummy variables to be used in between-groups analyses.
Skills$D1[Skills$Group == 1] <- 1
Skills$D2[Skills$Group == 1] <- 0
Skills$D3[Skills$Group == 1] <- 0
Skills$D4[Skills$Group == 1] <- 0
Skills$D1[Skills$Group == 2] <- 0
Skills$D2[Skills$Group == 2] <- 1
Skills$D3[Skills$Group == 2] <- 0
Skills$D4[Skills$Group == 2] <- 0
Skills$D1[Skills$Group == 3] <- 0
Skills$D2[Skills$Group == 3] <- 0
Skills$D3[Skills$Group == 3] <- 1
Skills$D4[Skills$Group == 3] <- 0
Skills$D1[Skills$Group == 4] <- 0
Skills$D2[Skills$Group == 4] <- 0
Skills$D3[Skills$Group == 4] <- 0
Skills$D4[Skills$Group == 4] <- 1

# Add contrast codes to reflect main effects and interactions.
Skills$C1[Skills$Group == 1] <- -1
Skills$C2[Skills$Group == 1] <- -1
Skills$C3[Skills$Group == 1] <- 1
Skills$C1[Skills$Group == 2] <- 1
Skills$C2[Skills$Group == 2] <- -1
Skills$C3[Skills$Group == 2] <- -1
Skills$C1[Skills$Group == 3] <- -1
Skills$C2[Skills$Group == 3] <- 1
Skills$C3[Skills$Group == 3] <- -1
Skills$C1[Skills$Group == 4] <- 1
Skills$C2[Skills$Group == 4] <- 1
Skills$C3[Skills$Group == 4] <- 1

# Add contrast codes to reflect specialized comparisons.
Skills$S1[Skills$Group == 1] <- 3
Skills$S2[Skills$Group == 1] <- 0
Skills$S3[Skills$Group == 1] <- 0
Skills$S1[Skills$Group == 2] <- -1
Skills$S2[Skills$Group == 2] <- 2
Skills$S3[Skills$Group == 2] <- 0
Skills$S1[Skills$Group == 3] <- -1
Skills$S2[Skills$Group == 3] <- -1
Skills$S3[Skills$Group == 3] <- 1
Skills$S1[Skills$Group == 4] <- -1
```

```r
Skills$S2[Skills$Group == 4] <- -1
Skills$S3[Skills$Group == 4] <- -1

# Outcome linear combinations to be used in repeated measures
# analyses.
Skills$Sum <- Skills$P_Verbal + Skills$P_Quant + Skills$C_Verbal +
    Skills$C_Quant
Skills$Domain <- Skills$P_Verbal - Skills$P_Quant + Skills$C_Verbal -
    Skills$C_Quant
Skills$Mode <- Skills$P_Verbal + Skills$P_Quant - Skills$C_Verbal -
    Skills$C_Quant
Skills$DxM <- Skills$P_Verbal - Skills$P_Quant - Skills$C_Verbal +
    Skills$C_Quant

# Create a non-factor version of the condition variables before
# converting them to factors.
Skills$Tx_P_NF <- Skills$Tx_P
Skills$Tx_C_NF <- Skills$Tx_C

# Convert to factors
Skills$Tx_P = factor(Skills$Tx_P, levels = c(1, 2), labels = c("No Tx(P)",
    "Tx(P)"))
Skills$Tx_C = factor(Skills$Tx_C, levels = c(1, 2), labels = c("No Tx(C)",
    "Tx(C)"))

# Sort file by Group
Skills <- Skills[order(Skills$Group), ]
```

2 Multivariate Assumptions and Diagnostics

2.1 Multivariate Normality

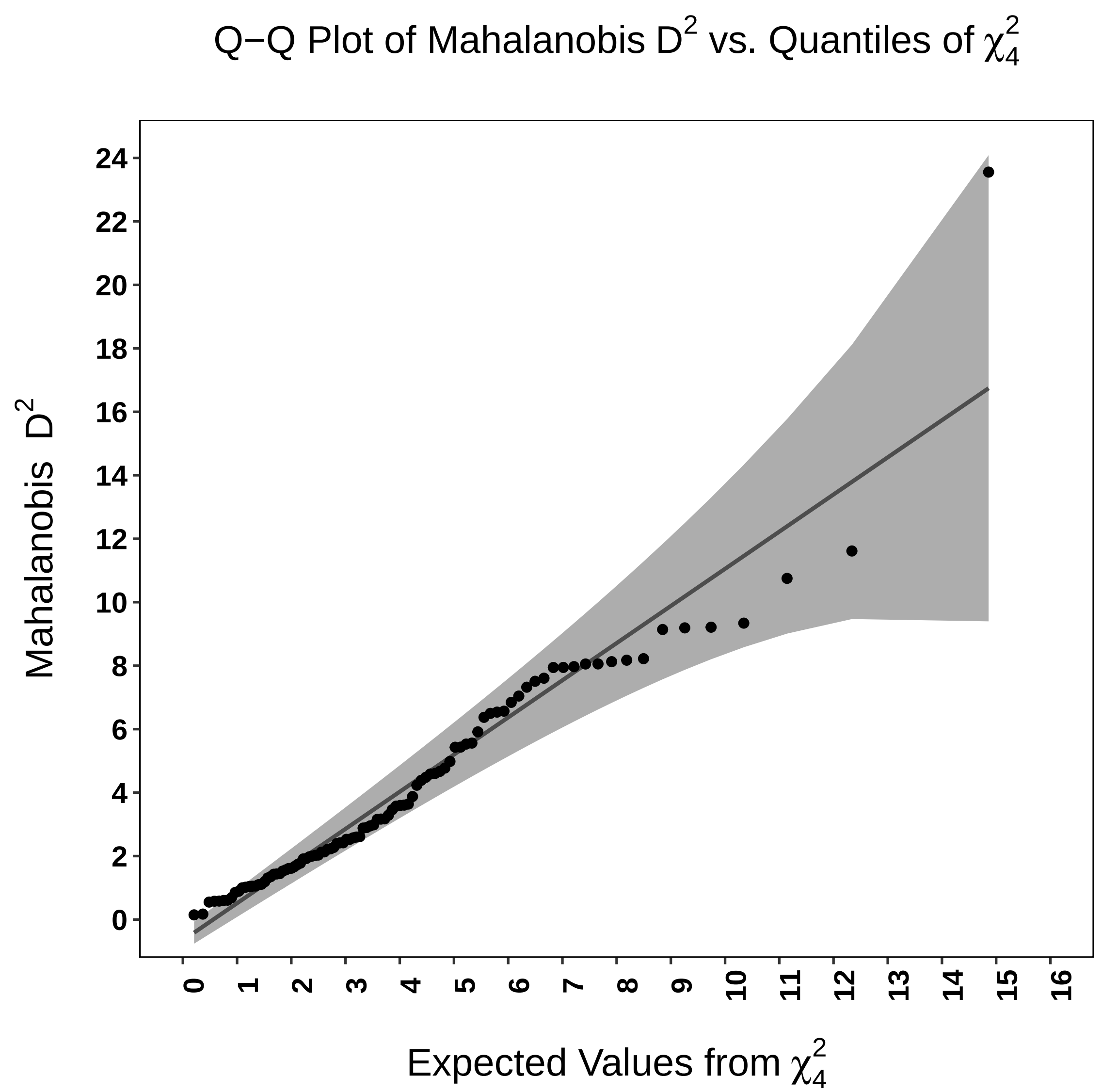
The classification part of discriminant analysis (as well as any significance tests for the discriminant functions) rely on the multivariate normality assumption. Because MANOVA is inherently a discriminant analysis, we make the same assumption. The tests are performed on the residualized data so that group differences do not affect the results. Note that a violation of multivariate normality will also affect the test of homogeneity of covariance matrices.

2.2 Full Sample

```
mvn(Skills[, 9:12], mvnTest = "mardia")

## $multivariateNormality
##              Test        Statistic            p value Result
## 1 Mardia Skewness  33.714208460761 0.0281242141329968     NO
## 2 Mardia Kurtosis 2.67851447156136 0.0073949536550868     NO
## 3             MVN             <NA>               <NA>     NO
##
## $univariateNormality
##           Test   Variable Statistic   p value Normality
## 1 Shapiro-Wilk P_Verbal_R    0.9857    0.3545    YES
## 2 Shapiro-Wilk P_Quant_R     0.9843    0.2825    YES
## 3 Shapiro-Wilk C_Verbal_R    0.9881    0.5174    YES
## 4 Shapiro-Wilk C_Quant_R     0.9867    0.4203    YES
##
## $Descriptives
##              n       Mean Std.Dev  Median    Min   Max   25th
## P_Verbal_R 100 -2.821e-16  10.248 -1.2840 -34.55 26.09 -5.576
## P_Quant_R  100 -6.008e-17   9.626 -1.0052 -31.45 24.02 -5.440
## C_Verbal_R 100  2.696e-16   9.927  1.2883 -24.66 25.50 -7.084
## C_Quant_R  100 -1.776e-16   9.279 -0.8129 -26.11 20.75 -6.727
##             75th     Skew Kurtosis
## P_Verbal_R 7.550 -0.04563   0.5699
## P_Quant_R  6.135 -0.23775   0.6170
## C_Verbal_R 6.623 -0.15956  -0.2856
## C_Quant_R  7.167 -0.09505  -0.2942
```

```
CV <- cov(Skills[, 9:12])
D2_1 <- mahalanobis(Skills[, 9:12], center = colMeans(Skills[, 9:12]),
    cov = CV)
D2_1 <- as.data.frame(D2_1)
ggplot(D2_1, aes(sample = D2_1)) + stat_qq_band(distribution = "chisq",
    dparams = list(df = 4)) + stat_qq_line(distribution = "chisq",
    dparams = list(df = 4)) + stat_qq(distribution = "qchisq", dparams = list(df = 4)) +
    scale_y_continuous(breaks = seq(0, 24, 2)) + scale_x_continuous(breaks = seq(0,
    16, 1)) + coord_cartesian(xlim = c(0, 16), ylim = c(0, 24)) +
    xlab(expression("Expected Values from" * ~chi[4]^2)) + ylab(expression("Mahalanobis " *
    ~D^2)) + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
```

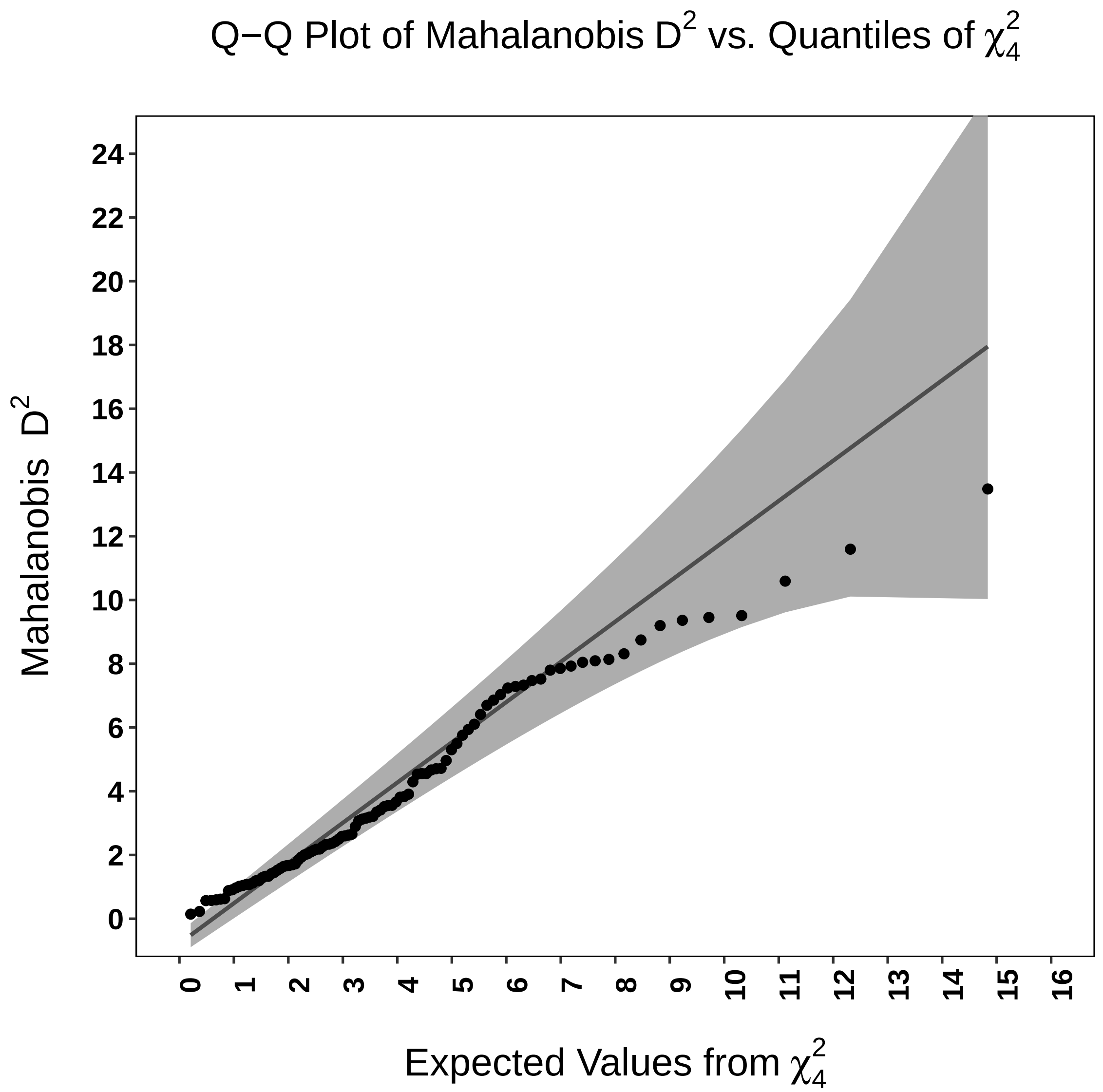
```
size = 12, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
plot.title = element_text(size = 16, face = "bold", margin = margin(0,
    0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
    linetype = 1, color = "black"), panel.grid.major = element_blank(),
panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
legend.title = element_blank()) + ggtitle(expression("Q-Q Plot of Mahalanobis" *
~D^2 * " vs. Quantiles of" * ~chi[4]^2))
```



2.3 Outlier Excluded

```
Skills$D2_1 <- D2_1
Skills_Trimmed <- Skills[which(Skills$D2_1 != max(Skills$D2_1)), ]

mvn(Skills_Trimmed[, 9:12], mvnTest = "mardia")

## $multivariateNormality
##              Test        Statistic           p value Result
## 1 Mardia Skewness  20.2783280472259 0.440644455966184    YES
## 2 Mardia Kurtosis 0.600058653103908 0.548467146873458    YES
## 3             MVN              <NA>              <NA>    YES
##
## $univariateNormality
##           Test   Variable Statistic   p value Normality
## 1 Shapiro-Wilk P_Verbal_R    0.9857    0.3630    YES
## 2 Shapiro-Wilk P_Quant_R     0.9889    0.5820    YES
## 3 Shapiro-Wilk C_Verbal_R    0.9872    0.4598    YES
## 4 Shapiro-Wilk C_Quant_R     0.9862    0.3923    YES
##
## $Descriptives
##             n   Mean Std.Dev  Median    Min   Max   25th  75th
## P_Verbal_R 99 0.3490   9.684 -1.2790 -21.66 26.09 -5.405 7.550
## P_Quant_R  99 0.3177   9.133 -0.7615 -23.53 24.02 -5.114 6.200
## C_Verbal_R 99 0.1044   9.922  1.3085 -24.66 25.50 -7.033 6.710
## C_Quant_R  99 0.2638   8.941 -0.4738 -24.09 20.75 -6.692 7.215
##                Skew Kurtosis
## P_Verbal_R  0.29712 -0.16599
## P_Quant_R   0.02811  0.07662
## C_Verbal_R -0.18123 -0.25780
## C_Quant_R   0.05667 -0.57682
```

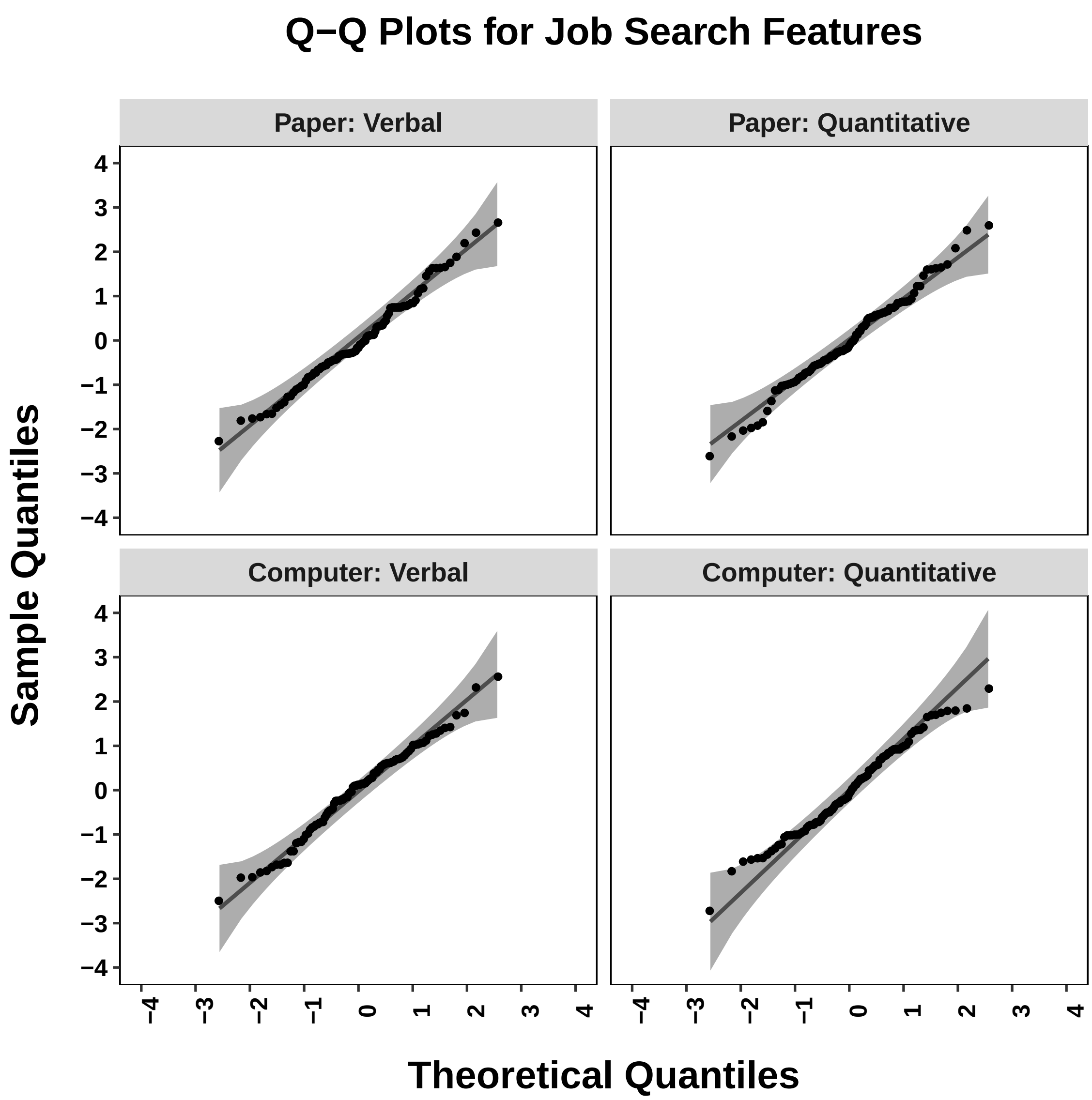
```
CV <- cov(Skills_Trimmed[, 9:12])
D2_1 <- mahalanobis(Skills_Trimmed[, 9:12], center = colMeans(Skills_Trimmed[,
    9:12]), cov = CV)
D2_1 <- as.data.frame(D2_1)
ggplot(D2_1, aes(sample = D2_1)) + stat_qq_band(distribution = "chisq",
    dparams = list(df = 4)) + stat_qq_line(distribution = "chisq",
    dparams = list(df = 4)) + stat_qq(distribution = "qchisq", dparams = list(df = 4)) +
    scale_y_continuous(breaks = seq(0, 24, 2)) + scale_x_continuous(breaks = seq(0,
    16, 1)) + coord_cartesian(xlim = c(0, 16), ylim = c(0, 24)) +
    xlab(expression("Expected Values from" * ~chi[4]^2)) + ylab(expression("Mahalanobis " *
    ~D^2)) + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle(expression("Q-Q Plot of Mahalanobis" *
    ~D^2 * " vs. Quantiles of" * ~chi[4]^2))
```



2.4 Univariate Normality

```
Skills_Trimmed_QQ <- scale(Skills_Trimmed[, 9:12])
Data_long <- melt(Skills_Trimmed_QQ)
Data_long <- as.data.frame(Data_long)
names(Data_long) <- c("Index", "feature", "value")
Data_long$feature_F <- factor(Data_long$feature, levels = c("P_Verbal_R",
    "P_Quant_R", "C_Verbal_R", "C_Quant_R"), labels = c("Paper: Verbal",
    "Paper: Quantitative", "Computer: Verbal", "Computer: Quantitative"))
p <- ggplot(Data_long, aes(sample = value)) + stat_qq_band() + stat_qq_line() +
    stat_qq(distribution = qnorm, size = 1) + scale_y_continuous(breaks = seq(-4,
    4, 1)) + scale_x_continuous(breaks = seq(-4, 4, 1)) + coord_cartesian(xlim = c(-4,
    4), ylim = c(-4, 4)) + xlab("Theoretical Quantiles") + ylab("Sample Quantiles") +
```

```
    theme(text = element_text(size = 14, family = "sans", color = "black",
        face = "bold"), axis.text.y = element_text(colour = "black",
        size = 10, face = "bold"), axis.text.x = element_text(colour = "black",
        size = 10, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
        0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
        15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
        plot.title = element_text(size = 16, face = "bold", margin = margin(0,
            0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
            linetype = 1, color = "black"), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
        plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
        legend.title = element_blank()) + ggtitle("Q-Q Plots for Job Search Features")
p + facet_wrap(~feature_F)
```



2.5 Homogeneity Assumption

We assume in discriminant analysis that the separate group variance-covariance matrices are homogeneous. Box's test can be used to test this assumption. Note, however, that it is also sensitive to violations of multivariate normality.

```
boxM(Skills[, 2:5], Skills$Group)

##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  Skills[, 2:5]
## Chi-Sq (approx.) = 93, df = 30, p-value = 2e-08

boxM(Skills[, 2:5], Skills$Group)$cov

## $`1`
##          P_Verbal P_Quant C_Verbal C_Quant
## P_Verbal   112.10   94.54    67.22   46.25
## P_Quant     94.54   99.71    82.84   70.41
## C_Verbal    67.22   82.84   117.58  104.82
## C_Quant     46.25   70.41   104.82  114.46
##
## $`2`
##          P_Verbal P_Quant C_Verbal C_Quant
## P_Verbal   164.89  121.83    90.29  121.84
## P_Quant    121.83  118.23    52.23   89.24
## C_Verbal    90.29   52.23   105.61   95.50
## C_Quant    121.84   89.24    95.50  120.37
##
## $`3`
##          P_Verbal P_Quant C_Verbal C_Quant
## P_Verbal   122.97   77.14    60.02   34.16
## P_Quant     77.14   87.49    76.05   46.13
## C_Verbal    60.02   76.05   106.12   72.31
## C_Quant     34.16   46.13    72.31   81.09
##
## $`4`
##          P_Verbal P_Quant C_Verbal C_Quant
## P_Verbal    33.24   37.85    33.51   22.40
## P_Quant     37.85   76.80    68.76   37.15
## C_Verbal    33.51   68.76    77.15   42.04
## C_Quant     22.40   37.15    42.04   39.21

boxM(Skills[, 2:5], Skills$Group)$pooled

##          P_Verbal P_Quant C_Verbal C_Quant
## P_Verbal   108.30   82.84    62.76   56.16
## P_Quant     82.84   95.56    69.97   60.74
## C_Verbal    62.76   69.97   101.62   78.67
## C_Quant     56.16   60.74    78.67   88.79

boxM(Skills_Trimmed[, 2:5], Skills_Trimmed$Group)

##
##  Box's M-test for Homogeneity of Covariance Matrices
```

```
##
## data:  Skills_Trimmed[, 2:5]
## Chi-Sq (approx.) = 81, df = 30, p-value = 0.000002
```

```
boxM(Skills_Trimmed[, 2:5], Skills_Trimmed$Group)$cov
```

```
## $`1`
##          P_Verbal P_Quant C_Verbal C_Quant
## P_Verbal   112.10   94.54    67.22   46.25
## P_Quant     94.54   99.71    82.84   70.41
## C_Verbal    67.22   82.84   117.58  104.82
## C_Quant     46.25   70.41   104.82  114.46
##
## $`2`
##          P_Verbal P_Quant C_Verbal C_Quant
## P_Verbal   117.98   77.91    78.04   86.27
## P_Quant     77.91   78.56    39.78   55.92
## C_Verbal    78.04   39.78   105.37   87.43
## C_Quant     86.27   55.92    87.43   94.73
##
## $`3`
##          P_Verbal P_Quant C_Verbal C_Quant
## P_Verbal   122.97   77.14    60.02   34.16
## P_Quant     77.14   87.49    76.05   46.13
## C_Verbal    60.02   76.05   106.12   72.31
## C_Quant     34.16   46.13    72.31   81.09
##
## $`4`
##          P_Verbal P_Quant C_Verbal C_Quant
## P_Verbal    33.24   37.85    33.51   22.40
## P_Quant     37.85   76.80    68.76   37.15
## C_Verbal    33.51   68.76    77.15   42.04
## C_Quant     22.40   37.15    42.04   39.21
```

```
boxM(Skills_Trimmed[, 2:5], Skills_Trimmed$Group)$pooled
```

```
##          P_Verbal P_Quant C_Verbal C_Quant
## P_Verbal    96.35   71.80    59.51   46.86
## P_Quant     71.80   85.71    67.14   52.37
## C_Verbal    59.51   67.14   101.52   76.54
## C_Quant     46.86   52.37    76.54   82.24
```

3 Original MANOVA

Here is the original MANOVA that may be suspect because of the violations of homogeneity and multivariate normality.

```
# This function takes as input the data frame used for a
# discriminant analysis along with the object into which the
# discriminant analysis results are saved. The candisc( ) function
# is assumed to be used for the discriminant analysis. The
# function return a chi-square test of the hypothesis that the
# current discriminant function and all subsequent discriminant
```

```
# functions provide no significant group separation. The test
# parallels the F ratio version reported by candisc( ) function.
DA_Chi_Square <- function(data_frame, candisc_object) {
    n <- length(data_frame[, 1])
    q <- length(candisc_object$coeffs.std[, 1])
    g <- candisc_object$dfh + 1
    W <- Wilks(candisc_object)
    results <- matrix(NA, nrow = candisc_object$ndim, ncol = 3)
    for (i in seq(1, candisc_object$ndim, 1)) {
        k <- i - 1
        chi_test <- -(n - (q + g)/2 - 1) * log(W$`LR test stat`[i])
        chi_df <- (q - k) * (g - k - 1)
        chi_p <- pchisq(chi_test, chi_df, lower.tail = FALSE)
        results[i, ] <- c(chi_test, chi_df, chi_p)
    }
    results <- as.data.frame(results)
    names(results) <- c("Chi_Sq", "df", "p")
    return(results)
}
```

```
LM_1 <- lm(cbind(P_Verbal, P_Quant, C_Verbal, C_Quant) ~ as.factor(Group),
    data = Skills)
LDA_1 <- candisc(LM_1, data = Skills)
LDA_1

##
## Canonical Discriminant Analysis for as.factor(Group):
##
##   CanRsq Eigenvalue Difference Percent Cumulative
## 1  0.863      6.313       1.11    52.3       52.3
## 2  0.839      5.199       1.11    43.0       95.3
## 3  0.362      0.567       1.11     4.7      100.0
##
## Test of H0: The canonical correlations in the
## current row and all that follow are zero
##
##   LR test stat approx F numDF denDF Pr(> F)
## 1        0.014     82.3    12   246  <2e-16
## 2        0.103     66.3     6   188  <2e-16
## 3        0.638              2

summary(LDA_1)

##
## Canonical Discriminant Analysis for as.factor(Group):
##
##   CanRsq Eigenvalue Difference Percent Cumulative
## 1 0.8633     6.3128      1.114  52.263      52.26
## 2 0.8387     5.1989      1.114  43.042      95.30
## 3 0.3619     0.5671      1.114   4.695     100.00
##
## Class means:
##
##      Can1    Can2     Can3
```

```
## 1 -0.4635  1.1546 -1.21186
## 2  1.8729  2.8081  0.67677
## 3 -3.8292 -0.7785  0.49992
## 4  2.4198 -3.1842  0.03517
##
##  std coefficients:
##             Can1    Can2    Can3
## P_Verbal  0.8044 -0.7131 -1.3283
## P_Quant   0.5891  1.2191  1.4285
## C_Verbal -0.4769 -0.1206 -0.5982
## C_Quant  -0.1877 -1.0700  0.8519
```

```
LDA_1$coeffs.raw
```

```
##              Can1     Can2     Can3
## P_Verbal  0.07729 -0.06852 -0.12764
## P_Quant   0.06026  0.12471  0.14614
## C_Verbal -0.04731 -0.01197 -0.05934
## C_Quant  -0.01992 -0.11356  0.09041
```

```
LDA_1$coeffs.std
```

```
##             Can1    Can2    Can3
## P_Verbal  0.8044 -0.7131 -1.3283
## P_Quant   0.5891  1.2191  1.4285
## C_Verbal -0.4769 -0.1206 -0.5982
## C_Quant  -0.1877 -1.0700  0.8519
```

```
LDA_1$structure
```

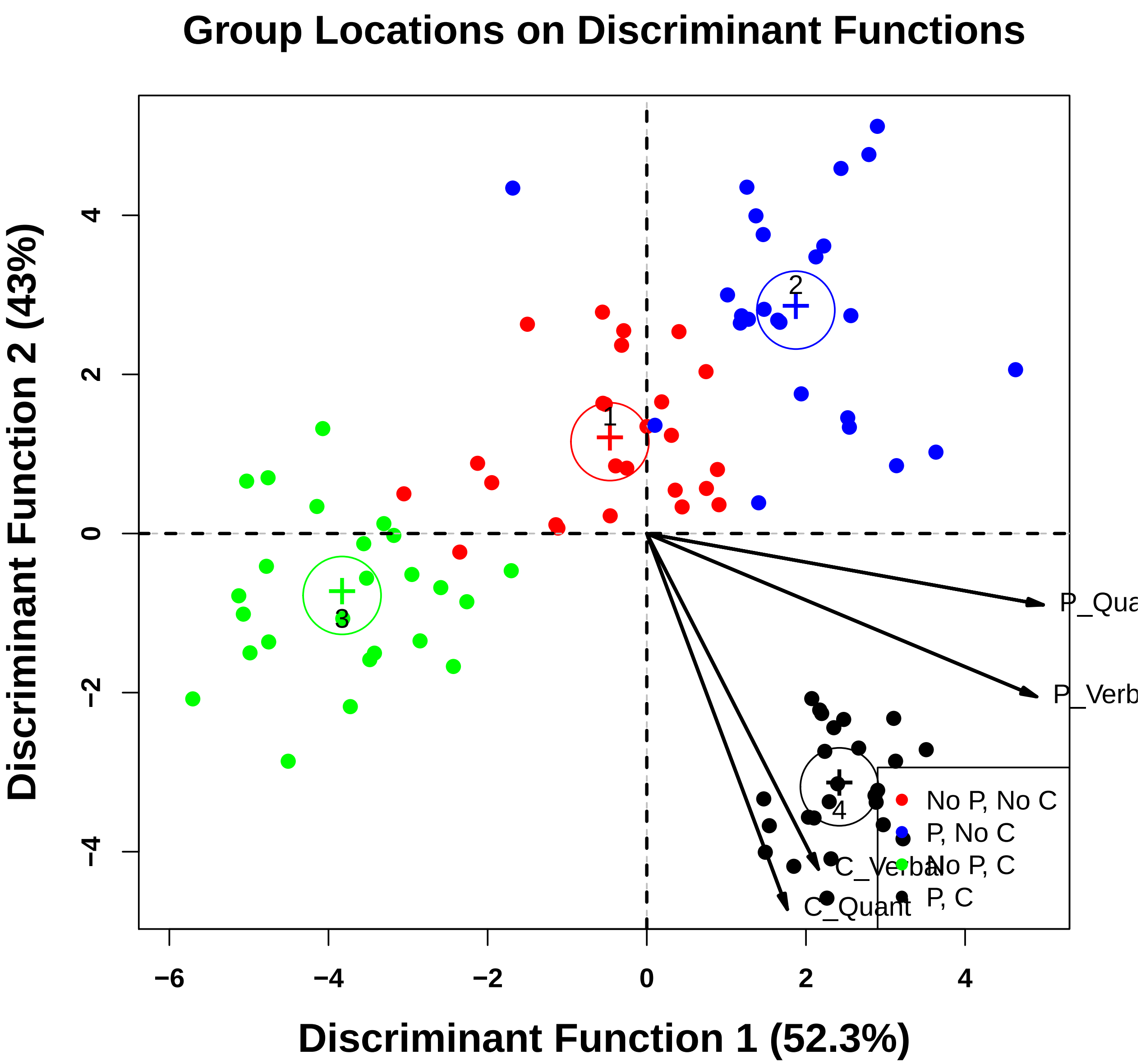
```
##            Can1    Can2     Can3
## P_Verbal 0.9196 -0.3851 -0.01672
## P_Quant  0.9350 -0.1684  0.26787
## C_Verbal 0.4047 -0.7919  0.22928
## C_Quant  0.3313 -0.8866  0.30943
```

```
DA_Chi_Square(Skills, LDA_1)
```

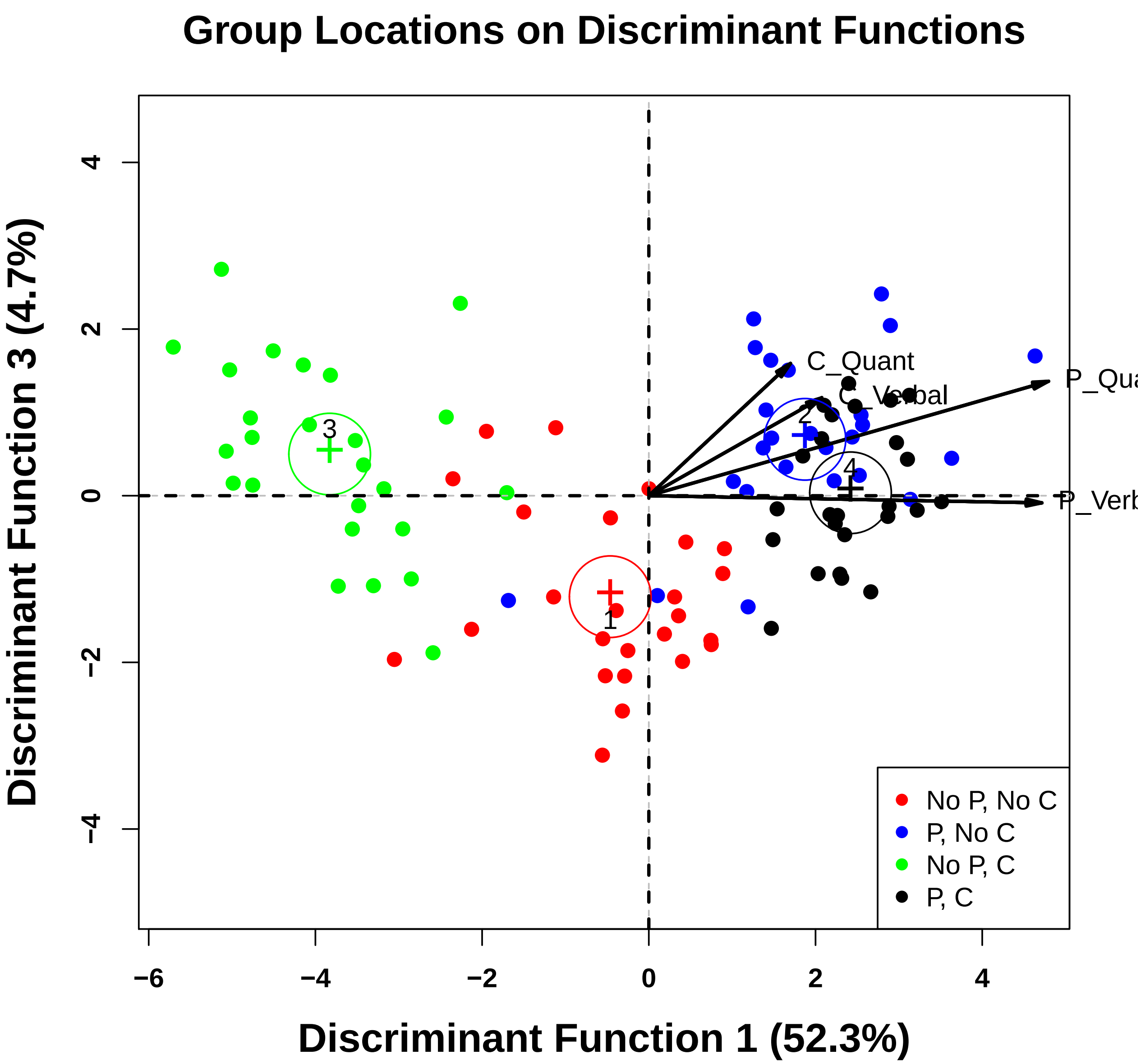
```
##   Chi_Sq df         p
## 1 405.01 12 3.292e-79
## 2 215.99  6 7.437e-44
## 3  42.68  2 5.402e-10
```

```
# Wilks' Lambda
Actual_Wilks_1 <- (1/(1 + LDA_1$eigenvalues[1])) * (1/(1 + LDA_1$eigenvalues[2])) *
    (1/(1 + LDA_1$eigenvalues[3]))
Actual_Wilks_2 <- (1/(1 + LDA_1$eigenvalues[2])) * (1/(1 + LDA_1$eigenvalues[3]))
Actual_Wilks_3 <- (1/(1 + LDA_1$eigenvalues[3]))

# Hotelling-Lawley Trace
Actual_HL_1 <- LDA_1$eigenvalues[1] + LDA_1$eigenvalues[2] + LDA_1$eigenvalues[3]
Actual_HL_2 <- LDA_1$eigenvalues[2] + LDA_1$eigenvalues[3]
Actual_HL_3 <- LDA_1$eigenvalues[3]

# Piallai's Trace
Actual_Pillai_1 <- (LDA_1$eigenvalues[1]/(1 + LDA_1$eigenvalues[1])) +
```

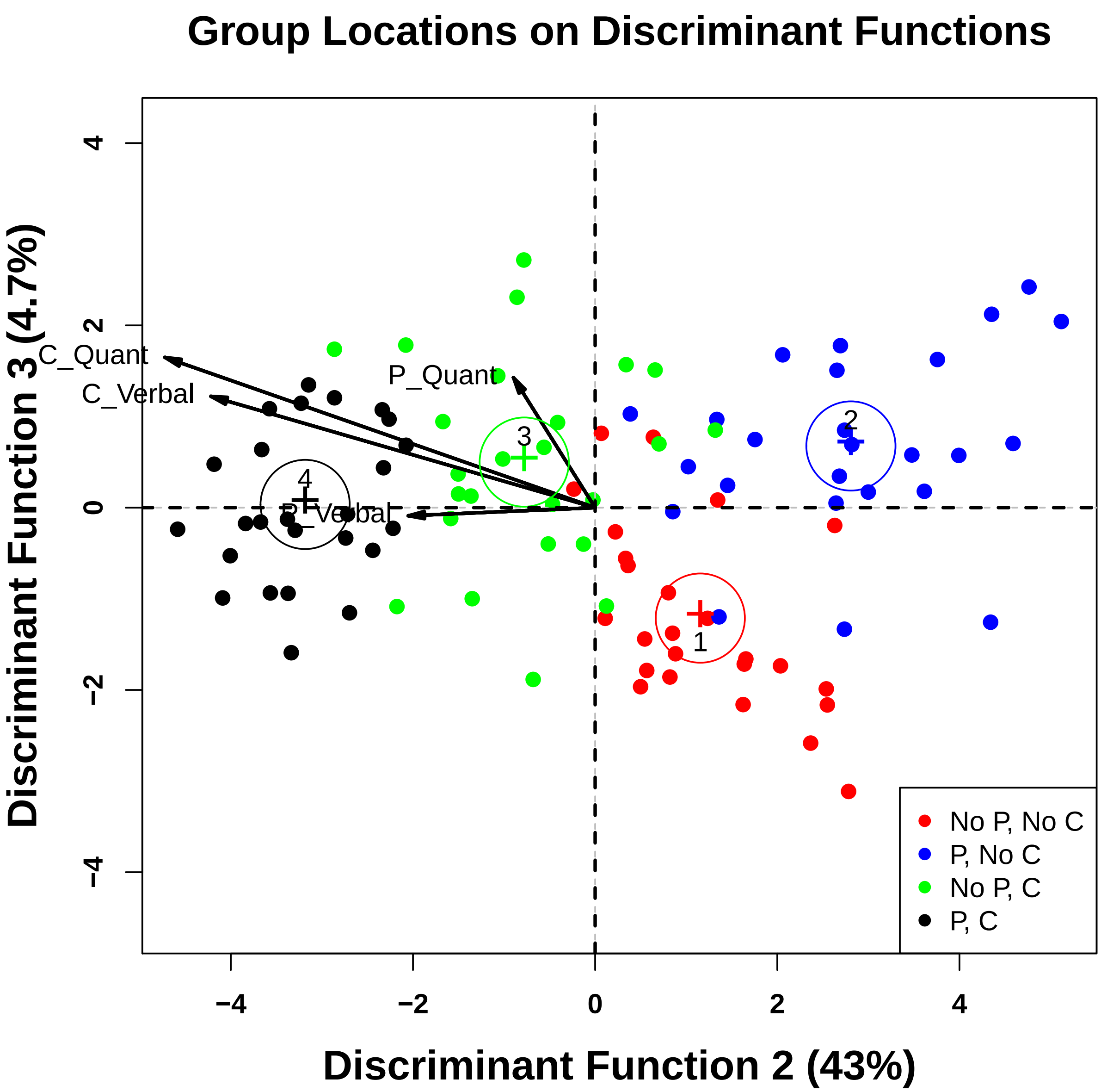
```r
    (LDA_1$eigenvalues[2]/(1 + LDA_1$eigenvalues[2])) + (LDA_1$eigenvalues[3]/(1 +
    LDA_1$eigenvalues[3]))
Actual_Pillai_2 <- (LDA_1$eigenvalues[2]/(1 + LDA_1$eigenvalues[2])) +
    (LDA_1$eigenvalues[3]/(1 + LDA_1$eigenvalues[3]))
Actual_Pillai_3 <- (LDA_1$eigenvalues[3]/(1 + LDA_1$eigenvalues[3]))

Actual_Wilks_1

## [1] 0.01408

Actual_Wilks_2

## [1] 0.1029

Actual_Wilks_3

## [1] 0.6381

Actual_HL_1

## [1] 12.08

Actual_HL_2

## [1] 5.766

Actual_HL_3

## [1] 0.5671

Actual_Pillai_1

## [1] 2.064

Actual_Pillai_2

## [1] 1.201

Actual_Pillai_3

## [1] 0.3619

plot(LDA_1, main = list("Group Locations on Discriminant Functions",
    cex = 1.5), cex = 1.25, font.axis = 2, col = c("red", "blue",
    "green", "black"), pch = c(16, 16, 16, 16), font.lab = 2, cex.lab = 1.5,
    prefix = "Discriminant Function ", var.col = "black", var.lwd = 2,
    which = c(1, 2))

## Vector scale factor set to 5.327

abline(v = 0, lty = 2, lwd = 2, col = "black")
abline(h = 0, lty = 2, lwd = 2, col = "black")
legend("bottomright", c("No P, No C", "P, No C", "No P, C", "P, C"),
    col = c("red", "blue", "green", "black"), pch = 16)
```




```r
plot(LDA_1, main = list("Group Locations on Discriminant Functions",
    cex = 1.5), cex = 1.25, font.axis = 2, col = c("red", "blue",
    "green", "black"), pch = c(16, 16, 16, 16), font.lab = 2, cex.lab = 1.5,
    prefix = "Discriminant Function ", var.col = "black", var.lwd = 2,
    which = c(1, 3))

## Vector scale factor set to 5.129

abline(v = 0, lty = 2, lwd = 2, col = "black")
abline(h = 0, lty = 2, lwd = 2, col = "black")
legend("bottomright", c("No P, No C", "P, No C", "No P, C", "P, C"),
    col = c("red", "blue", "green", "black"), pch = 16)
```




```r
plot(LDA_1, main = list("Group Locations on Discriminant Functions",
    cex = 1.5), cex = 1.25, font.axis = 2, col = c("red", "blue",
    "green", "black"), pch = c(16, 16, 16, 16), font.lab = 2, cex.lab = 1.5,
    prefix = "Discriminant Function ", var.col = "black", var.lwd = 2,
    which = c(2, 3))

## Vector scale factor set to 5.327

abline(v = 0, lty = 2, lwd = 2, col = "black")
abline(h = 0, lty = 2, lwd = 2, col = "black")
legend("bottomright", c("No P, No C", "P, No C", "No P, C", "P, C"),
    col = c("red", "blue", "green", "black"), pch = 16)
```



4 Bootstrap Analysis

If the assumptions underlying the MANOVA (homogeneous covariance matrices, multivariate normality) are not viable, an alternative approach can be taken that does not make these assumptions: bootstrapping. In the bootstrapping approach, we assume that whatever population the sample came from, it is representative of that population. Therefore we can sample randomly from the sample, with replacement, to get repeated representative samples of the same size on which we can repeat the analyses. The resulting empirical sampling distributions of parameters can be used to make inferences. Note that we may want to randomly sample with replacement from the sample of cases, but leave intact each case's response profile, so that the covariance structure that underlies the responses is retained.

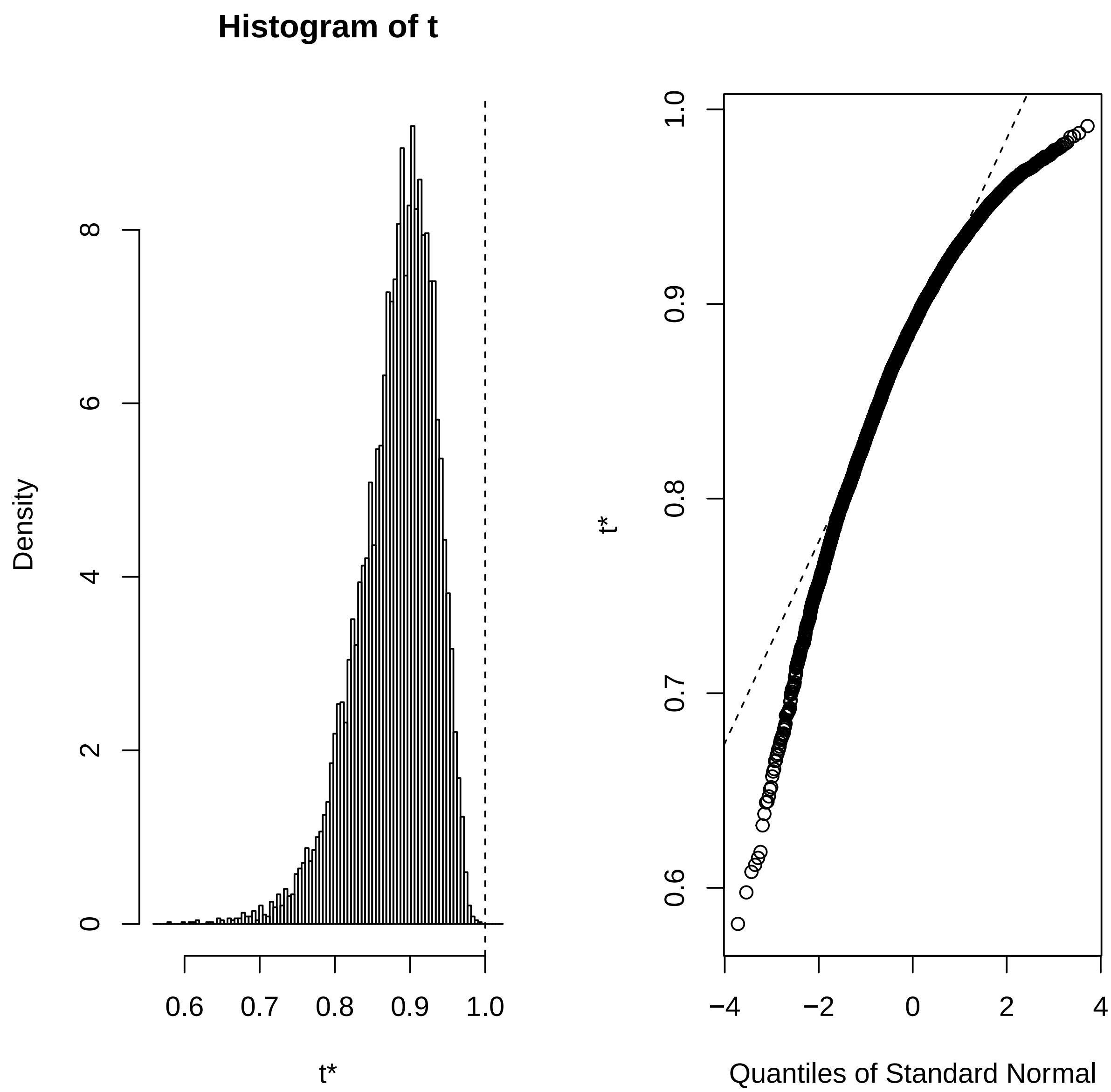
Because we are interested in a test of the null hypothesis that the discriminant functions cannot separate the groups, we will draw the bootstrap samples from the residualized data. This will produce null-consistent sampling distributions of Wilks, Hotelling-Lawley, and Pillai indices that we can then compare to the values from the original analysis. We will use the original data, including the outlier, in order explore the robustness of this approach.

Note that the only confidence interval method that will make sense here is the percentile method. Because we are using the residualized data, the adjustments made with the normative, basic, and bias-corrected and accelerated methods will not be correct; they rely on the original parameter estimates using the non-residualized data.

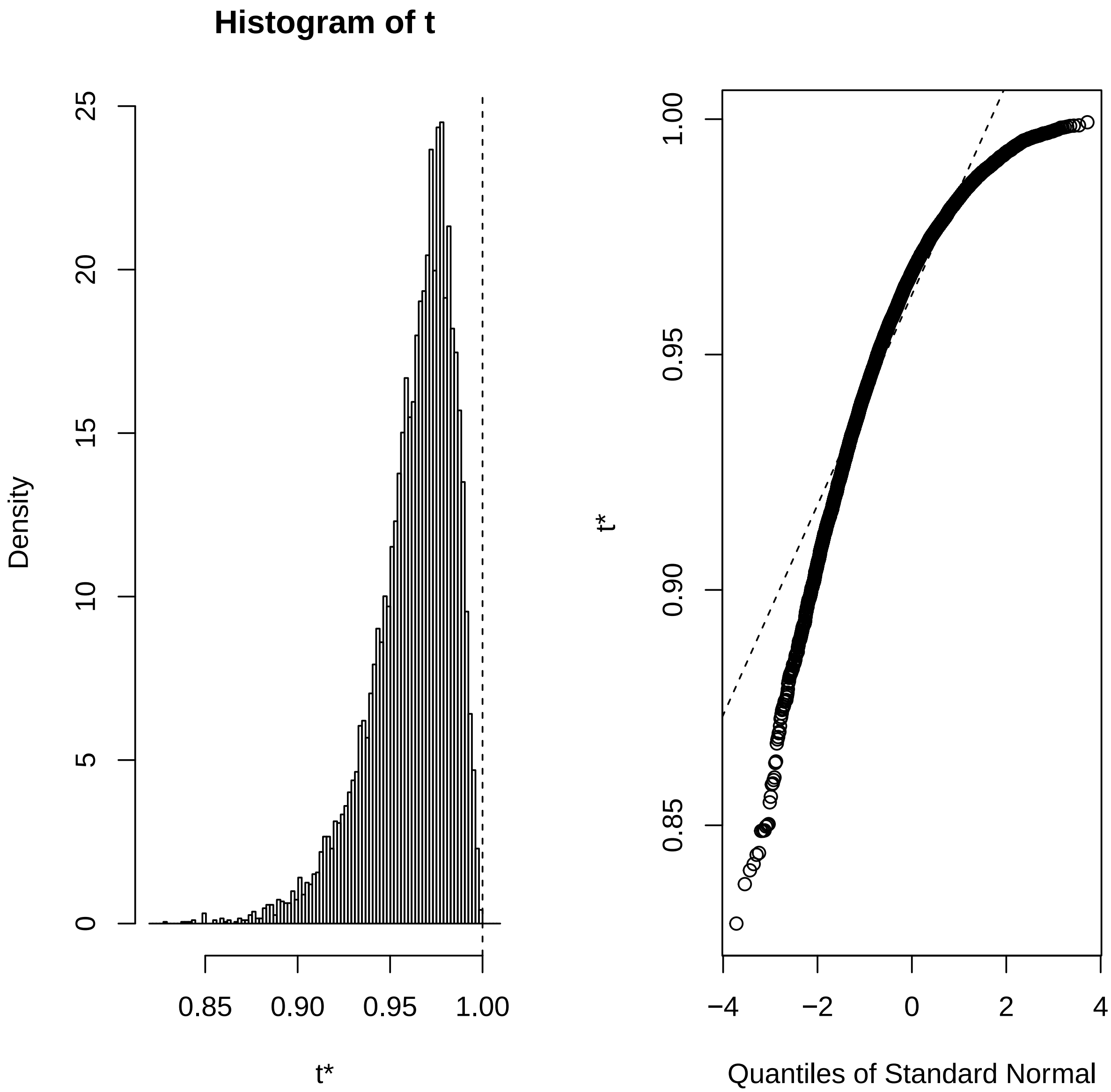
```
# This function will conduct the linear discriminant analysis
# using resamples of the data in order to get the MANOVA results.
# The formula that is passed is for the lm( ) function that the
# candisc( ) function uses in order to get discriminant function
# results. The functions uses the eigenvalues to calculate
# estimates of Wilks Lambda, Hotelling-Lawley trace, and Pillai
# trace, using the step-down method (test all three, exclude the
# largest and test the remaining two, test the last).
lda_boot <- function(formula, data, indices) {
    boot_data <- data[indices, ]
    boot_MLM <- lm(formula, data = boot_data)
    boot_fit <- candisc(boot_MLM, data = boot_data)
    Wilks_1 <- (1/(1 + boot_fit$eigenvalues[1])) * (1/(1 + boot_fit$eigenvalues[2])) *
        (1/(1 + boot_fit$eigenvalues[3]))
    Wilks_2 <- (1/(1 + boot_fit$eigenvalues[2])) * (1/(1 + boot_fit$eigenvalues[3]))
    Wilks_3 <- (1/(1 + boot_fit$eigenvalues[3]))
    HL_1 <- boot_fit$eigenvalues[1] + boot_fit$eigenvalues[2] + boot_fit$eigenvalues[3]
    HL_2 <- boot_fit$eigenvalues[2] + boot_fit$eigenvalues[3]
    HL_3 <- boot_fit$eigenvalues[3]
    Pillai_1 <- (boot_fit$eigenvalues[1]/(1 + boot_fit$eigenvalues[1])) +
        (boot_fit$eigenvalues[2]/(1 + boot_fit$eigenvalues[2])) +
        (boot_fit$eigenvalues[3]/(1 + boot_fit$eigenvalues[3]))
    Pillai_2 <- (boot_fit$eigenvalues[2]/(1 + boot_fit$eigenvalues[2])) +
        (boot_fit$eigenvalues[3]/(1 + boot_fit$eigenvalues[3]))
    Pillai_3 <- (boot_fit$eigenvalues[3]/(1 + boot_fit$eigenvalues[3]))
    results <- rbind(Wilks_1, Wilks_2, Wilks_3, HL_1, HL_2, HL_3,
        Pillai_1, Pillai_2, Pillai_3)
    return(results)
}
```

```
# Select the data from the original file that will be passed to
# the bootstrapping function.
boot_input <- Skills[, c(6, 9:12)]

# Call the boot( ) function from the boot library and request
# 10000 resamples.
boot_results <- boot(data = boot_input, statistic = lda_boot, R = 10000,
    formula = cbind(P_Verbal_R, P_Quant_R, C_Verbal_R, C_Quant_R) ~
        as.factor(Group))
```

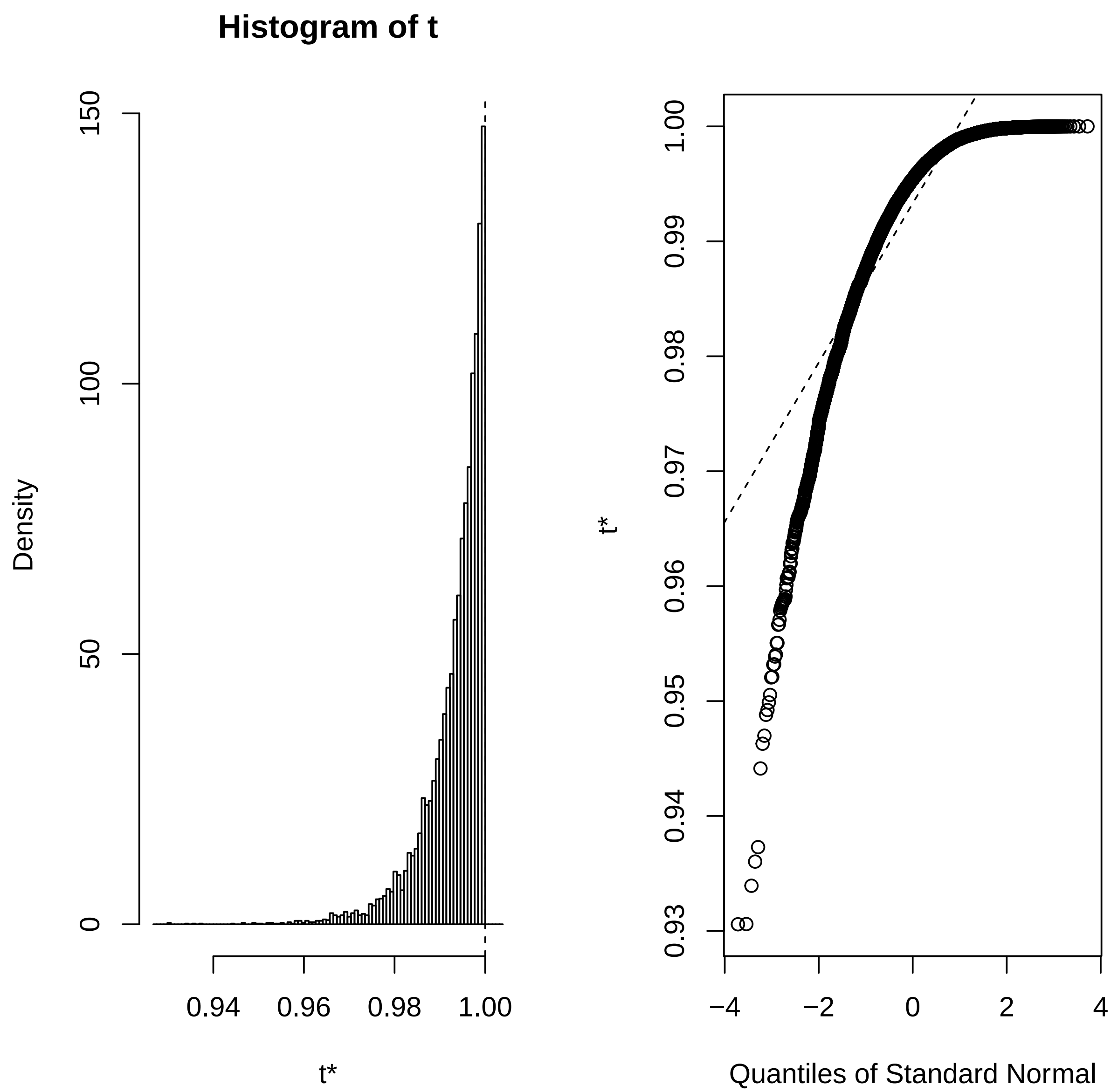
```r
# Plot the bootstrapping results using the default plot( )
# function.
plot(boot_results, index = 1)
```



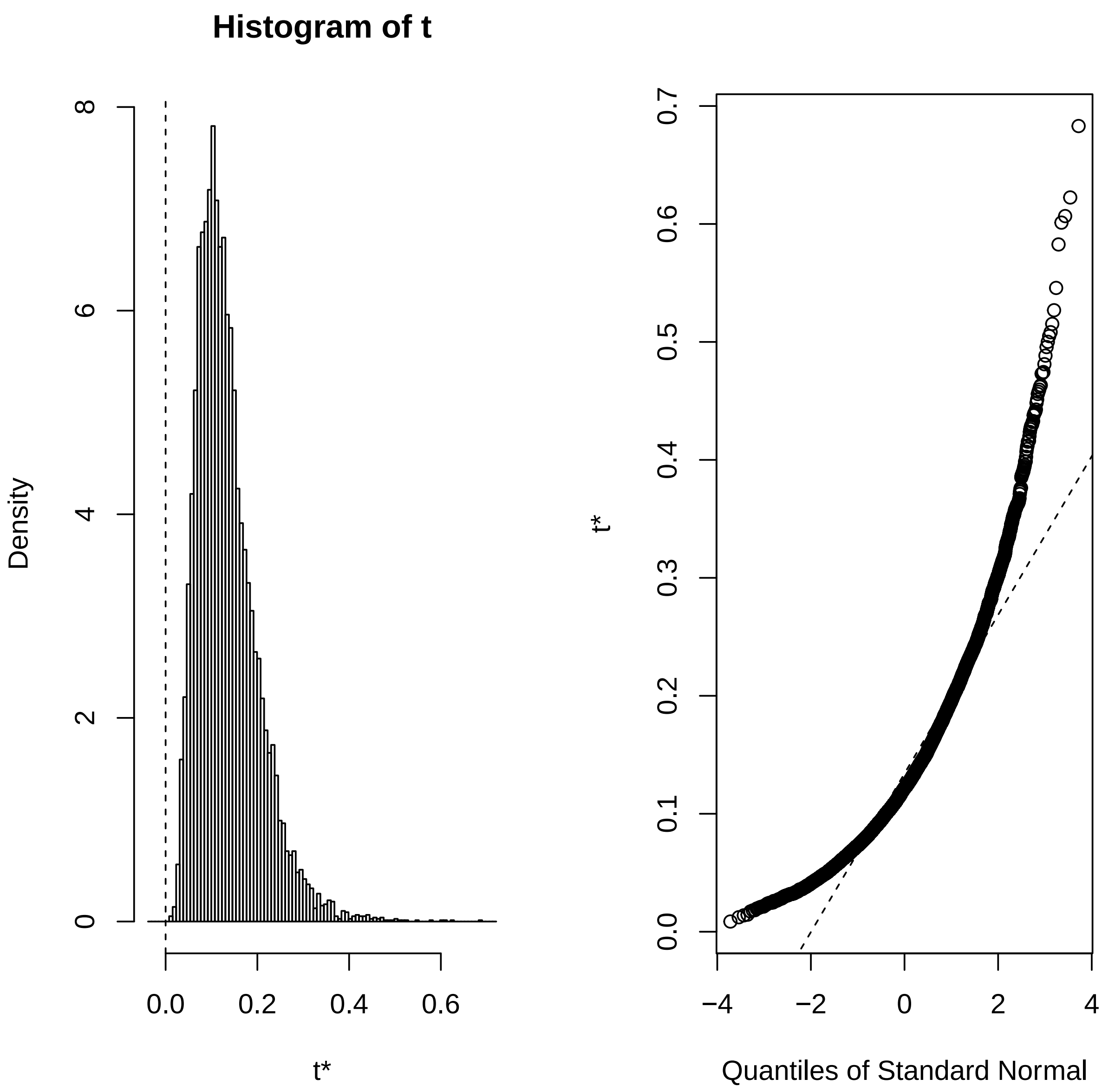

```r
plot(boot_results, index = 2)
```



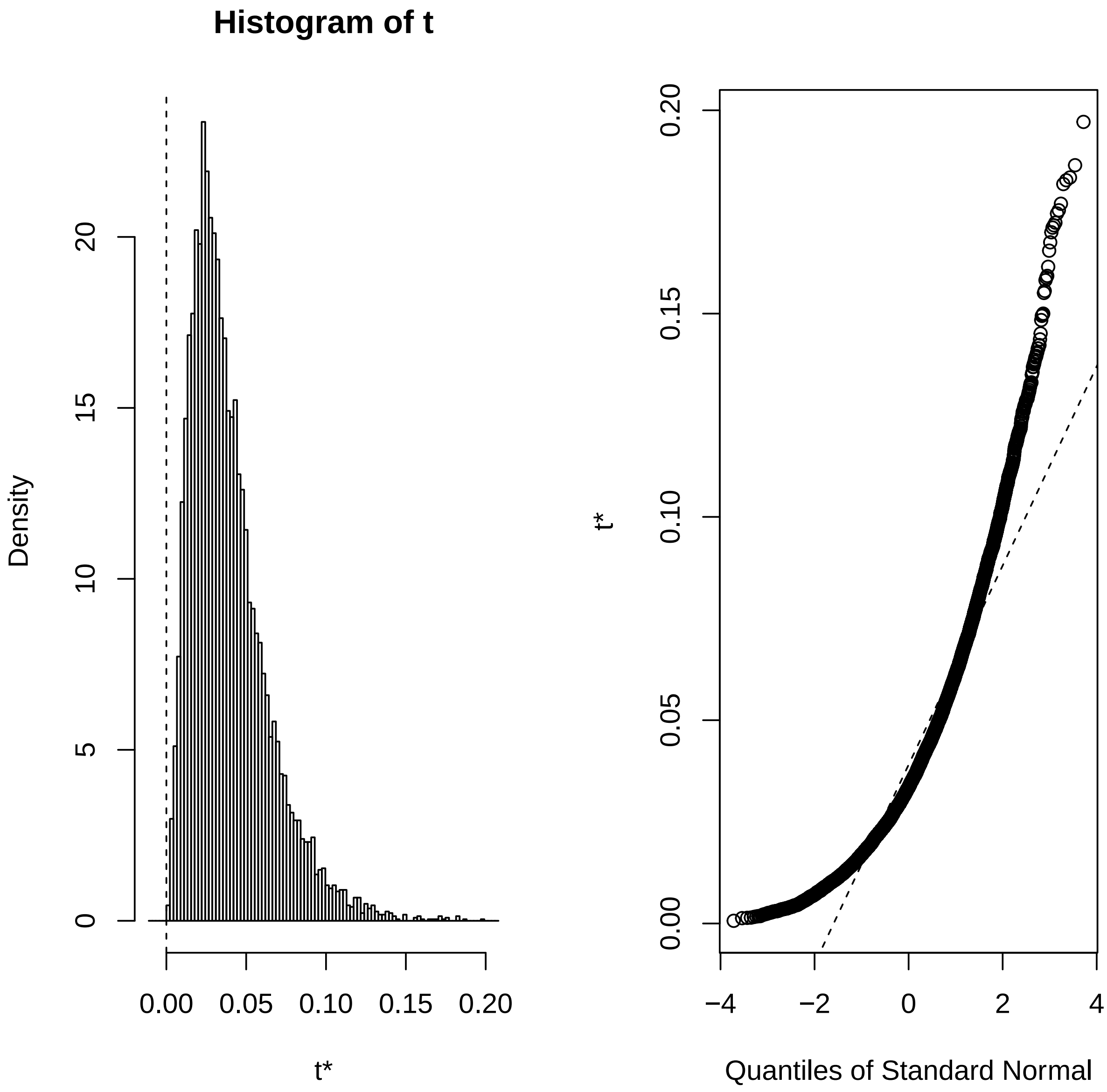

```
plot(boot_results, index = 3)
```



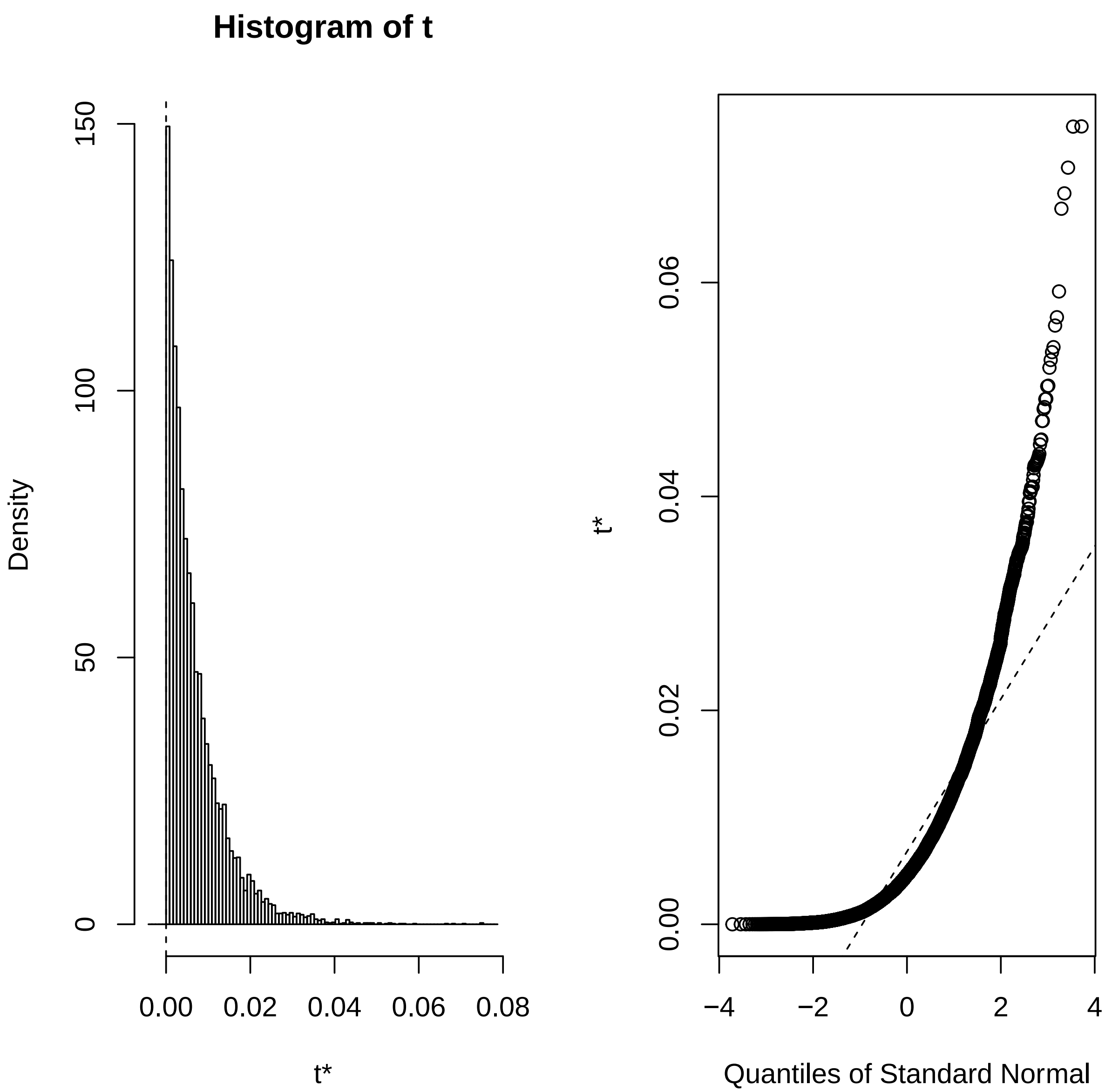

```r
plot(boot_results, index = 4)
```



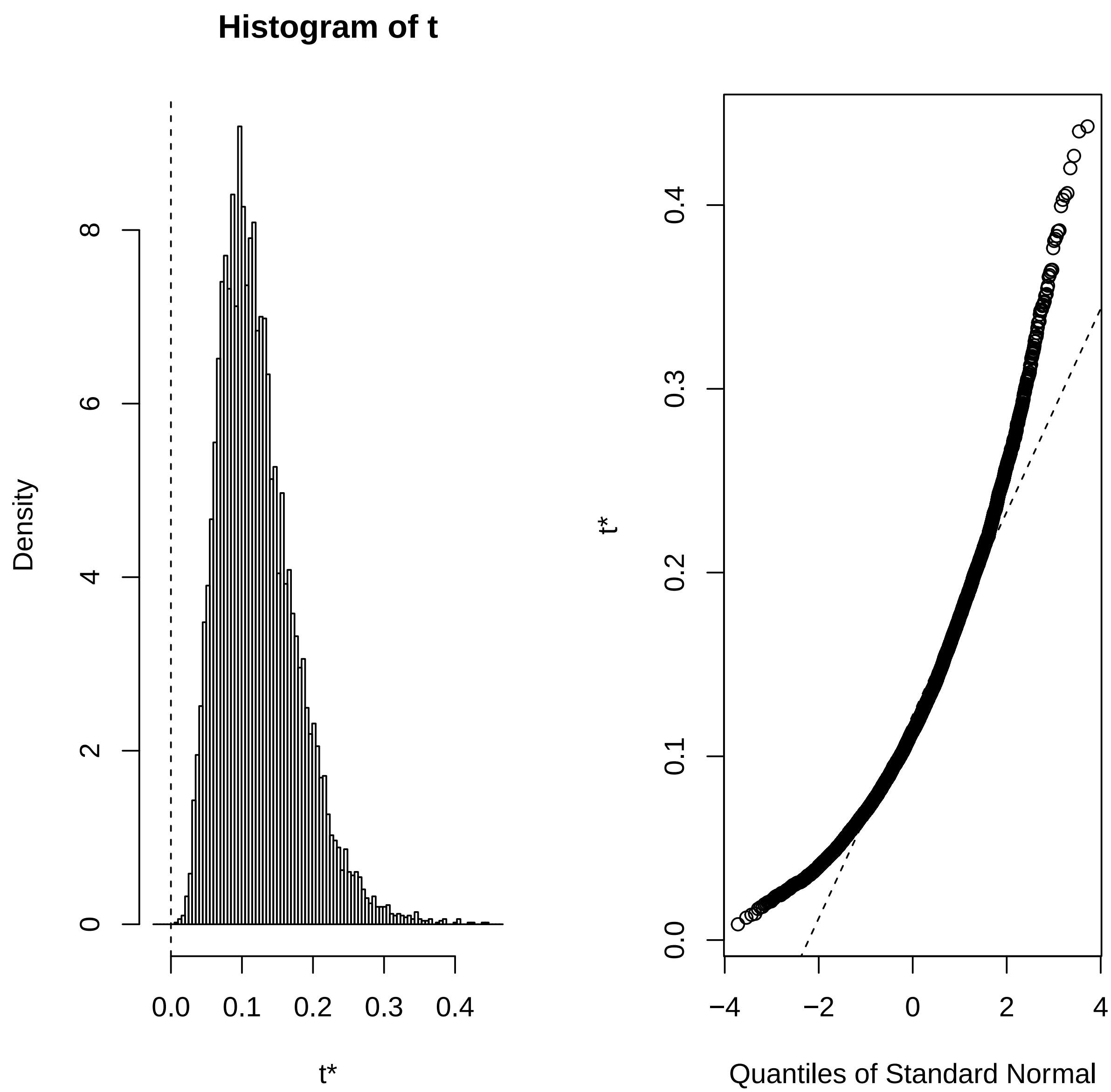

```r
plot(boot_results, index = 5)
```



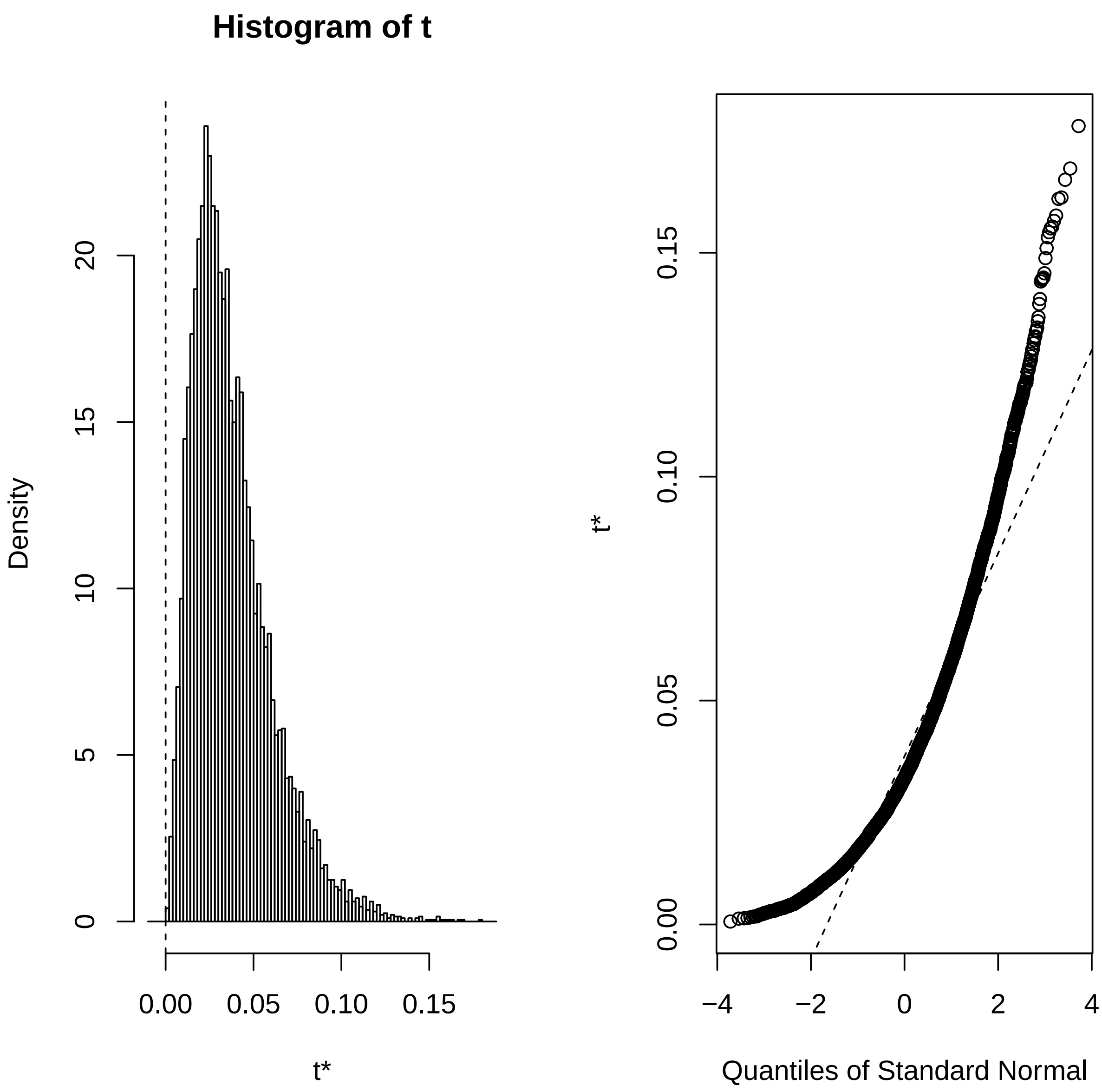

```
plot(boot_results, index = 6)
```



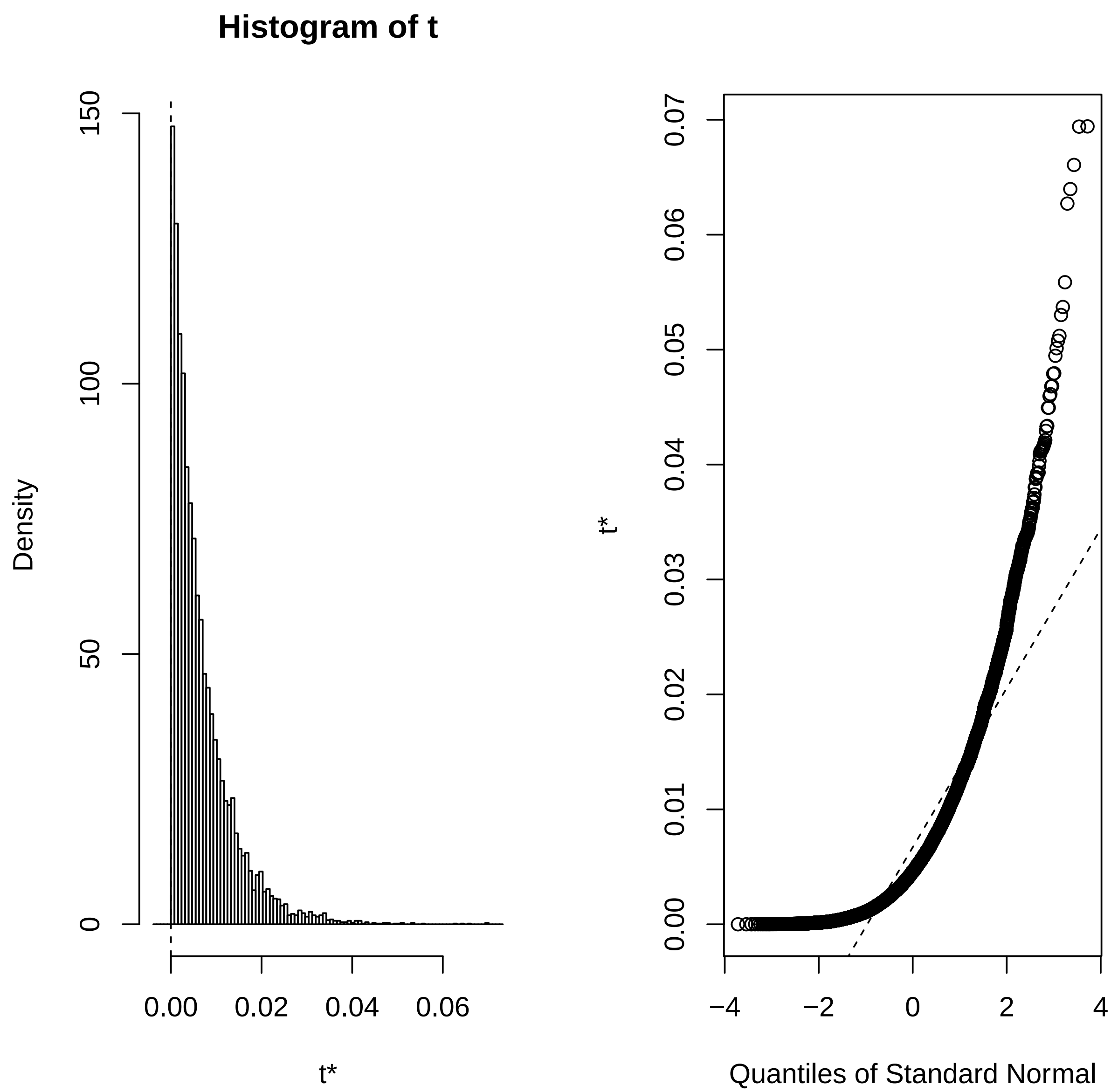

```
plot(boot_results, index = 7)
```




```r
plot(boot_results, index = 8)
```




```
plot(boot_results, index = 9)
```



4.1 Outlier Elimination Function

Occasionally the bootstrapping process will produce extreme outliers due to the vagaries of random sampling. Those outliers can create problems for histograms. This function identifies and eliminates outliers from the data frame to be used in a histogram. It tallies the number of eliminated outliers so that can be indicated on the histogram.

```
outlier_detect <- function(data, Z = 4) {
    data_2a <- matrix(NA, nrow = (length(data[1])))
    data_2b <- matrix(NA, nrow = (length(data[1])))
    data_3 <- scale(data)
    counter_a <- 0
    counter_b <- 0
    for (i in seq(1, length(data), 1)) {
```

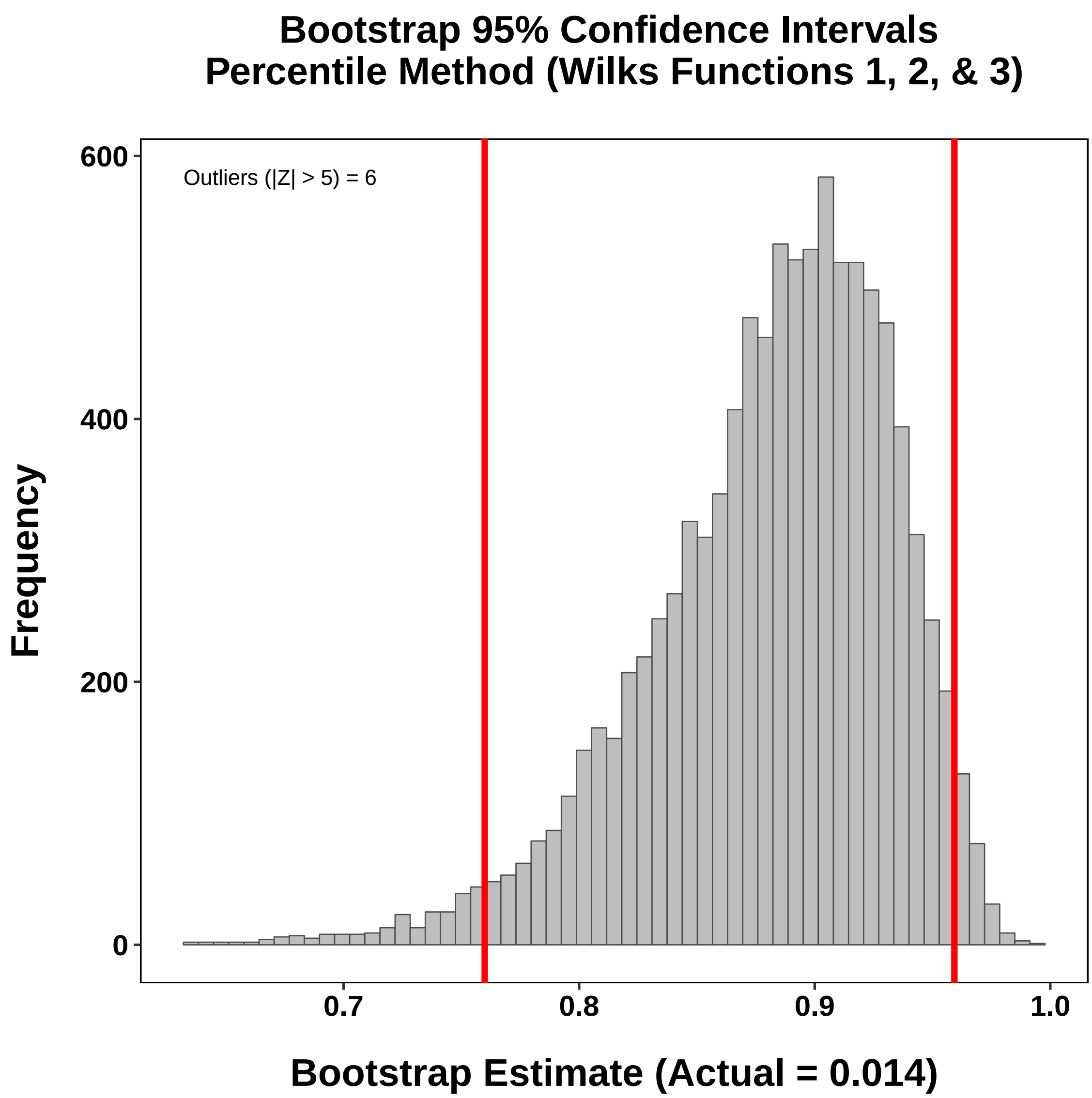
```
        if (abs(data_3[i]) <= Z) {
            counter_a <- counter_a + 1
            data_2a[counter_a] <- data[i]
        } else {
            counter_b <- counter_b + 1
            data_2b[counter_b] <- data[i]
        }
    }
    data_2a <- na.omit(data_2a)
    data_2b <- na.omit(data_2b)
    results <- list(data_2a, data_2b, counter_a, counter_b)
    return(results)
}
```

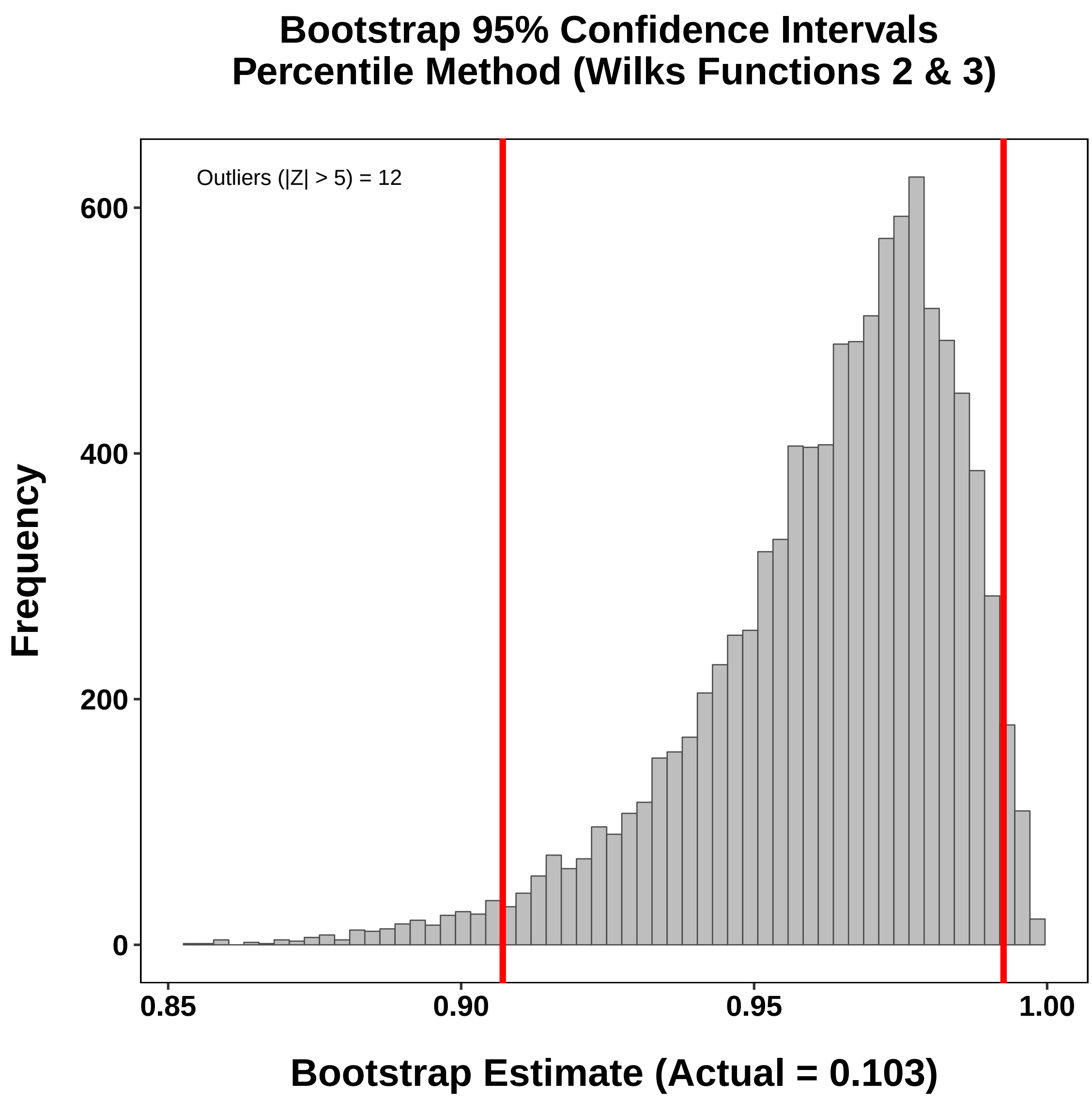
4.2 Simple Bootstrapping with Percentile Method Confidence Intervals

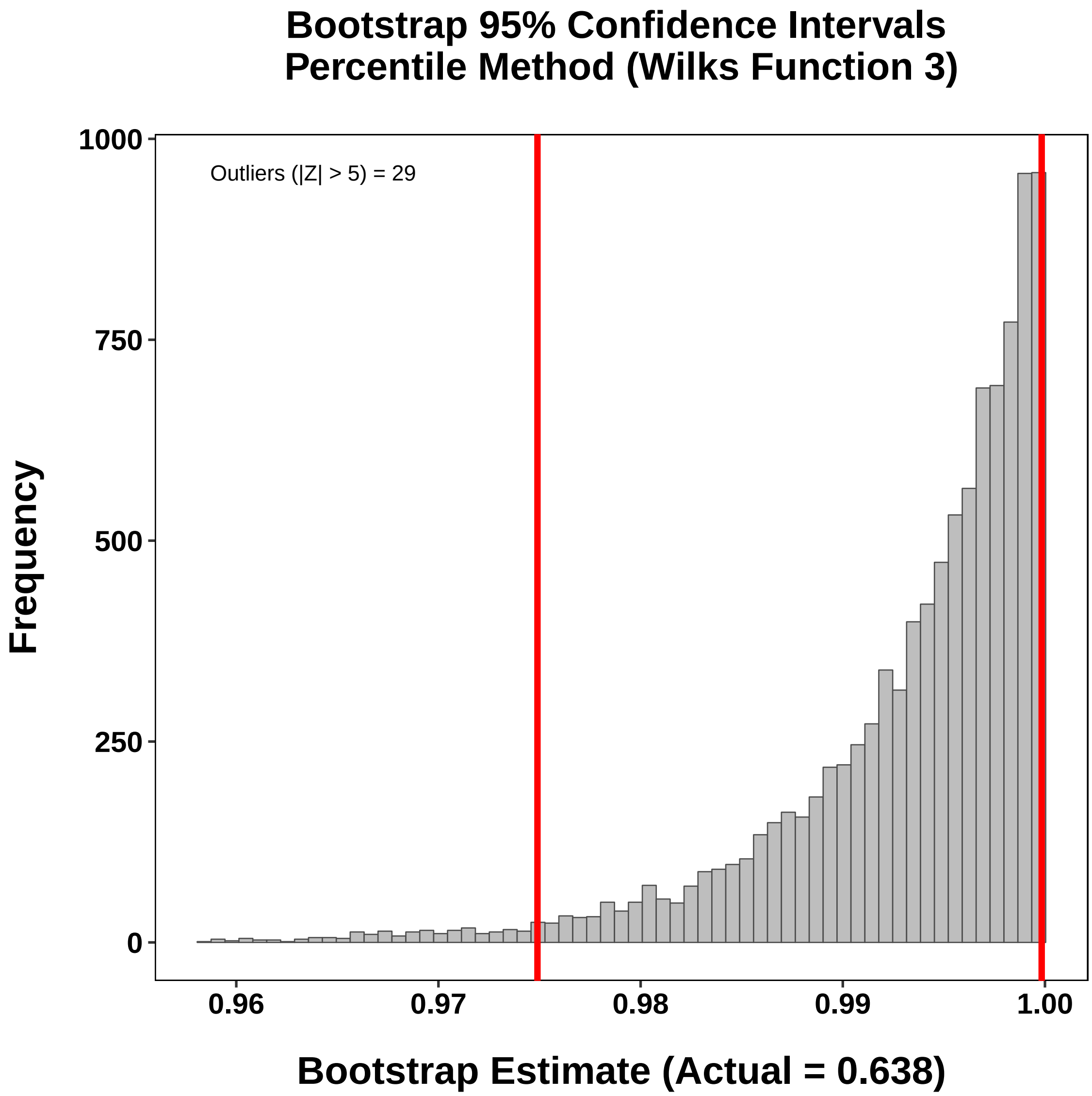
Of the several ways to calculate bootstrap confidence intervals, the bias corrected and accelerated is the most commonly recommended. Because we are using the residualized data, the adjustments made with the bias-corrected and accelerated method (as well as the normative and basic methods) will not be correct; they rely on the original parameter estimates using the non-residualized data. The percentile method is used.

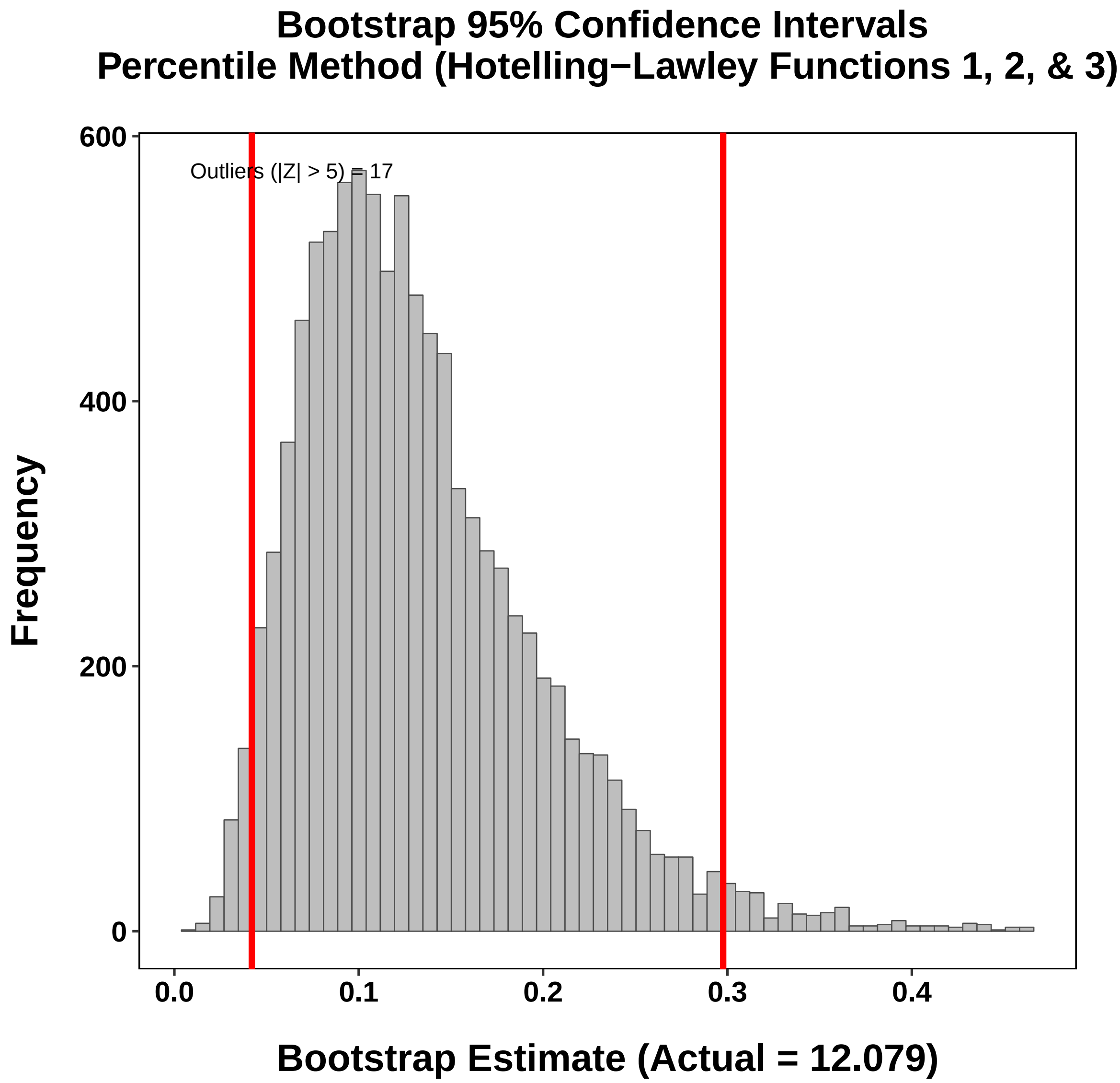
```
Effects <- c("Wilks Functions 1, 2, & 3", "Wilks Functions 2 & 3",
    "Wilks Function 3", "Hotelling-Lawley Functions 1, 2, & 3", "Hotelling-Lawley Functions 2 & 3",
    "Hotelling-Lawley Function 3", "Pillai Functions 1, 2, & 3", "Pillai Functions 2 & 3",
    "Pillai Function 3")

Original <- matrix(NA, nrow = 9)
Original[1] <- Actual_Wilks_1
Original[2] <- Actual_Wilks_2
Original[3] <- Actual_Wilks_3
Original[4] <- Actual_HL_1
Original[5] <- Actual_HL_2
Original[6] <- Actual_HL_3
Original[7] <- Actual_Pillai_1
Original[8] <- Actual_Pillai_2
Original[9] <- Actual_Pillai_3

Outlier_Z <- 5

# Number of bins specified using the Friedman-Diaconis rule.
for (j in seq(1, 9, 1)) {
    trimmed_data <- outlier_detect(boot_results$t[, j], Outlier_Z)
    plot_data <- as.data.frame(trimmed_data[[1]])
    names(plot_data) <- c("t")
    plot <- ggplot(plot_data, aes(x = t)) + geom_histogram(bins = round((max(plot_data$t) -
        min(plot_data$t))/(2 * IQR(plot_data$t) * length(plot_data$t)^(-1/3))),
        color = "grey30", fill = "grey", size = 0.01, na.rm = TRUE)

    p <- ggplot(plot_data, aes(x = t)) + geom_histogram(bins = round((max(plot_data$t) -
        min(plot_data$t))/(2 * IQR(plot_data$t) * length(plot_data$t)^(-1/3))),
        color = "grey30", fill = "grey", size = 0.25, na.rm = TRUE) +
```

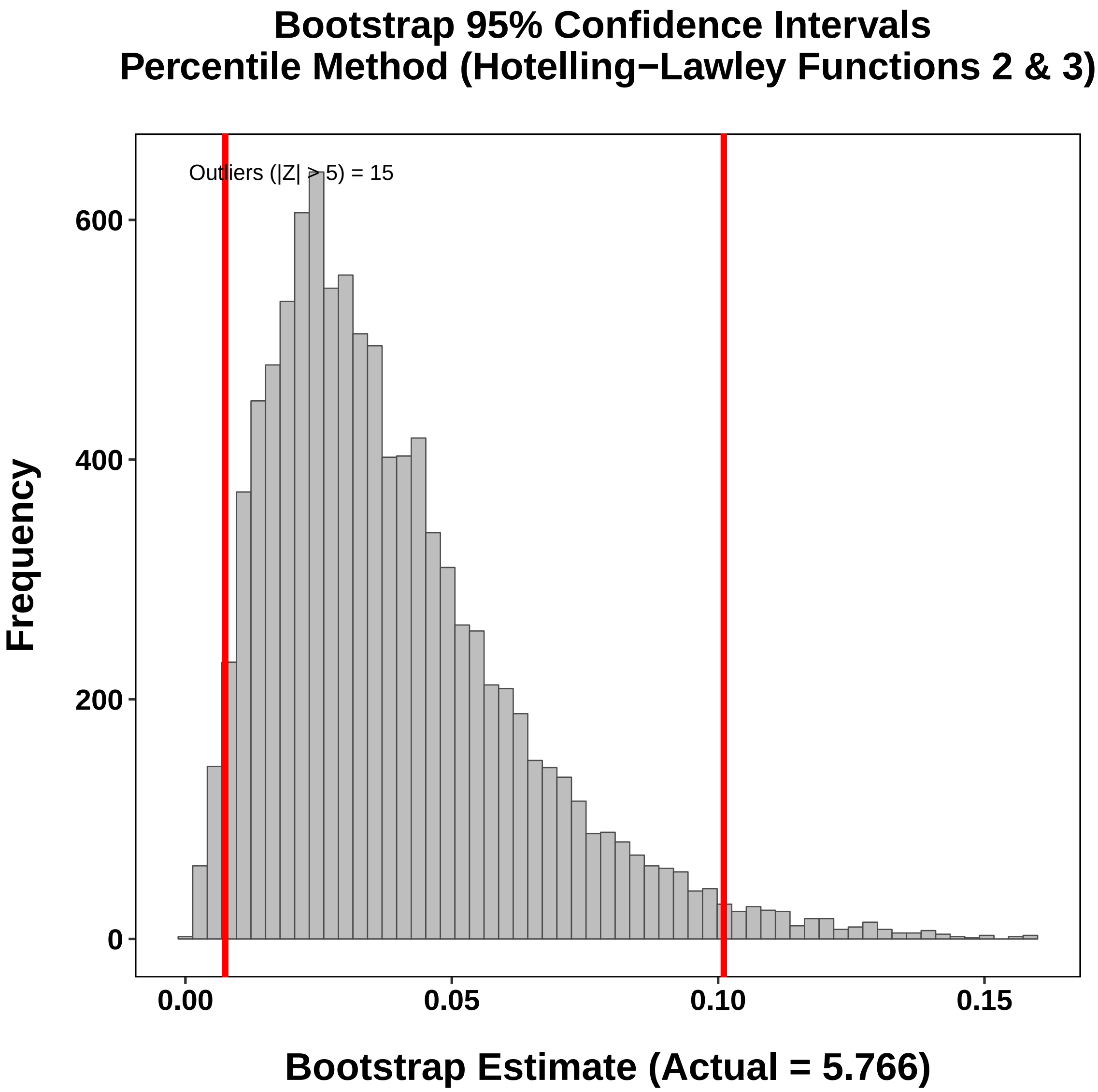
```
        xlab(paste("Bootstrap Estimate (Actual = ", toString(round(Original[j],
            digits = 3)), ")", sep = "")) + ylab("Frequency") + theme(text = element_text(size = 14,
        family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
        size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
        size = 12, angle = 0, face = "bold"), axis.title.x = element_text(margin = margin(15,
        0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
        15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
        plot.title = element_text(size = 16, face = "bold", margin = margin(0,
            0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
            linetype = 1, color = "black"), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
        plot.margin = unit(c(1, 1, 1, 1), "cm")) + geom_vline(xintercept = boot.ci(boot_results,
        type = "perc", index = j)$perc[4], size = 1.25, color = "red") +
        geom_vline(xintercept = boot.ci(boot_results, type = "perc",
            index = j)$perc[5], size = 1.25, color = "red") + annotate("text",
        x = min(plot_data$t), y = max(ggplot_build(plot)$data[[1]]$count),
        label = paste("Outliers (|Z| > ", toString(round(Outlier_Z,
            2)), ") = ", toString(trimmed_data[[4]]), sep = ""), color = "black",
        angle = 0, hjust = 0, size = 3) + ggtitle(paste("Bootstrap 95% Confidence Intervals \nPercentile
        toString(Effects[j]), ")", sep = ""))
    print(p)
}
```

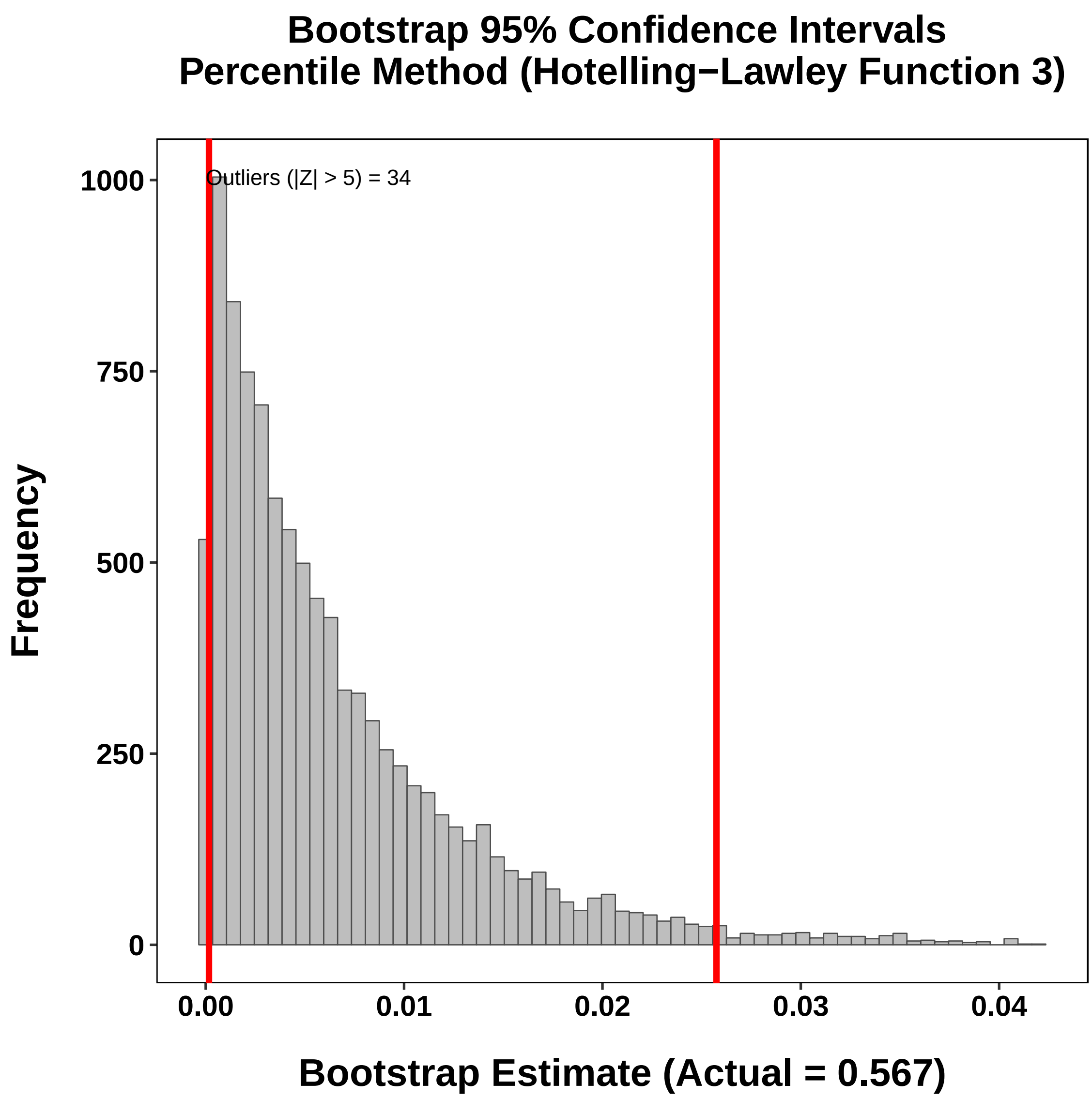

Bootstrap 95% Confidence Intervals
Percentile Method (Wilks Functions 1, 2, & 3)
Outliers (|Z| > 5) = 6
Frequency
600
400
200
0
0.7
0.8
0.9
1.0
Bootstrap Estimate (Actual = 0.014)

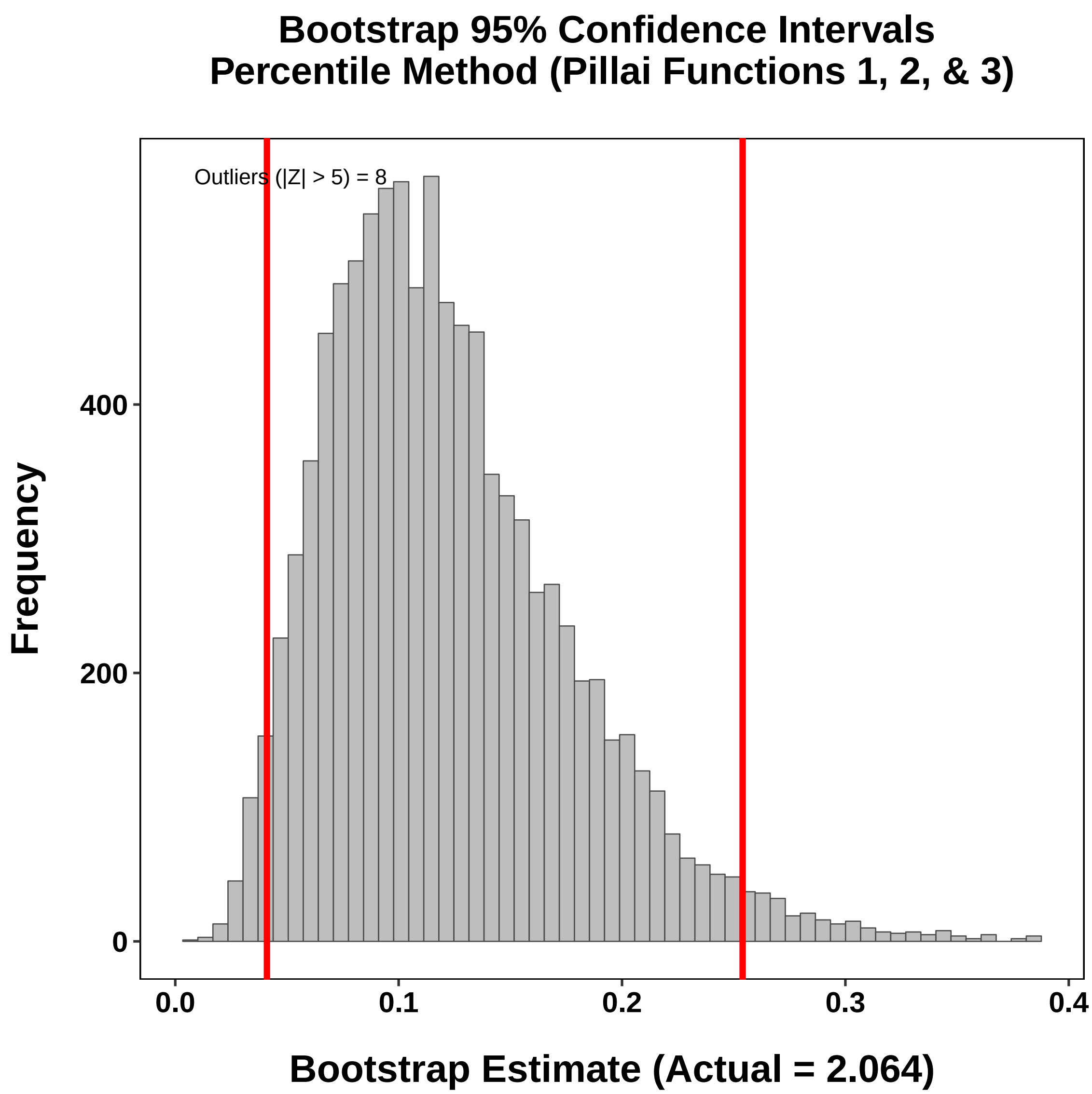

Bootstrap 95% Confidence Intervals
Percentile Method (Wilks Functions 2 & 3)
Outliers (|Z| > 5) = 12
600
400
200
0
Frequency
0.85
0.90
0.95
1.00
Bootstrap Estimate (Actual = 0.103)

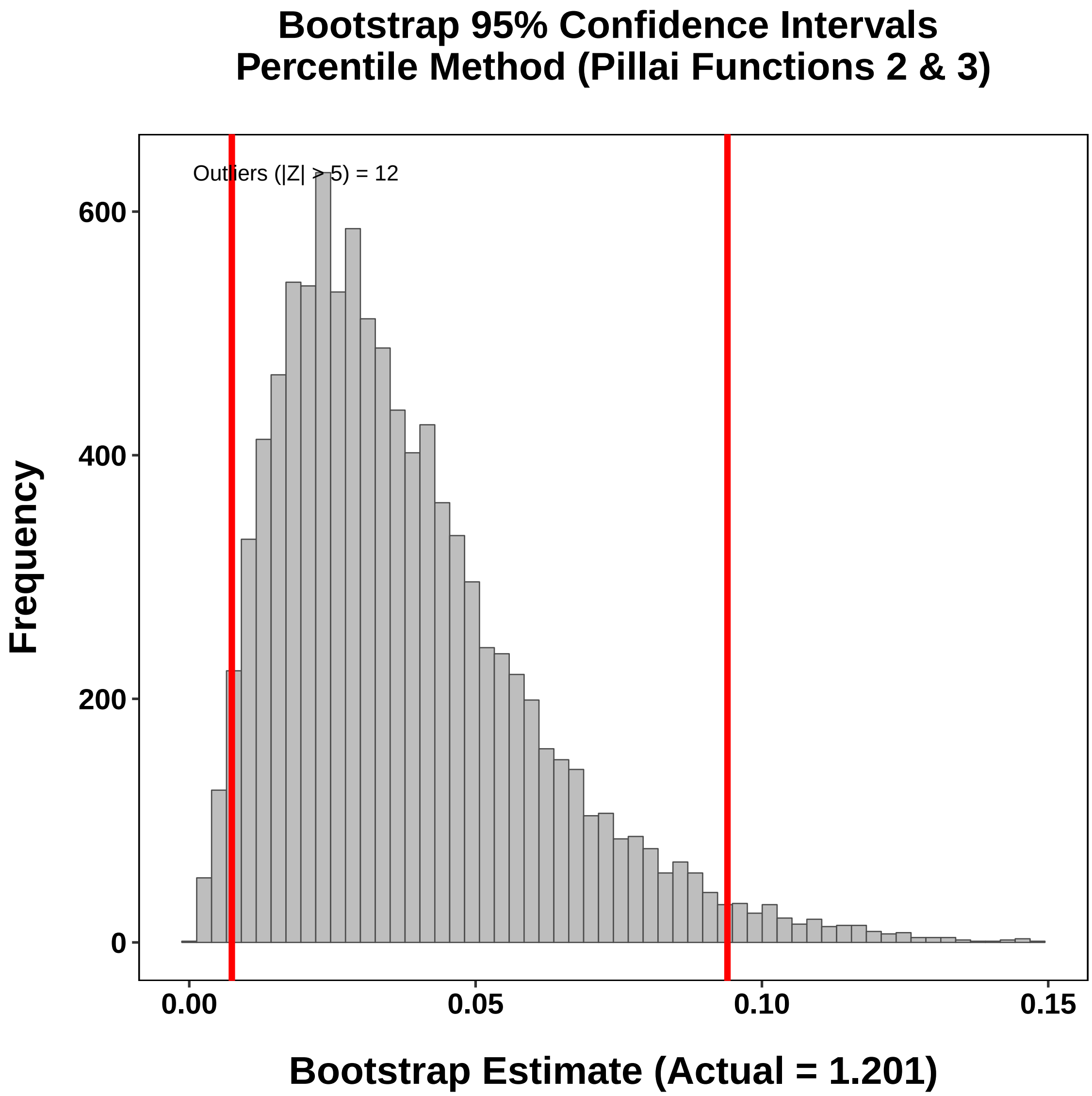

Bootstrap 95% Confidence Intervals
Percentile Method (Wilks Function 3)
Outliers (|Z| > 5) = 29
Frequency
1000
750
500
250
0
0.96
0.97
0.98
0.99
1.00
Bootstrap Estimate (Actual = 0.638)

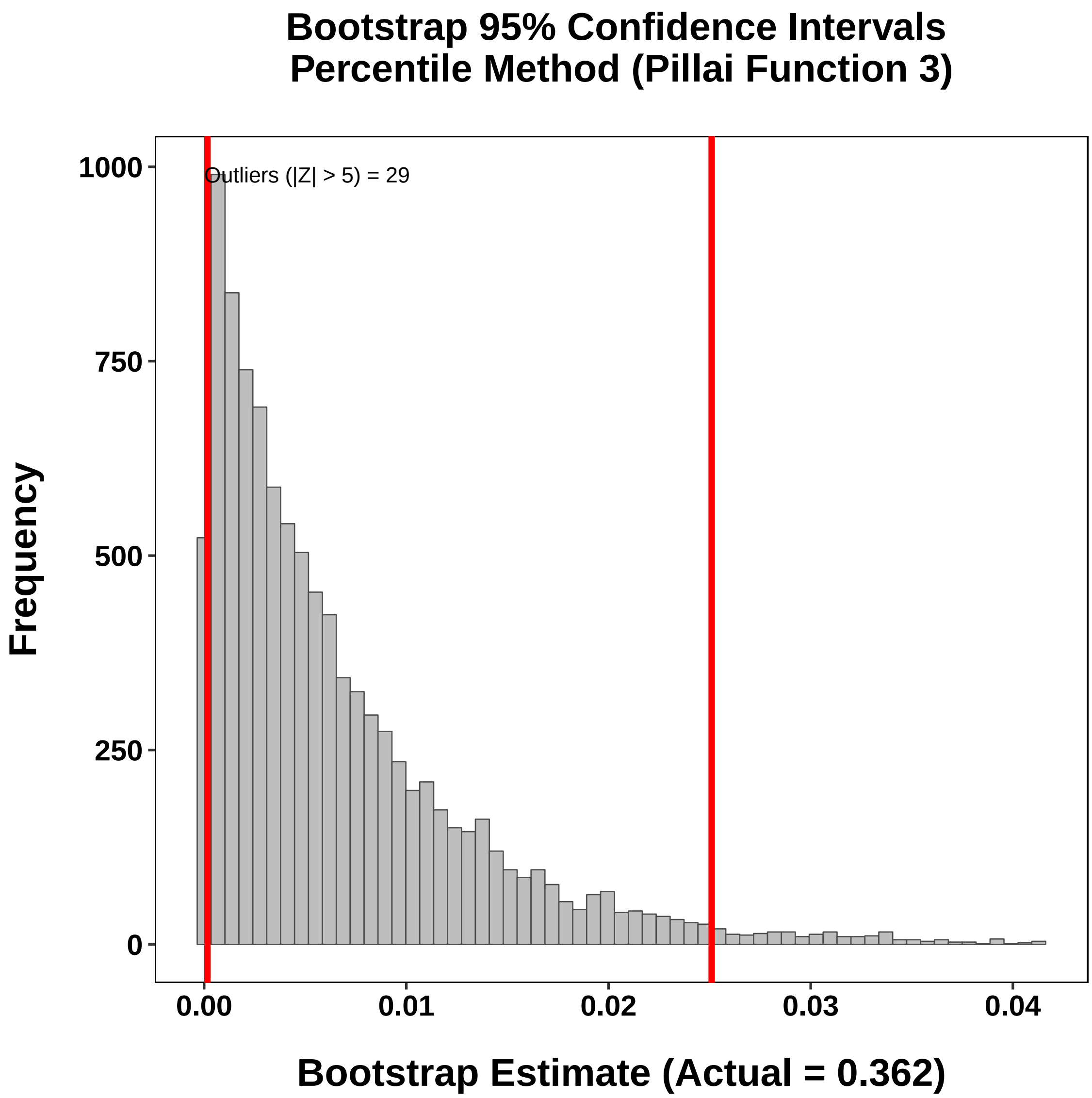

Bootstrap 95% Confidence Intervals
Percentile Method (Hotelling−Lawley Functions 1, 2, & 3)
Outliers (|Z| > 5) = 17
Frequency
0
200
400
600
0.0
0.1
0.2
0.3
0.4
Bootstrap Estimate (Actual = 12.079)


Bootstrap 95% Confidence Intervals
Percentile Method (Hotelling−Lawley Functions 2 & 3)
Outliers (|Z| > 5) = 15
Frequency
600
400
200
0
0.00
0.05
0.10
0.15
Bootstrap Estimate (Actual = 5.766)


Bootstrap 95% Confidence Intervals
Percentile Method (Hotelling−Lawley Function 3)
Outliers (|Z| > 5) = 34
Frequency
1000
750
500
250
0
0.00
0.01
0.02
0.03
0.04
Bootstrap Estimate (Actual = 0.567)


Bootstrap 95% Confidence Intervals
Percentile Method (Pillai Functions 1, 2, & 3)
Outliers (|Z| > 5) = 8
Frequency
0
200
400
0.0
0.1
0.2
0.3
0.4
Bootstrap Estimate (Actual = 2.064)


Bootstrap 95% Confidence Intervals
Percentile Method (Pillai Functions 2 & 3)
Outliers (|Z| > 5) = 12
600
400
200
0
Frequency
0.00
0.05
0.10
0.15
Bootstrap Estimate (Actual = 1.201)



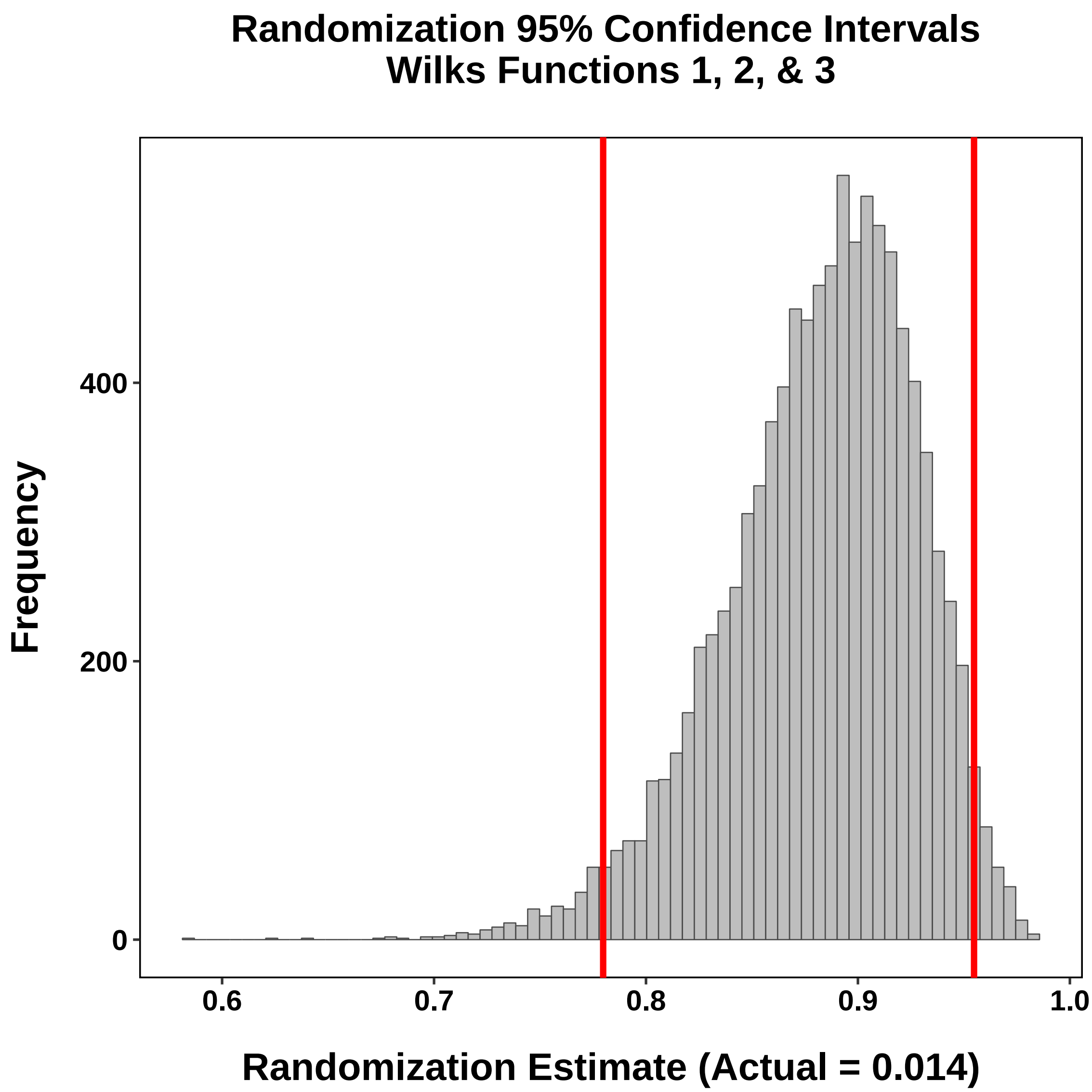
5 Randomization Tests

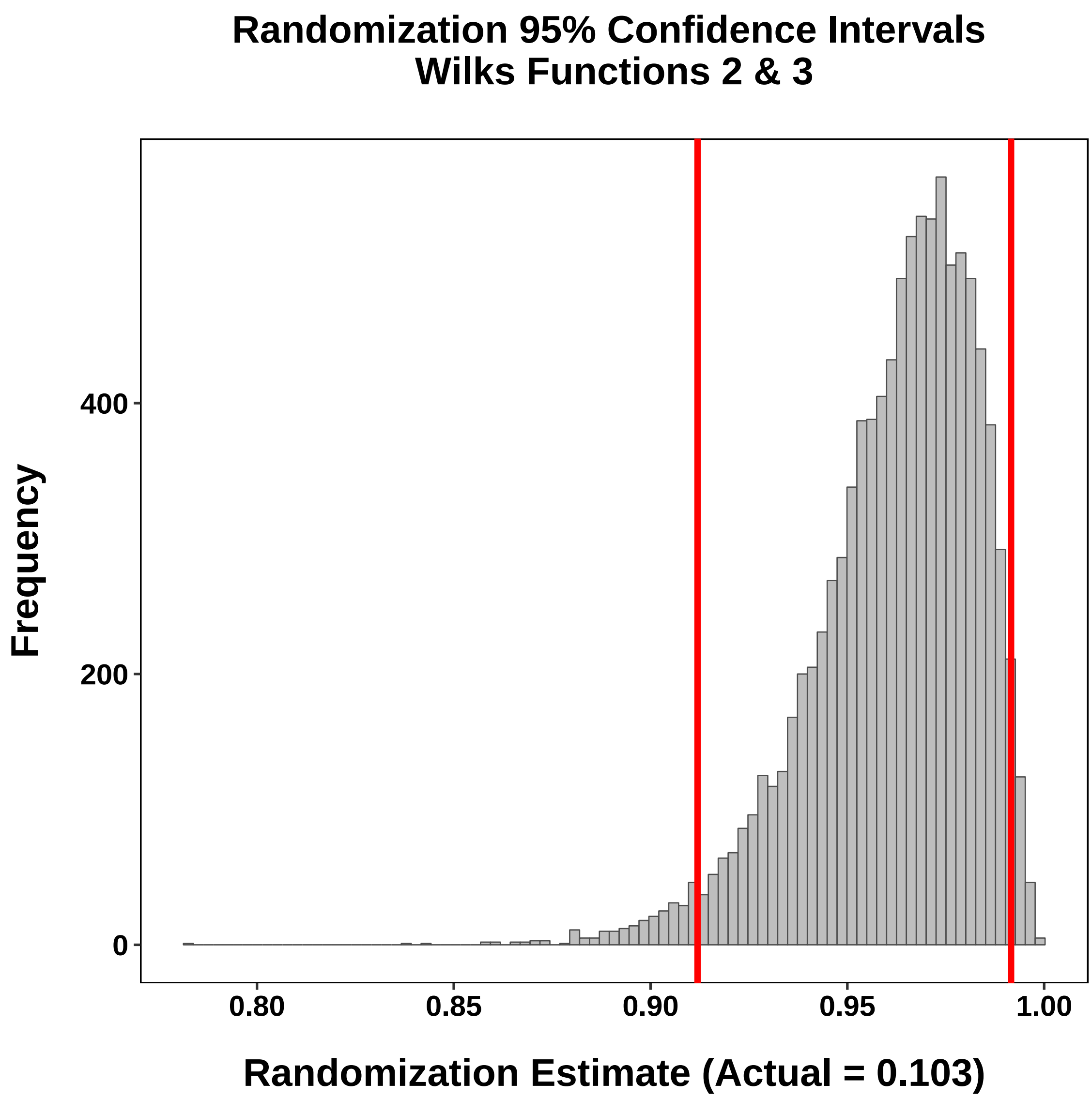
An alternative to the bootstrap takes a different approach to the data. When participants are randomly assigned to groups, we assume under the null hypothesis that these assignment are inconsequential or arbitrary. In a randomization test, we actually make the group assignments arbitrary. On each of a large number of trials, the group assignments are shuffled randomly so that participants will get new group assignments that may not match their original assignments. The statistical analyses are conducted on each of these reshuffled samples and the location of the parameters from the original analysis are compared to the permutation distributions to determine if the original estimates are rare under the null.

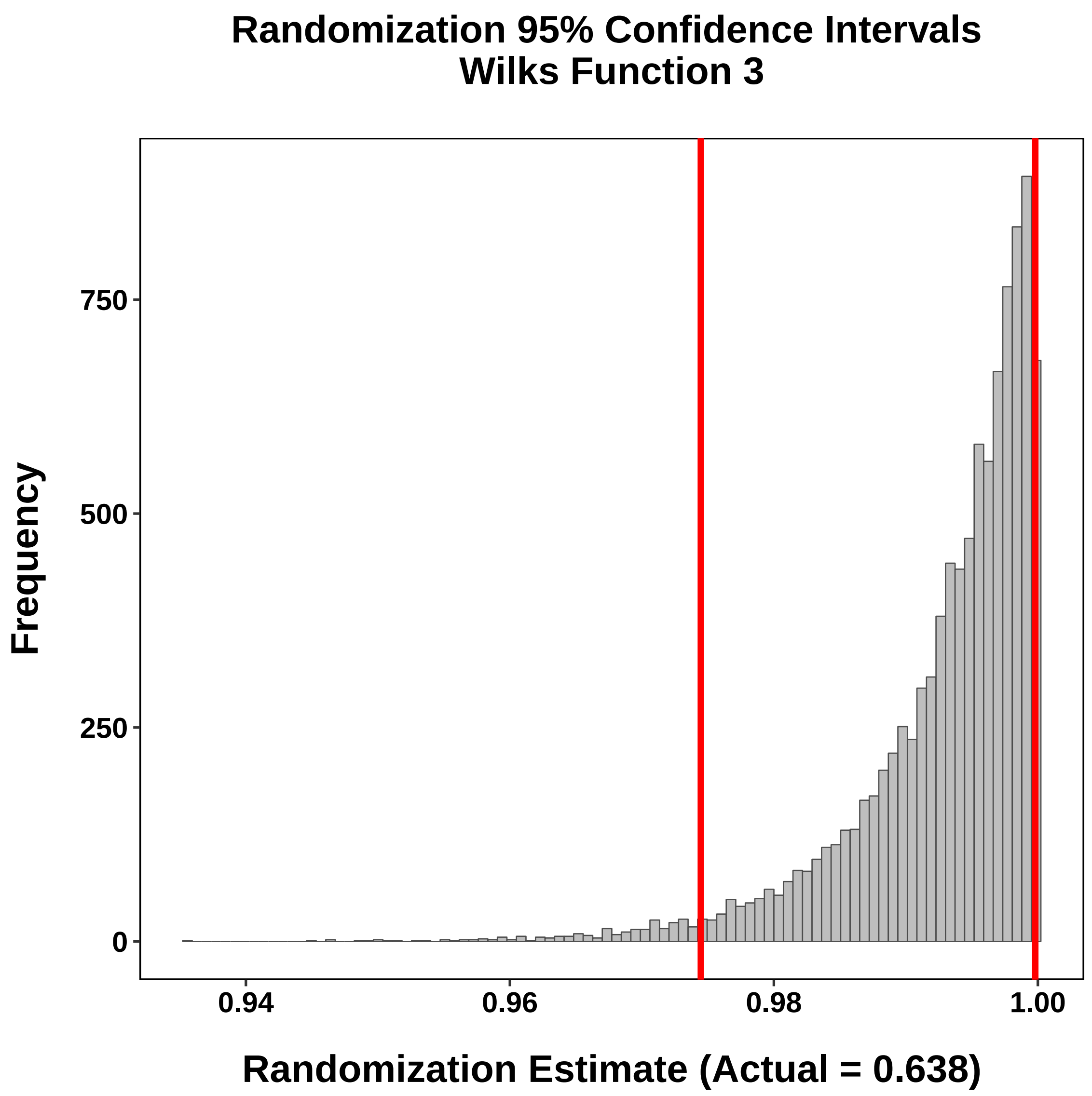
```r
# First, reserve space in the variable, diffs
rand_fits <- matrix(NA, nrow = 10000, ncol = 9)
# Execute a loop 10000 times
for (i in 1:10000) {
    # Shuffle the data in the group assignment vector.
    Skills$Group_perm <- sample(Skills$Group)
    # Run the MANOVA and get the parameter estimates.
    LM_P <- lm(cbind(P_Verbal, P_Quant, C_Verbal, C_Quant) ~ as.factor(Group_perm),
        data = Skills)
    LDA_P <- candisc(LM_P, data = Skills)
    rand_fits[i, 1] <- (1/(1 + LDA_P$eigenvalues[1])) * (1/(1 + LDA_P$eigenvalues[2])) *
        (1/(1 + LDA_P$eigenvalues[3]))
    rand_fits[i, 2] <- (1/(1 + LDA_P$eigenvalues[2])) * (1/(1 + LDA_P$eigenvalues[3]))
    rand_fits[i, 3] <- (1/(1 + LDA_P$eigenvalues[3]))
    rand_fits[i, 4] <- LDA_P$eigenvalues[1] + LDA_P$eigenvalues[2] +
        LDA_P$eigenvalues[3]
    rand_fits[i, 5] <- LDA_P$eigenvalues[2] + LDA_P$eigenvalues[3]
    rand_fits[i, 6] <- LDA_P$eigenvalues[3]
    rand_fits[i, 7] <- (LDA_P$eigenvalues[1]/(1 + LDA_P$eigenvalues[1])) +
        (LDA_P$eigenvalues[2]/(1 + LDA_P$eigenvalues[2])) + (LDA_P$eigenvalues[3]/(1 +
        LDA_P$eigenvalues[3]))
    rand_fits[i, 8] <- (LDA_P$eigenvalues[2]/(1 + LDA_P$eigenvalues[2])) +
        (LDA_P$eigenvalues[3]/(1 + LDA_P$eigenvalues[3]))
    rand_fits[i, 9] <- (LDA_P$eigenvalues[3]/(1 + LDA_P$eigenvalues[3]))
}
```

```r
# Number of bins specified using the Friedman-Diaconis rule.
for (j in seq(1, 9, 1)) {
    plot_data <- as.data.frame(rand_fits[, j])
    names(plot_data) <- c("t")
    plot <- ggplot(plot_data, aes(x = t)) + geom_histogram(bins = round((max(plot_data$t) -
        min(plot_data$t))/(2 * IQR(plot_data$t) * length(plot_data$t)^(-1/3))),
        color = "grey30", fill = "grey", size = 0.01, na.rm = TRUE)

    p <- ggplot(plot_data, aes(x = t)) + geom_histogram(bins = round((max(plot_data$t) -
        min(plot_data$t))/(2 * IQR(plot_data$t) * length(plot_data$t)^(-1/3))),
        color = "grey30", fill = "grey", size = 0.25, na.rm = TRUE) +
        xlab(paste("Randomization Estimate (Actual = ", toString(round(Original[j],
            digits = 3)), ")", sep = "")) + ylab("Frequency") + theme(text = element_text(size = 14,
        family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
        size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
        size = 12, angle = 0, face = "bold"), axis.title.x = element_text(margin = margin(15,
        0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
        15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
        plot.title = element_text(size = 16, face = "bold", margin = margin(0,
            0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
            linetype = 1, color = "black"), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
        plot.margin = unit(c(1, 1, 1, 1), "cm")) + geom_vline(xintercept = quantile(rand_fits[,
        j], c(0.025)), size = 1.25, color = "red") + geom_vline(xintercept = quantile(rand_fits[,
        j], c(0.975)), size = 1.25, color = "red") + ggtitle(paste("Randomization 95% Confidence Interva
        toString(Effects[j]), sep = ""))
    print(p)
```

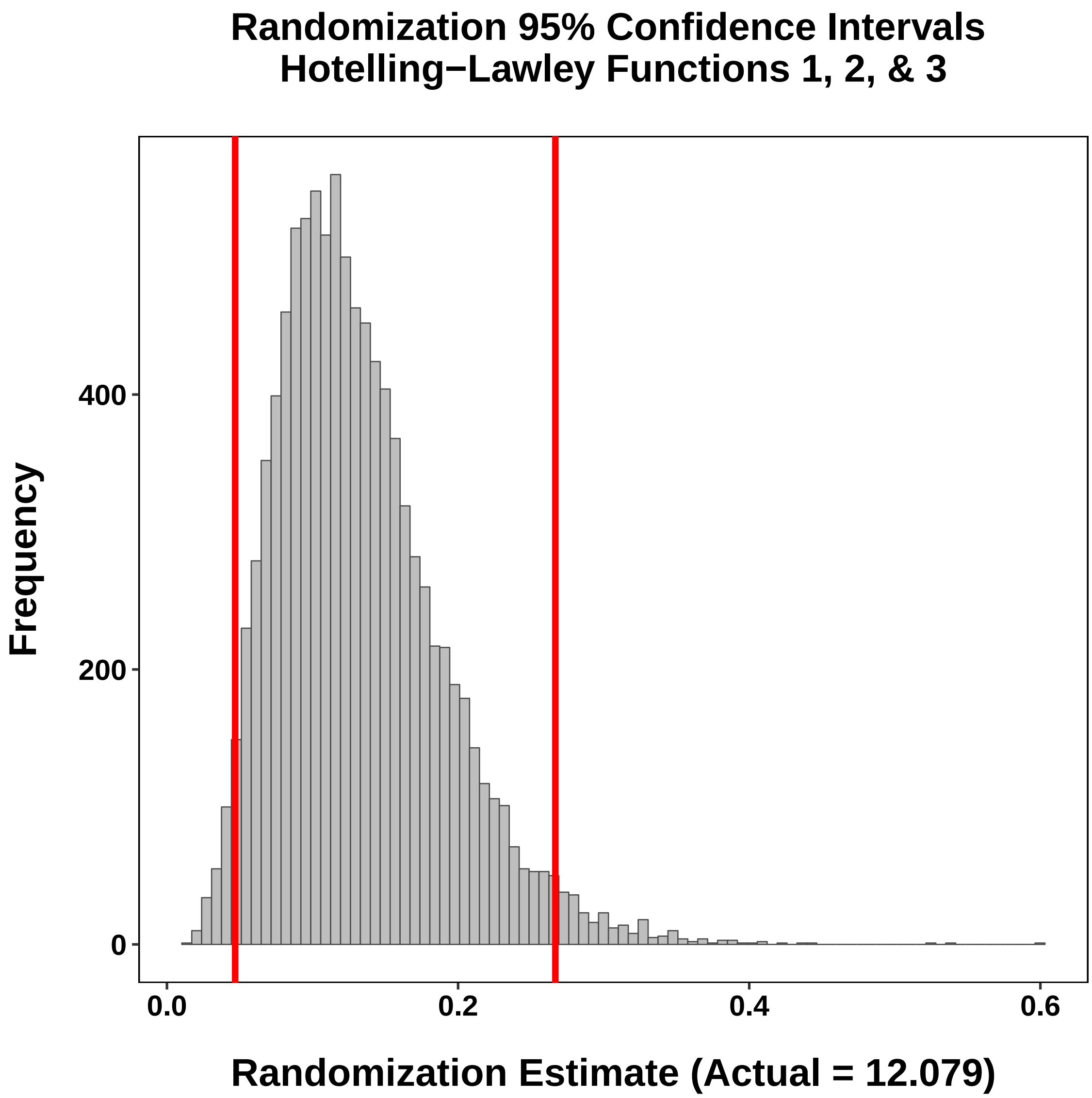
```
}
```

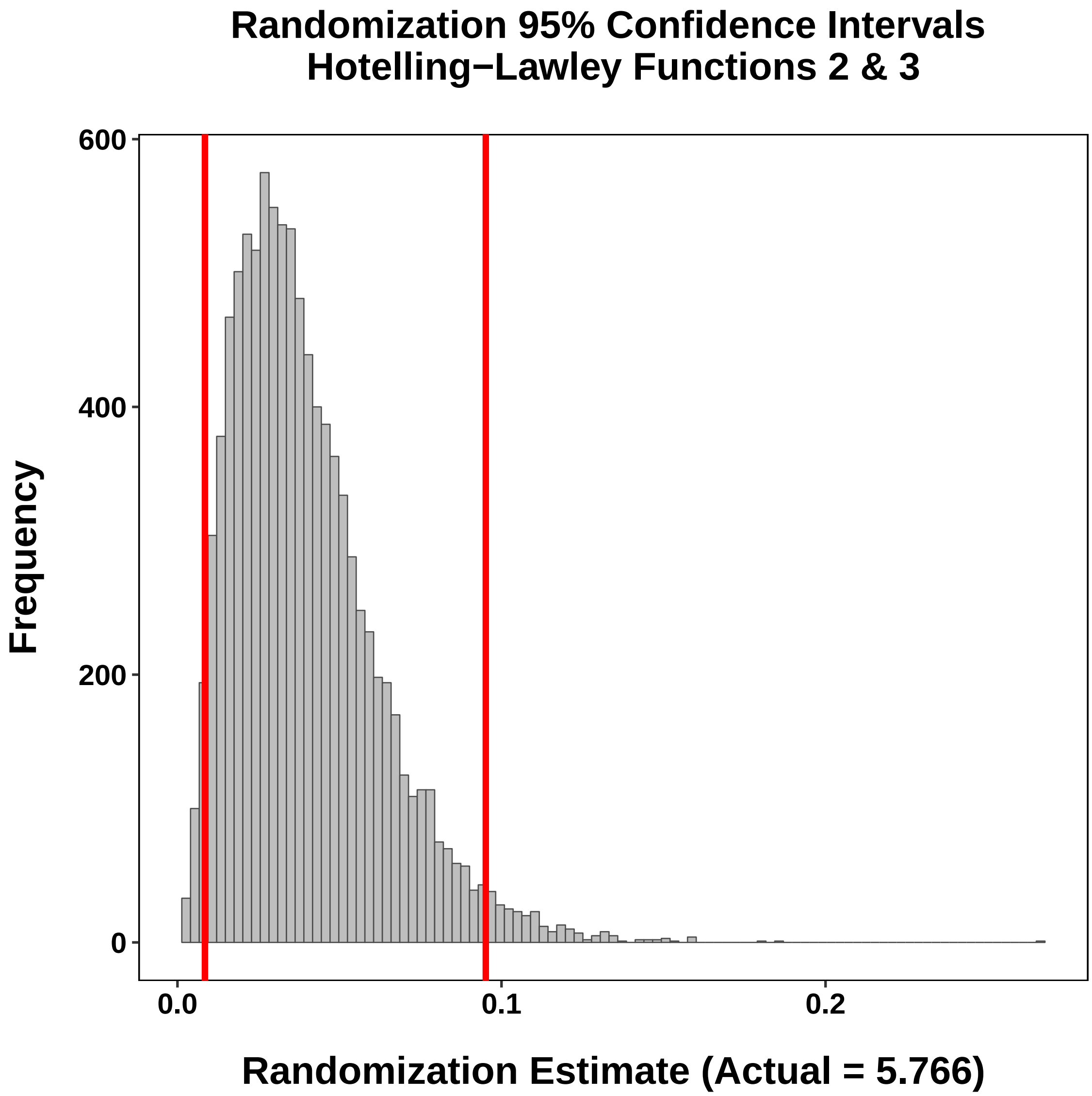

Randomization 95% Confidence Intervals
Wilks Functions 1, 2, & 3
Frequency
0
200
400
0.6
0.7
0.8
0.9
1.0
Randomization Estimate (Actual = 0.014)


Randomization 95% Confidence Intervals
Wilks Functions 2 & 3
Frequency
400
200
0
0.80
0.85
0.90
0.95
1.00
Randomization Estimate (Actual = 0.103)

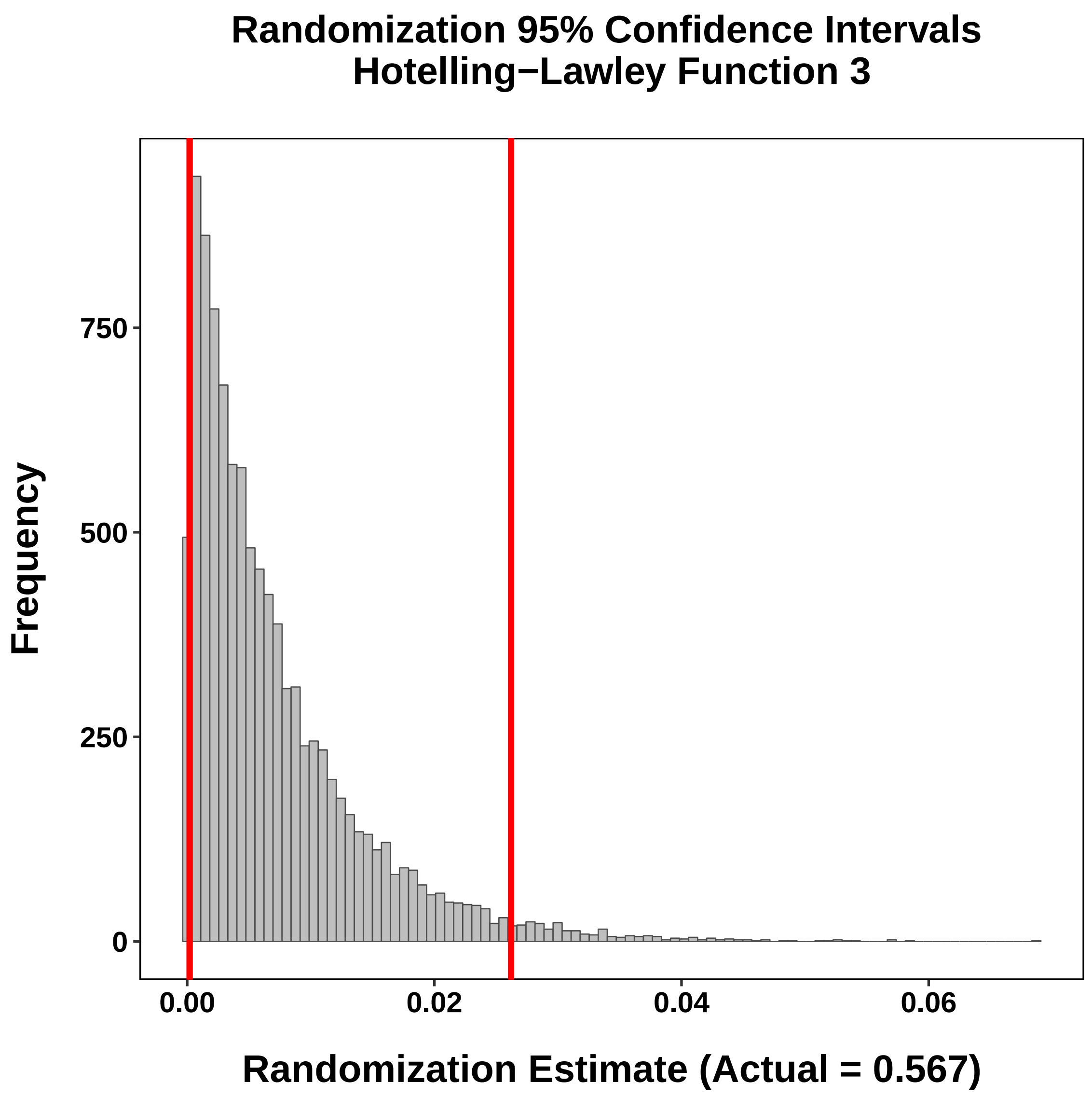


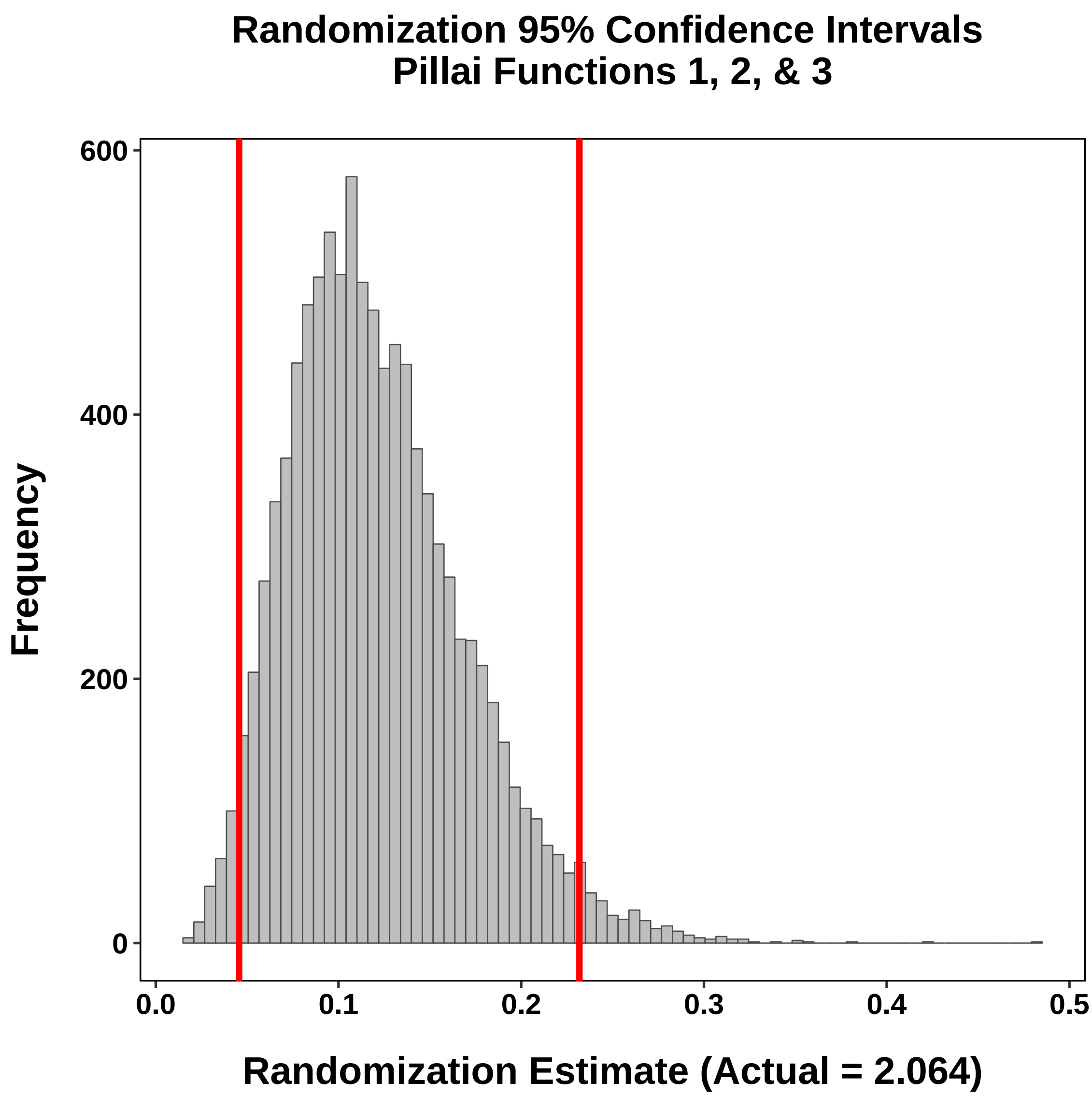
Randomization 95% Confidence Intervals
Wilks Function 3
Frequency
750
500
250
0
0.94
0.96
0.98
1.00
Randomization Estimate (Actual = 0.638)

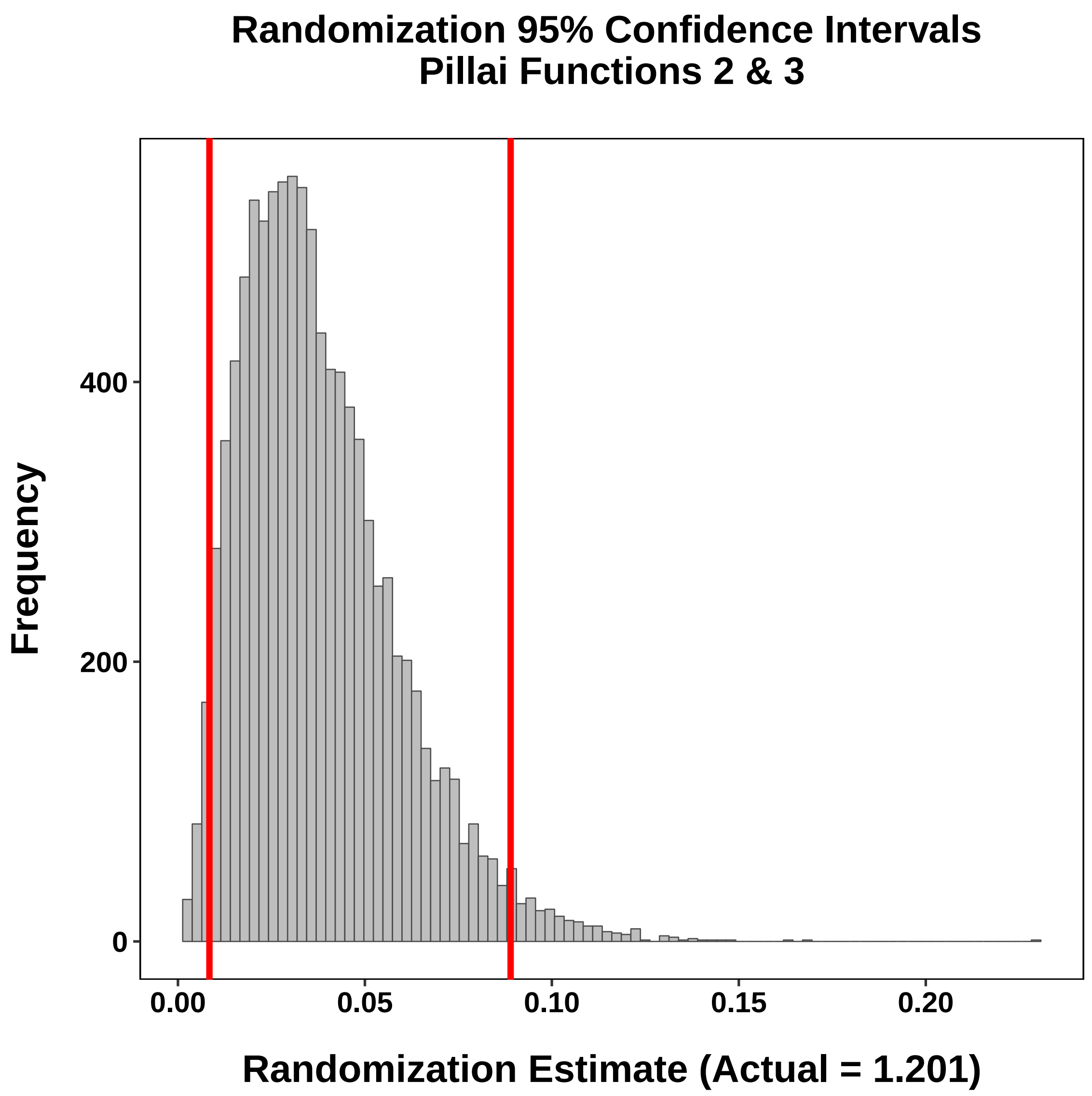

Randomization 95% Confidence Intervals
Hotelling−Lawley Functions 1, 2, & 3
Frequency
400
200
0
0.0
0.2
0.4
0.6
Randomization Estimate (Actual = 12.079)



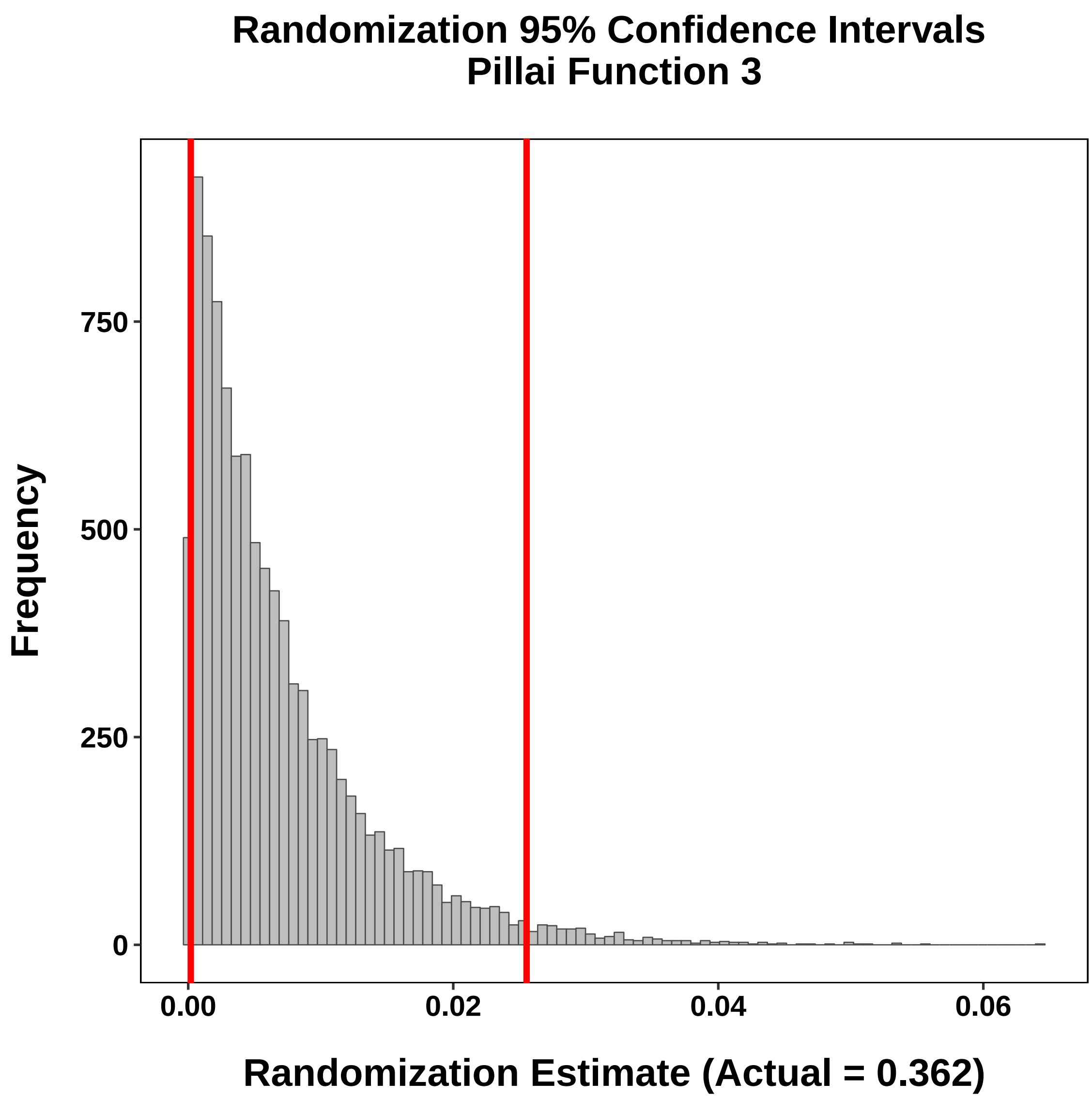
Randomization 95% Confidence Intervals
Hotelling−Lawley Functions 2 & 3
Frequency
600
400
200
0
0.0
0.1
0.2
Randomization Estimate (Actual = 5.766)


Randomization 95% Confidence Intervals
Hotelling−Lawley Function 3
Frequency
750
500
250
0
0.00
0.02
0.04
0.06
Randomization Estimate (Actual = 0.567)


Randomization 95% Confidence Intervals
Pillai Functions 1, 2, & 3
Frequency
600
400
200
0
0.0
0.1
0.2
0.3
0.4
0.5
Randomization Estimate (Actual = 2.064)



Randomization 95% Confidence Intervals
Pillai Functions 2 & 3
Frequency
400
200
0
0.00
0.05
0.10
0.15
0.20
Randomization Estimate (Actual = 1.201)



6 MANCOVA

Adding covariates to a multivariate analysis is a direct extension of univariate analysis of covariance. The goals are (a) to reduce the error for testing a target effect and (b) to adjust for differences that might otherwise make inferences ambiguous. The interest is in asking a question that begins, "controlling for . . ."

```
LM_2 <- lm(cbind(C_Verbal, C_Quant) ~ Tx_C, data = Skills_Trimmed)
LDA_2 <- candisc(LM_2, data = Skills_Trimmed)
LDA_2

##
## Canonical Discriminant Analysis for Tx_C:
```

```
##
##   CanRsq Eigenvalue Difference Percent Cumulative
## 1  0.543       1.19                100        100
##
## Test of H0: The canonical correlations in the
## current row and all that follow are zero
##
##   LR test stat approx F numDF denDF Pr(> F)
## 1        0.457              2
```

```
summary(LDA_2)
```

```
##
## Canonical Discriminant Analysis for Tx_C:
##
##   CanRsq Eigenvalue Difference Percent Cumulative
## 1 0.5426      1.186                100        100
##
## Class means:
##
## [1] -1.089  1.067
##
##  std coefficients:
## C_Verbal  C_Quant
##   -1.204    2.007
```

```
DA_Chi_Square(Skills_Trimmed, LDA_2)
```

```
##   Chi_Sq df         p
## 1  75.09  2 4.939e-17
```

```
LDA_2$coeffs.std
```

```
##            Can1
## C_Verbal -1.204
## C_Quant   2.007
```

```
LDA_2$structure
```

```
##            Can1
## C_Verbal 0.7887
## C_Quant  0.9457
```

```
summary(aov(C_Verbal ~ C_Quant + Tx_C, data = Skills_Trimmed))
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## C_Quant      1  27482   27482     915 < 2e-16
## Tx_C         1    362     362      12 0.00078
## Residuals   96   2884      30
```

```
summary(aov(C_Quant ~ C_Verbal + Tx_C, data = Skills_Trimmed))
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## C_Verbal     1  34728   34728    1177 < 2e-16
## Tx_C         1   1270    1270      43 2.7e-09
## Residuals   96   2831      29
```

```
MANOVA_2 <- manova(cbind(C_Verbal, C_Quant) ~ Tx_C, data = Skills_Trimmed)
summary(MANOVA_2, test = "Wilks")

##           Df Wilks approx F num Df den Df Pr(>F)
## Tx_C       1 0.457     56.9      2     96 <2e-16
## Residuals 97

cor(MANOVA_2$residuals)

##          C_Verbal C_Quant
## C_Verbal   1.0000  0.9265
## C_Quant    0.9265  1.0000
```

```
LM_3 <- lm(cbind(C_Verbal, C_Quant) ~ P_Verbal + P_Quant + Tx_C, data = Skills_Trimmed)
LDA_3 <- candisc(LM_3, data = Skills_Trimmed)
LDA_3

##
## Canonical Discriminant Analysis for P_Verbal:
##
##   CanRsq Eigenvalue Difference Percent Cumulative
## 1  0.108      0.121                100        100
##
## Test of H0: The canonical correlations in the
## current row and all that follow are zero
##
##   LR test stat approx F numDF denDF Pr(> F)
## 1        0.892              2

summary(LDA_3)

##
## Canonical Discriminant Analysis for P_Verbal:
##
##   CanRsq Eigenvalue Difference Percent Cumulative
## 1  0.108      0.121                100        100
##
## Class means:
##
##  [1] -2.07733 -1.12737 -0.96809  1.11177 -0.76891  0.94509
##  [7] -1.67703 -0.54440 -0.04980 -0.16548 -1.61833  2.00621
## [13] -1.42111  0.73406 -2.92963 -0.07477 -1.76196 -3.26386
## [19]  0.32677 -1.98246  0.14156  0.69832 -0.59869 -1.02098
## [25] -0.57881 -2.32004 -5.28783  0.02196  1.79768  1.46932
## [31] -0.13263 -4.24577 -3.00343 -3.94610 -1.43137 -3.27376
## [37]  0.03150 -2.81993 -1.50987 -2.99616 -3.80089 -1.89121
## [43] -2.18922 -0.52037 -2.23641 -1.99809 -2.84430 -2.95304
## [49] -2.23475 -1.80165 -2.12236 -0.98665 -3.44174 -1.83355
## [55] -2.00442 -2.35417 -2.87157 -1.27766 -2.83772 -2.48327
## [61] -2.73408 -0.85676 -1.49041 -0.90520 -0.49126 -0.06572
## [67] -0.75873 -1.13402  1.03594 -0.04021 -0.47634  2.26263
## [73]  3.36477  0.28491  3.18238  0.45611  2.94375  3.62673
## [79]  0.35456  2.85599  4.56857  2.72739  4.15097  3.82557
## [85]  3.22461  4.85226  3.60613  3.35152  4.93858  2.75445
```

```
## [91]  3.60100  4.95182  4.94353  4.18884  4.37888
##
##  std coefficients:
## C_Verbal  C_Quant
## -0.01967  1.01462
```

```
DA_Chi_Square(Skills_Trimmed, LDA_3)
```

```
##   Chi_Sq df        p
## 1  10.97  2 0.004154
```

```
LDA_3$coeffs.std
```

```
##               Can1
## C_Verbal -0.01967
## C_Quant   1.01462
```

```
LDA_3$structure
```

```
##           Can1
## C_Verbal 0.944
## C_Quant  1.000
```

```
summary(aov(C_Verbal ~ C_Quant + P_Verbal + P_Quant + Tx_C, data = Skills_Trimmed))
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## C_Quant      1  27482   27482   957.6 < 2e-16
## P_Verbal     1    444     444    15.5 0.00016
## P_Quant      1     98      98     3.4 0.06835
## Tx_C         1      6       6     0.2 0.65456
## Residuals   94   2698      29
```

```
summary(aov(C_Quant ~ C_Verbal + P_Verbal + P_Quant + Tx_C, data = Skills_Trimmed))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## C_Verbal     1  34728   34728 1354.09 <2e-16
## P_Verbal     1     28      28    1.10 0.2964
## P_Quant      1    230     230    8.98 0.0035
## Tx_C         1   1432    1432   55.83  4e-11
## Residuals   94   2411      26
```

```
MANOVA_3 <- manova(cbind(C_Verbal, C_Quant) ~ P_Verbal + P_Quant +
    Tx_C, data = Skills_Trimmed)
summary(MANOVA_3, test = "Wilks")
```

```
##           Df Wilks approx F num Df den Df Pr(>F)
## P_Verbal   1 0.243    146.7      2     94 <2e-16
## P_Quant    1 0.863      7.4      2     94  0.001
## Tx_C       1 0.246    144.0      2     94 <2e-16
## Residuals 95
```

```
cor(MANOVA_3$residuals)
```

```
##          C_Verbal C_Quant
## C_Verbal   1.0000  0.7476
## C_Quant    0.7476  1.0000
```

```
LM_4 <- lm(cbind(C_Verbal, C_Quant) ~ P_Verbal + P_Quant + Tx_C +
    Tx_C:P_Verbal + Tx_C:P_Quant, data = Skills_Trimmed)
LDA_4 <- candisc(LM_4, data = Skills_Trimmed)
LDA_4

##
## Canonical Discriminant Analysis for P_Verbal:
##
##   CanRsq Eigenvalue Difference Percent Cumulative
## 1 0.0517     0.0545                100        100
##
## Test of H0: The canonical correlations in the
## current row and all that follow are zero
##
##   LR test stat approx F numDF denDF Pr(> F)
## 1        0.948              2

summary(LDA_4)

##
## Canonical Discriminant Analysis for P_Verbal:
##
##   CanRsq Eigenvalue Difference Percent Cumulative
## 1 0.0517    0.05452                100        100
##
## Class means:
##
##  [1] -0.40950 -0.90295  1.39893  1.91989  0.36383  1.05448
##  [7] -2.07411 -0.37287  0.10658  0.12710 -0.25612  3.14529
## [13]  0.33842  0.86526 -1.93431 -0.15168 -1.80101 -2.24200
## [19]  0.30224 -0.10578  0.17843  0.32422 -0.81487  0.09943
## [25]  0.09277 -2.76121 -3.43904 -0.93116  1.10824  1.16859
## [31]  0.02050 -3.21488 -2.27856 -3.49121 -0.86832 -2.54037
## [37]  0.26012 -0.92064 -0.53064 -1.59321 -2.81289 -1.62903
## [43] -1.76391 -0.60856 -1.32387 -2.03077 -2.88576 -2.40897
## [49] -1.65772 -1.95111 -2.77302 -0.86867 -2.08907 -1.43724
## [55] -1.24861 -1.11824 -3.33457 -1.56034 -1.74913 -1.55886
## [61] -2.12608 -1.23982 -1.52827 -1.59013 -0.62136 -1.68678
## [67] -0.24888 -1.43929 -0.14331  0.36000 -0.52070  2.37180
## [73]  4.05802 -0.51084  2.02819  0.07376  3.37335  2.29182
## [79]  0.74618  1.37024  4.33112  1.92596  2.25767  2.52329
## [85]  1.87188  3.48936  3.55727  2.18633  2.89679  1.46919
## [91]  2.22361  3.55435  3.14274  2.58040  2.90851
##
##  std coefficients:
## C_Verbal  C_Quant
##  -0.8716   1.4719

DA_Chi_Square(Skills_Trimmed, LDA_4)

##   Chi_Sq df       p
## 1  5.096  2 0.07823

LDA_4$coeffs.std
```

```
##             Can1
## C_Verbal -0.8716
## C_Quant   1.4719
```

```
LDA_4$structure
```

```
##            Can1
## C_Verbal 0.7723
## C_Quant  0.9368
```

```
MANOVA_4 <- manova(cbind(C_Verbal, C_Quant) ~ P_Verbal + P_Quant +
    Tx_C + Tx_C:P_Verbal + Tx_C:P_Quant, data = Skills_Trimmed)
summary(MANOVA_4, test = "Wilks")
```

```
##                Df Wilks approx F num Df den Df     Pr(>F)
## P_Verbal        1 0.219    164.0      2     92    < 2e-16
## P_Quant         1 0.863      7.3      2     92     0.0011
## Tx_C            1 0.237    148.0      2     92    < 2e-16
## P_Verbal:Tx_C   1 0.995      0.2      2     92     0.7898
## P_Quant:Tx_C    1 0.737     16.4      2     92 0.00000082
## Residuals      93
```

7 Profile Analysis

Extending MANOVA to repeated measures has some advantages. One of the simplest is a profile analysis. In a profile analysis, several different measures that use the same scale are compared—their profile is assessed, often by comparing groups. A key assumption is that the measures can be directly compared—their metrics are the same. This is no problem when the "different measures" are simple replications over time. The assumption is very important to consider when the measures are truly different conceptually.

When the same measures are collected over time, a profile analysis is the multivariate approach to repeated measures that might be used if the assumptions for a univariate repeated measures ANOVA (e.g., sphericity) are not met. When used with different measures, collected on one occasion with a common measurement scale, a profile analysis addresses questions such as "is X elevated relative to Y and Z?" If groups are included, the question becomes "is the difference between X and Y greater in Group A than in Group B?"

A profile analysis addresses three major questions:

(a) Are the profiles parallel? This addresses whether the pattern of differences across measures is similar for the several groups compared (the interaction).

(b) If the profiles are parallel, are they coincident? Coincident and parallel profiles will have no group differences in the between-subjects part of the design—there will be no group main effect).

(c) If the profiles are parallel, are they also level? The flatness of the profiles is addressed by collapsing across groups and testing for whether the measures are similar in their means (the within-subjects part of the design—a main effect).

```
Profile_1 <- manova(as.matrix(Profiles[, 1:11]) ~ Profiles[, 12],
    data = Profiles)
summary(Profile_1, test = "Wilks")
```

```
##                  Df Wilks approx F num Df den Df Pr(>F)
## Profiles[, 12]    1 0.906     1.44     11    152   0.16
## Residuals       162
```

```
Measure <- factor(c("info", "comp", "arith", "simil", "vocab", "digit",
    "pictcomp", "parang", "block", "object", "coding"), levels = c("info",
    "comp", "arith", "simil", "vocab", "digit", "pictcomp", "parang",
    "block", "object", "coding"))
idata <- data.frame(Measure)

LM_1 <- lm(cbind(info, comp, arith, simil, vocab, digit, pictcomp,
    parang, block, object, coding) ~ Profiles$AgeMate, data = Profiles)
ANOVA_1 <- Anova(LM_1, idata = idata, idesign = ~Measure, type = 2)
summary(ANOVA_1, multivariate = FALSE)
```

```
##
## Univariate Type II Repeated-Measures ANOVA Assuming Sphericity
##
##                          Sum Sq num Df Error SS den Df F value
## (Intercept)              179322      1     4916    161 5872.55
## Profiles$AgeMate             50      2     4916    161    0.81
## Measure                    1190     10     9408   1610   20.37
## Profiles$AgeMate:Measure    218     20     9408   1610    1.87
##                          Pr(>F)
## (Intercept)              <2e-16
## Profiles$AgeMate          0.446
## Measure                  <2e-16
## Profiles$AgeMate:Measure  0.011
##
##
## Mauchly Tests for Sphericity
##
##                          Test statistic  p-value
## Measure                           0.281 1.32e-18
## Profiles$AgeMate:Measure          0.281 1.32e-18
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
##  for Departure from Sphericity
##
##                          GG eps Pr(>F[GG])
## Measure                   0.771     <2e-16
## Profiles$AgeMate:Measure  0.771      0.021
##
##                          HF eps Pr(>F[HF])
## Measure                  0.8135  1.130e-29
## Profiles$AgeMate:Measure 0.8135  1.887e-02
```

```
MANOVA_1 <- Manova(LM_1, idata = idata, idesign = ~Measure, type = 2)
summary(MANOVA_1)

##
## Type II Repeated Measures MANOVA Tests:
##
## ------------------------------------------
##
## Term: (Intercept)
##
##  Response transformation matrix:
##          (Intercept)
## info               1
## comp               1
## arith              1
## simil              1
## vocab              1
## digit              1
## pictcomp           1
## parang             1
## block              1
## object             1
## coding             1
##
## Sum of squares and products for the hypothesis:
##             (Intercept)
## (Intercept)     1972538
##
## Multivariate Tests: (Intercept)
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            1      0.97     5873      1    161 <2e-16
## Wilks             1      0.03     5873      1    161 <2e-16
## Hotelling-Lawley  1     36.48     5873      1    161 <2e-16
## Roy               1     36.48     5873      1    161 <2e-16
##
## ------------------------------------------
##
## Term: Profiles$AgeMate
##
##  Response transformation matrix:
##          (Intercept)
## info               1
## comp               1
## arith              1
## simil              1
## vocab              1
## digit              1
## pictcomp           1
## parang             1
## block              1
## object             1
## coding             1
##
## Sum of squares and products for the hypothesis:
```

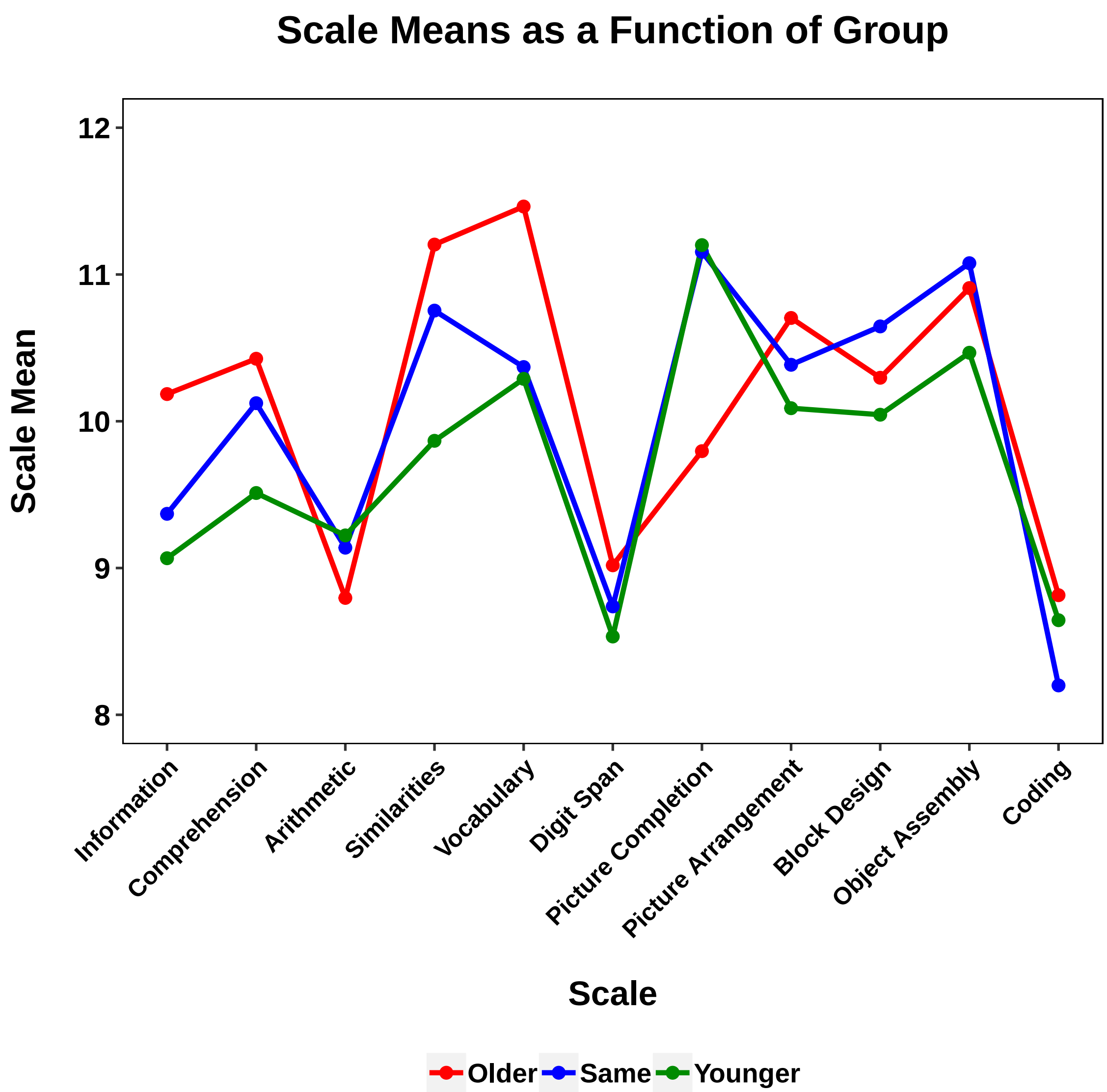
```
##             (Intercept)
## (Intercept)       545.7
##
## Multivariate Tests: Profiles$AgeMate
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            2    0.0100   0.8124      2    161  0.446
## Wilks             2    0.9900   0.8124      2    161  0.446
## Hotelling-Lawley  2    0.0101   0.8124      2    161  0.446
## Roy               2    0.0101   0.8124      2    161  0.446
##
## ------------------------------------------
##
## Term: Measure
##
##  Response transformation matrix:
##          Measure1 Measure2 Measure3 Measure4 Measure5 Measure6
## info            1        0        0        0        0        0
## comp            0        1        0        0        0        0
## arith           0        0        1        0        0        0
## simil           0        0        0        1        0        0
## vocab           0        0        0        0        1        0
## digit           0        0        0        0        0        1
## pictcomp        0        0        0        0        0        0
## parang          0        0        0        0        0        0
## block           0        0        0        0        0        0
## object          0        0        0        0        0        0
## coding         -1       -1       -1       -1       -1       -1
##          Measure7 Measure8 Measure9 Measure10
## info            0        0        0         0
## comp            0        0        0         0
## arith           0        0        0         0
## simil           0        0        0         0
## vocab           0        0        0         0
## digit           0        0        0         0
## pictcomp        1        0        0         0
## parang          0        1        0         0
## block           0        0        1         0
## object          0        0        0         1
## coding         -1       -1       -1        -1
##
## Sum of squares and products for the hypothesis:
##           Measure1 Measure2 Measure3 Measure4 Measure5 Measure6
## Measure1    174.15   258.65    88.62    360.7    368.9    42.25
## Measure2    258.65   384.15   131.62    535.7    547.9    62.75
## Measure3     88.62   131.62    45.10    183.5    187.7    21.50
## Measure4    360.67   535.67   183.54    747.0    764.0    87.50
## Measure5    368.91   547.91   187.73    764.0    781.5    89.50
## Measure6     42.25    62.75    21.50     87.5     89.5    10.25
## Measure7    370.98   550.98   188.78    768.3    785.9    90.00
## Measure8    318.42   472.92   162.04    659.5    674.5    77.25
## Measure9    311.21   462.21   158.37    644.5    659.2    75.50
## Measure10   393.65   584.65   200.32    815.2    833.9    95.50
##           Measure7 Measure8 Measure9 Measure10
```

```
## Measure1     371.0   318.42    311.2     393.6
## Measure2     551.0   472.92    462.2     584.6
## Measure3     188.8   162.04    158.4     200.3
## Measure4     768.3   659.45    644.5     815.2
## Measure5     785.9   674.52    659.2     833.9
## Measure6      90.0    77.25     75.5      95.5
## Measure7     790.2   678.29    662.9     838.5
## Measure8     678.3   582.20    569.0     719.7
## Measure9     662.9   569.01    556.1     703.4
## Measure10    838.5   719.74    703.4     889.8
##
## Multivariate Tests: Measure
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            1    0.4752    13.76     10    152 <2e-16
## Wilks             1    0.5248    13.76     10    152 <2e-16
## Hotelling-Lawley  1    0.9053    13.76     10    152 <2e-16
## Roy               1    0.9053    13.76     10    152 <2e-16
##
## ------------------------------------------
##
## Term: Profiles$AgeMate:Measure
##
##  Response transformation matrix:
##          Measure1 Measure2 Measure3 Measure4 Measure5 Measure6
## info            1        0        0        0        0        0
## comp            0        1        0        0        0        0
## arith           0        0        1        0        0        0
## simil           0        0        0        1        0        0
## vocab           0        0        0        0        1        0
## digit           0        0        0        0        0        1
## pictcomp        0        0        0        0        0        0
## parang          0        0        0        0        0        0
## block           0        0        0        0        0        0
## object          0        0        0        0        0        0
## coding         -1       -1       -1       -1       -1       -1
##          Measure7 Measure8 Measure9 Measure10
## info            0        0        0         0
## comp            0        0        0         0
## arith           0        0        0         0
## simil           0        0        0         0
## vocab           0        0        0         0
## digit           0        0        0         0
## pictcomp        1        0        0         0
## parang          0        1        0         0
## block           0        0        1         0
## object          0        0        0         1
## coding         -1       -1       -1        -1
##
## Sum of squares and products for the hypothesis:
##           Measure1 Measure2 Measure3 Measure4 Measure5 Measure6
## Measure1    24.139   23.190   -7.691   33.422   23.154   11.636
## Measure2    23.190   30.199    6.608   39.060   17.761   17.947
## Measure3    -7.691    6.608   27.189    1.637  -15.301    8.254
## Measure4    33.422   39.060    1.637   52.376   28.123   22.051
```

```
## Measure5    23.154   17.761  -15.301   28.123   24.748    7.330
## Measure6    11.636   17.947    8.254   22.051    7.330   11.392
## Measure7   -25.298    3.311   56.867  -10.788  -39.898   11.403
## Measure8    14.832   20.823    6.892   26.306   10.506   12.767
## Measure9    10.930   27.051   25.768   29.660    1.116   19.412
## Measure10   14.474   28.091   20.460   32.492    5.853   19.100
##           Measure7 Measure8 Measure9 Measure10
## Measure1   -25.298   14.832   10.930    14.474
## Measure2     3.311   20.823   27.051    28.091
## Measure3    56.867    6.892   25.768    20.460
## Measure4   -10.788   26.306   29.660    32.492
## Measure5   -39.898   10.506    1.116     5.853
## Measure6    11.403   12.767   19.412    19.100
## Measure7   122.802    7.376   46.253    34.296
## Measure8     7.376   14.570   20.453    20.668
## Measure9    46.253   20.453   39.535    36.199
## Measure10   34.296   20.668   36.199    34.089
##
## Multivariate Tests: Profiles$AgeMate:Measure
##                  Df test stat approx F num Df den Df  Pr(>F)
## Pillai            2    0.2224    1.915     20    306 0.01131
## Wilks             2    0.7840    1.967     20    304 0.00868
## Hotelling-Lawley  2    0.2674    2.019     20    302 0.00666
## Roy               2    0.2321    3.551     10    153 0.00030
##
## Univariate Type II Repeated-Measures ANOVA Assuming Sphericity
##
##                          Sum Sq num Df Error SS den Df F value
## (Intercept)              179322      1     4916    161 5872.55
## Profiles$AgeMate             50      2     4916    161    0.81
## Measure                    1190     10     9408   1610   20.37
## Profiles$AgeMate:Measure    218     20     9408   1610    1.87
##                          Pr(>F)
## (Intercept)              <2e-16
## Profiles$AgeMate          0.446
## Measure                  <2e-16
## Profiles$AgeMate:Measure  0.011
##
##
## Mauchly Tests for Sphericity
##
##                          Test statistic  p-value
## Measure                           0.281 1.32e-18
## Profiles$AgeMate:Measure          0.281 1.32e-18
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
##  for Departure from Sphericity
##
##                          GG eps Pr(>F[GG])
## Measure                   0.771     <2e-16
## Profiles$AgeMate:Measure  0.771      0.021
##
##                          HF eps Pr(>F[HF])
```

```
## Measure                  0.8135  1.130e-29
## Profiles$AgeMate:Measure 0.8135  1.887e-02

Profile_Means <- aggregate(cbind(info, comp, arith, simil, vocab,
    digit, pictcomp, parang, block, object, coding) ~ AgeMate, Profiles,
    mean)

plot_data <- rbind(t(Profile_Means[1, 2:12]), t(Profile_Means[2, 2:12]),
    t(Profile_Means[3, 2:12]))
plot_data <- as.data.frame(plot_data)
names(plot_data) <- c("values")
plot_data$group <- c(rep("Older", 11), rep("Same", 11), rep("Younger",
    11))
plot_data$scale <- c("Information", "Comprehension", "Arithmetic",
    "Similarities", "Vocabulary", "Digit Span", "Picture Completion",
    "Picture Arrangement", "Block Design", "Object Assembly", "Coding",
    "Information", "Comprehension", "Arithmetic", "Similarities",
    "Vocabulary", "Digit Span", "Picture Completion", "Picture Arrangement",
    "Block Design", "Object Assembly", "Coding", "Information", "Comprehension",
    "Arithmetic", "Similarities", "Vocabulary", "Digit Span", "Picture Completion",
    "Picture Arrangement", "Block Design", "Object Assembly", "Coding")
plot_data$scale_n <- c(rep(seq(1, 11, 1), 3))
```

```
ggplot(plot_data, aes(x = scale_n, y = values, color = as.factor(group))) +
    geom_line(size = 1) + geom_point(size = 2) + scale_color_manual(values = c("red",
    "blue", "green4")) + coord_cartesian(xlim = c(1, 11), ylim = c(8,
    12)) + scale_y_continuous(breaks = seq(8, 12, 1)) + scale_x_continuous(breaks = seq(1,
    11, 1), labels = c("Information", "Comprehension", "Arithmetic",
    "Similarities", "Vocabulary", "Digit Span", "Picture Completion",
    "Picture Arrangement", "Block Design", "Object Assembly", "Coding")) +
    xlab("Scale") + ylab("Scale Mean") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 10, face = "bold", angle = 45, hjust = 1), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Scale Means as a Function of Group")
```



8 Doubly Multivariate Designs

When variables have different metrics (not commensurate), but are measured repeatedly, a doubly multivariate analysis is used. This approach can be thought of as a multivariate analysis of transformed scores (sums and differences).

In this study, 38 healthy young men and 37 age-matched psychiatric male in-patients were asked to engage in brief 10-minute conversations with two other people (the targets, actually research assistants blind to the study purpose or participant status). Participants were given some brief background information about the targets before meeting them. Both targets were described as holding steady jobs, having hobbies, and going to school part time.

One target (A) was described as having had a lifetime problem with seasonal allergies. The other target (B) was described has having been hospitalized in the past for a psychiatric problem. During each interview, the distance the participant sat from the other person (in cm) and the amount of eye contact (in seconds) were assessed. At the end of the 10-minute conversation, participants were asked to rate their liking for the target (on a 7-point scale).

The researchers hypothesized that all participants would distance themselves more from targets believed to have had a psychiatric problem and that they would like this target less than the target with no apparent history of psychiatric problems. Eye contact, however, was expected to show a different pattern. Participants were expected to engage in more eye contact with targets who were different from them. Healthy participants were expected to have more eye contact with targets thought to have psychiatric problems than with targets believed to be healthy. Psychiatric patients were expected to show the opposite pattern.

```
# Create a matrix that represents the sums and differences of the
# measures. The first three columns in the following matrix create
# sums of the distance, liking, and eye contact variables,
# collapsing over the two targets. The last three columns create
# difference scores comparing the responses to each target,
# separately for each measure (distance, liking, eye contact).
imatrix <- matrix(c(1, 0, 0, 1, 0, 0, 1, 0, 0, -1, 0, 0, 0, 1, 0,
    0, 1, 0, 0, 1, 0, 0, -1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, -1),
    nrow = 6, ncol = 6, byrow = TRUE)
colnames(imatrix) <- c("Distance_Sum", "Liking_Sum", "Eye_Contant_Sum",
    "Distance_Diff", "Liking_Diff", "Eye_Contanct_Diff")
rownames(imatrix) <- colnames(Double)[-c(7:9)]
(imatrix <- list(measure = imatrix[, 1:3], target = imatrix[, 4:6]))

## $measure
##                          Distance_Sum Liking_Sum Eye_Contant_Sum
## Interpersonal_Distance_A            1          0               0
## Interpersonal_Distance_B            1          0               0
## Liking_A                            0          1               0
## Liking_B                            0          1               0
## Eye_Contact_A                       0          0               1
## Eye_Contact_B                       0          0               1
##
## $target
##                          Distance_Diff Liking_Diff
## Interpersonal_Distance_A             1           0
## Interpersonal_Distance_B            -1           0
## Liking_A                             0           1
## Liking_B                             0          -1
## Eye_Contact_A                        0           0
## Eye_Contact_B                        0           0
##                          Eye_Contanct_Diff
## Interpersonal_Distance_A                 0
## Interpersonal_Distance_B                 0
## Liking_A                                 0
## Liking_B                                 0
## Eye_Contact_A                            1
## Eye_Contact_B                           -1
```

```r
# Contrast the two groups.
contrasts(Double$Group) <- matrix(c(1, -1), ncol = 1)
# Fit each measure in a linear model.
Double_Fit <- lm(cbind(Interpersonal_Distance_A, Interpersonal_Distance_B,
    Liking_A, Liking_B, Eye_Contact_A, Eye_Contact_B) ~ Group, data = Double)
# Get the doubly multivariate results.
Anova(Double_Fit, imatrix = imatrix, test = "Wilks")

##
## Type II Repeated Measures MANOVA Tests: Wilks test statistic
##               Df test stat approx F num Df den Df  Pr(>F)
## measure        1     0.012     1939      3     71 < 2e-16
## Group:measure  1     0.899        3      3     71   0.056
## target         1     0.362       42      3     71 1.2e-15
## Group:target   1     0.262       67      3     71 < 2e-16

summary.aov(Double_Fit)

##  Response Interpersonal_Distance_A :
##             Df Sum Sq Mean Sq F value Pr(>F)
## Group        1  15348   15348    4.56  0.036
## Residuals   73 245790    3367
##
##  Response Interpersonal_Distance_B :
##             Df Sum Sq Mean Sq F value Pr(>F)
## Group        1   1625    1625     0.5   0.48
## Residuals   73 236967    3246
##
##  Response Liking_A :
##             Df Sum Sq Mean Sq F value Pr(>F)
## Group        1    6.1    6.11    3.49  0.066
## Residuals   73  128.1    1.75
##
##  Response Liking_B :
##             Df Sum Sq Mean Sq F value Pr(>F)
## Group        1    0.5   0.459    0.21   0.65
## Residuals   73  158.2   2.167
##
##  Response Eye_Contact_A :
##             Df Sum Sq Mean Sq F value     Pr(>F)
## Group        1   6372    6372    32.4 0.00000025
## Residuals   73  14374     197
##
##  Response Eye_Contact_B :
##             Df Sum Sq Mean Sq F value     Pr(>F)
## Group        1   7787    7787    28.9 0.00000087
## Residuals   73  19654     269

# Get the separate repeated measures ANOVAs for each measure.
Measure <- factor(c("Target_A", "Target_B"), levels = c("Target_A",
    "Target_B"))
idata <- data.frame(Measure)

LM_2 <- lm(cbind(Interpersonal_Distance_A, Interpersonal_Distance_B) ~
    Group2, data = Double)
```

```r
Repeat_2 <- Anova(LM_2, idata = idata, idesign = ~Measure, type = 2)
summary(Repeat_2, multivariate = TRUE)
```

```
##
## Type II Repeated Measures MANOVA Tests:
##
## ------------------------------------------
##
## Term: (Intercept)
##
##  Response transformation matrix:
##                          (Intercept)
## Interpersonal_Distance_A           1
## Interpersonal_Distance_B           1
##
## Sum of squares and products for the hypothesis:
##             (Intercept)
## (Intercept)     6798687
##
## Multivariate Tests: (Intercept)
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            1     0.905    695.5      1     73 <2e-16
## Wilks             1     0.095    695.5      1     73 <2e-16
## Hotelling-Lawley  1     9.527    695.5      1     73 <2e-16
## Roy               1     9.527    695.5      1     73 <2e-16
##
## ------------------------------------------
##
## Term: Group2
##
##  Response transformation matrix:
##                          (Intercept)
## Interpersonal_Distance_A           1
## Interpersonal_Distance_B           1
##
## Sum of squares and products for the hypothesis:
##             (Intercept)
## (Intercept)       26963
##
## Multivariate Tests: Group2
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            1    0.0364    2.758      1     73  0.101
## Wilks             1    0.9636    2.758      1     73  0.101
## Hotelling-Lawley  1    0.0378    2.758      1     73  0.101
## Roy               1    0.0378    2.758      1     73  0.101
##
## ------------------------------------------
##
## Term: Measure
##
##  Response transformation matrix:
##                          Measure1
## Interpersonal_Distance_A        1
## Interpersonal_Distance_B       -1
```

```
##
## Sum of squares and products for the hypothesis:
##          Measure1
## Measure1     1643
##
## Multivariate Tests: Measure
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            1    0.0065    0.476      1     73  0.492
## Wilks             1    0.9935    0.476      1     73  0.492
## Hotelling-Lawley  1    0.0065    0.476      1     73  0.492
## Roy               1    0.0065    0.476      1     73  0.492
##
## ------------------------------------------
##
## Term: Group2:Measure
##
##  Response transformation matrix:
##                          Measure1
## Interpersonal_Distance_A        1
## Interpersonal_Distance_B       -1
##
## Sum of squares and products for the hypothesis:
##          Measure1
## Measure1     6984
##
## Multivariate Tests: Group2:Measure
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            1    0.0270    2.024      1     73  0.159
## Wilks             1    0.9730    2.024      1     73  0.159
## Hotelling-Lawley  1    0.0277    2.024      1     73  0.159
## Roy               1    0.0277    2.024      1     73  0.159
##
## Univariate Type II Repeated-Measures ANOVA Assuming Sphericity
##
##                 Sum Sq num Df Error SS den Df F value Pr(>F)
## (Intercept)    3399344      1   356798     73  695.50 <2e-16
## Group2           13481      1   356798     73    2.76   0.10
## Measure            821      1   125958     73    0.48   0.49
## Group2:Measure    3492      1   125958     73    2.02   0.16
```

```
LM_3 <- lm(cbind(Liking_A, Liking_B) ~ Group2, data = Double)
Repeat_3 <- Anova(LM_3, idata = idata, idesign = ~Measure, type = 2)
summary(Repeat_3, multivariate = TRUE)
```

```
##
## Type II Repeated Measures MANOVA Tests:
##
## ------------------------------------------
##
## Term: (Intercept)
##
##  Response transformation matrix:
##          (Intercept)
## Liking_A           1
```

```
## Liking_B           1
##
## Sum of squares and products for the hypothesis:
##             (Intercept)
## (Intercept)        7086
##
## Multivariate Tests: (Intercept)
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            1     0.938     1102      1     73 <2e-16
## Wilks             1     0.062     1102      1     73 <2e-16
## Hotelling-Lawley  1    15.102     1102      1     73 <2e-16
## Roy               1    15.102     1102      1     73 <2e-16
##
## ------------------------------------------
##
## Term: Group2
##
##  Response transformation matrix:
##          (Intercept)
## Liking_A           1
## Liking_B           1
##
## Sum of squares and products for the hypothesis:
##             (Intercept)
## (Intercept)       9.924
##
## Multivariate Tests: Group2
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            1    0.0207    1.544      1     73  0.218
## Wilks             1    0.9793    1.544      1     73  0.218
## Hotelling-Lawley  1    0.0212    1.544      1     73  0.218
## Roy               1    0.0212    1.544      1     73  0.218
##
## ------------------------------------------
##
## Term: Measure
##
##  Response transformation matrix:
##          Measure1
## Liking_A        1
## Liking_B       -1
##
## Sum of squares and products for the hypothesis:
##          Measure1
## Measure1    158.4
##
## Multivariate Tests: Measure
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            1    0.6051    111.9      1     73 <2e-16
## Wilks             1    0.3949    111.9      1     73 <2e-16
## Hotelling-Lawley  1    1.5326    111.9      1     73 <2e-16
## Roy               1    1.5326    111.9      1     73 <2e-16
##
## ------------------------------------------
```

```
##
## Term: Group2:Measure
##
##  Response transformation matrix:
##          Measure1
## Liking_A        1
## Liking_B       -1
##
## Sum of squares and products for the hypothesis:
##          Measure1
## Measure1    3.223
##
## Multivariate Tests: Group2:Measure
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            1    0.0302    2.276      1     73  0.136
## Wilks             1    0.9698    2.276      1     73  0.136
## Hotelling-Lawley  1    0.0312    2.276      1     73  0.136
## Roy               1    0.0312    2.276      1     73  0.136
##
## Univariate Type II Repeated-Measures ANOVA Assuming Sphericity
##
##                Sum Sq num Df Error SS den Df F value Pr(>F)
## (Intercept)      3543      1    234.6     73 1102.46 <2e-16
## Group2              5      1    234.6     73    1.54   0.22
## Measure            79      1     51.7     73  111.88 <2e-16
## Group2:Measure      2      1     51.7     73    2.28   0.14
```

```
LM_4 <- lm(cbind(Eye_Contact_A, Eye_Contact_B) ~ Group2, data = Double)
Repeat_4 <- Anova(LM_4, idata = idata, idesign = ~Measure, type = 2)
summary(Repeat_4, multivariate = TRUE)
```

```
##
## Type II Repeated Measures MANOVA Tests:
##
## ------------------------------------------
##
## Term: (Intercept)
##
##  Response transformation matrix:
##               (Intercept)
## Eye_Contact_A           1
## Eye_Contact_B           1
##
## Sum of squares and products for the hypothesis:
##             (Intercept)
## (Intercept)     1059934
##
## Multivariate Tests: (Intercept)
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            1      0.95     1391      1     73 <2e-16
## Wilks             1      0.05     1391      1     73 <2e-16
## Hotelling-Lawley  1     19.06     1391      1     73 <2e-16
## Roy               1     19.06     1391      1     73 <2e-16
##
```

```
## ------------------------------------------
##
## Term: Group2
##
##  Response transformation matrix:
##               (Intercept)
## Eye_Contact_A           1
## Eye_Contact_B           1
##
## Sum of squares and products for the hypothesis:
##             (Intercept)
## (Intercept)       70.83
##
## Multivariate Tests: Group2
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            1    0.0013  0.09296      1     73  0.761
## Wilks             1    0.9987  0.09296      1     73  0.761
## Hotelling-Lawley  1    0.0013  0.09296      1     73  0.761
## Roy               1    0.0013  0.09296      1     73  0.761
##
## ------------------------------------------
##
## Term: Measure
##
##  Response transformation matrix:
##               Measure1
## Eye_Contact_A        1
## Eye_Contact_B       -1
##
## Sum of squares and products for the hypothesis:
##          Measure1
## Measure1    28.21
##
## Multivariate Tests: Measure
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            1    0.0023   0.1656      1     73  0.685
## Wilks             1    0.9977   0.1656      1     73  0.685
## Hotelling-Lawley  1    0.0023   0.1656      1     73  0.685
## Roy               1    0.0023   0.1656      1     73  0.685
##
## ------------------------------------------
##
## Term: Group2:Measure
##
##  Response transformation matrix:
##               Measure1
## Eye_Contact_A        1
## Eye_Contact_B       -1
##
## Sum of squares and products for the hypothesis:
##          Measure1
## Measure1    28247
##
## Multivariate Tests: Group2:Measure
```

```
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            1    0.6943    165.8      1     73 <2e-16
## Wilks             1    0.3057    165.8      1     73 <2e-16
## Hotelling-Lawley  1    2.2716    165.8      1     73 <2e-16
## Roy               1    2.2716    165.8      1     73 <2e-16
##
## Univariate Type II Repeated-Measures ANOVA Assuming Sphericity
##
##                Sum Sq num Df Error SS den Df F value Pr(>F)
## (Intercept)    529967      1    27811     73 1391.11 <2e-16
## Group2             35      1    27811     73    0.09   0.76
## Measure            14      1     6217     73    0.17   0.69
## Group2:Measure  14124      1     6217     73  165.83 <2e-16
```

```
# Display the means.
describeBy(Double[, 1:6], group = Double$Group2, digits = 2)
```

```
##
##  Descriptive statistics by group
## group: Healthy Controls
##                         vars  n   mean    sd median trimmed
## Interpersonal_Distance_A    1 37 133.70 62.57    150  137.81
## Interpersonal_Distance_B    2 37 148.16 57.09    166  154.61
## Liking_A                    3 37   5.30  1.39      6    5.45
## Liking_B                    4 37   4.05  1.56      4    4.13
## Eye_Contact_A               5 37  50.41 13.37     53   51.06
## Eye_Contact_B               6 37  69.46 15.96     70   69.61
##                           mad min max range  skew kurtosis
## Interpersonal_Distance_A 74.13  25 200   175 -0.36    -1.35
## Interpersonal_Distance_B 50.41  25 200   175 -0.76    -0.77
## Liking_A                  1.48   1   7     6 -1.06     1.02
## Liking_B                  1.48   1   7     6 -0.17    -0.73
## Eye_Contact_A            10.38  12  77    65 -0.64     0.29
## Eye_Contact_B            11.86  34  99    65 -0.14    -0.48
##                            se
## Interpersonal_Distance_A 10.29
## Interpersonal_Distance_B  9.39
## Liking_A                  0.23
## Liking_B                  0.26
## Eye_Contact_A             2.20
## Eye_Contact_B             2.62
## --------------------------------------------------------
## group: Patients
##                         vars  n   mean    sd median trimmed
## Interpersonal_Distance_A    1 38 162.32 53.23  199.0  170.16
## Interpersonal_Distance_B    2 38 157.47 56.86  189.0  165.41
## Liking_A                    3 38   5.87  1.26    6.0    6.00
## Liking_B                    4 38   4.21  1.38    4.0    4.22
## Eye_Contact_A               5 38  68.84 14.65   70.5   69.69
## Eye_Contact_B               6 38  49.08 16.84   49.5   50.22
##                           mad min max range  skew kurtosis   se
## Interpersonal_Distance_A  1.48  25 200   175 -1.16     0.00 8.64
## Interpersonal_Distance_B 16.31  25 200   175 -1.10    -0.22 9.22
## Liking_A                  1.48   3   7     4 -0.80    -0.61 0.20
```

```
## Liking_B                  1.48   1   7     6 -0.01    -0.33 0.22
## Eye_Contact_A            13.34  16  94    78 -1.03     2.50 2.38
## Eye_Contact_B            11.86   0  78    78 -0.86     1.31 2.73
```

```
# Create the sums and differences for entry in discriminant
# analysis to produce additional results.
Double$Distance_Sum <- 0.7071 * Double$Interpersonal_Distance_A +
    0.7071 * Double$Interpersonal_Distance_B
Double$Liking_Sum <- 0.7071 * Double$Liking_A + 0.7071 * Double$Liking_B
Double$Eye_Contact_Sum <- 0.7071 * Double$Eye_Contact_A + 0.7071 *
    Double$Eye_Contact_B
Double$Distance_Diff <- 0.7071 * Double$Interpersonal_Distance_A -
    0.7071 * Double$Interpersonal_Distance_B
Double$Liking_Diff <- 0.7071 * Double$Liking_A - 0.7071 * Double$Liking_B
Double$Eye_Contact_Diff <- 0.7071 * Double$Eye_Contact_A - 0.7071 *
    Double$Eye_Contact_B

LM_Sum <- lm(cbind(Distance_Sum, Liking_Sum, Eye_Contact_Sum) ~ Group2,
    data = Double)
LDA_Sum <- candisc(LM_Sum, data = Double)
LDA_Sum
```

```
##
## Canonical Discriminant Analysis for Group2:
##
##   CanRsq Eigenvalue Difference Percent Cumulative
## 1  0.101      0.112                100        100
##
## Test of H0: The canonical correlations in the
## current row and all that follow are zero
##
##   LR test stat approx F numDF denDF Pr(> F)
## 1        0.899     2.64     3    71   0.056
```

```
summary(LDA_Sum)
```

```
##
## Canonical Discriminant Analysis for Group2:
##
##   CanRsq Eigenvalue Difference Percent Cumulative
## 1 0.1005     0.1118                100        100
##
## Class means:
##
## [1] -0.3342  0.3254
##
##  std coefficients:
##    Distance_Sum      Liking_Sum Eye_Contact_Sum
##         1.03266         0.92210         0.01531
```

```
LDA_Sum$coeffs.std
```

```
##                     Can1
## Distance_Sum    1.03266
## Liking_Sum      0.92210
## Eye_Contact_Sum 0.01531
```

```
LDA_Sum$structure

##                     Can1
## Distance_Sum     0.6018
## Liking_Sum       0.4540
## Eye_Contact_Sum -0.1125

LM_Diff <- lm(cbind(Distance_Diff, Liking_Diff, Eye_Contact_Diff) ~
    Group2, data = Double)
LDA_Diff <- candisc(LM_Diff, data = Double)
LDA_Diff

##
## Canonical Discriminant Analysis for Group2:
##
##   CanRsq Eigenvalue Difference Percent Cumulative
## 1  0.738       2.82                100        100
##
## Test of H0: The canonical correlations in the
## current row and all that follow are zero
##
##   LR test stat approx F numDF denDF Pr(> F)
## 1        0.262     66.8     3    71  <2e-16

summary(LDA_Diff)

##
## Canonical Discriminant Analysis for Group2:
##
##   CanRsq Eigenvalue Difference Percent Cumulative
## 1 0.7383      2.821                100        100
##
## Class means:
##
## [1] -1.679  1.635
##
##  std coefficients:
##    Distance_Diff      Liking_Diff Eye_Contact_Diff
##           0.3898          -0.1990           1.0946

LDA_Diff$coeffs.std

##                      Can1
## Distance_Diff     0.3898
## Liking_Diff      -0.1990
## Eye_Contact_Diff  1.0946

LDA_Diff$structure

##                     Can1
## Distance_Diff    0.1912
## Liking_Diff      0.2024
## Eye_Contact_Diff 0.9698

MANOVA_5 <- manova(cbind(Distance_Sum, Liking_Sum, Eye_Contact_Sum) ~
    Group2, data = Double)
cor(MANOVA_5$residuals)
```

```
##                  Distance_Sum Liking_Sum Eye_Contact_Sum
## Distance_Sum          1.0000    -0.4795         -0.5912
## Liking_Sum           -0.4795     1.0000          0.5297
## Eye_Contact_Sum      -0.5912     0.5297          1.0000
```

```
MANOVA_6 <- manova(cbind(Distance_Diff, Liking_Diff, Eye_Contact_Diff) ~
    Group2, data = Double)
cor(MANOVA_6$residuals)
```

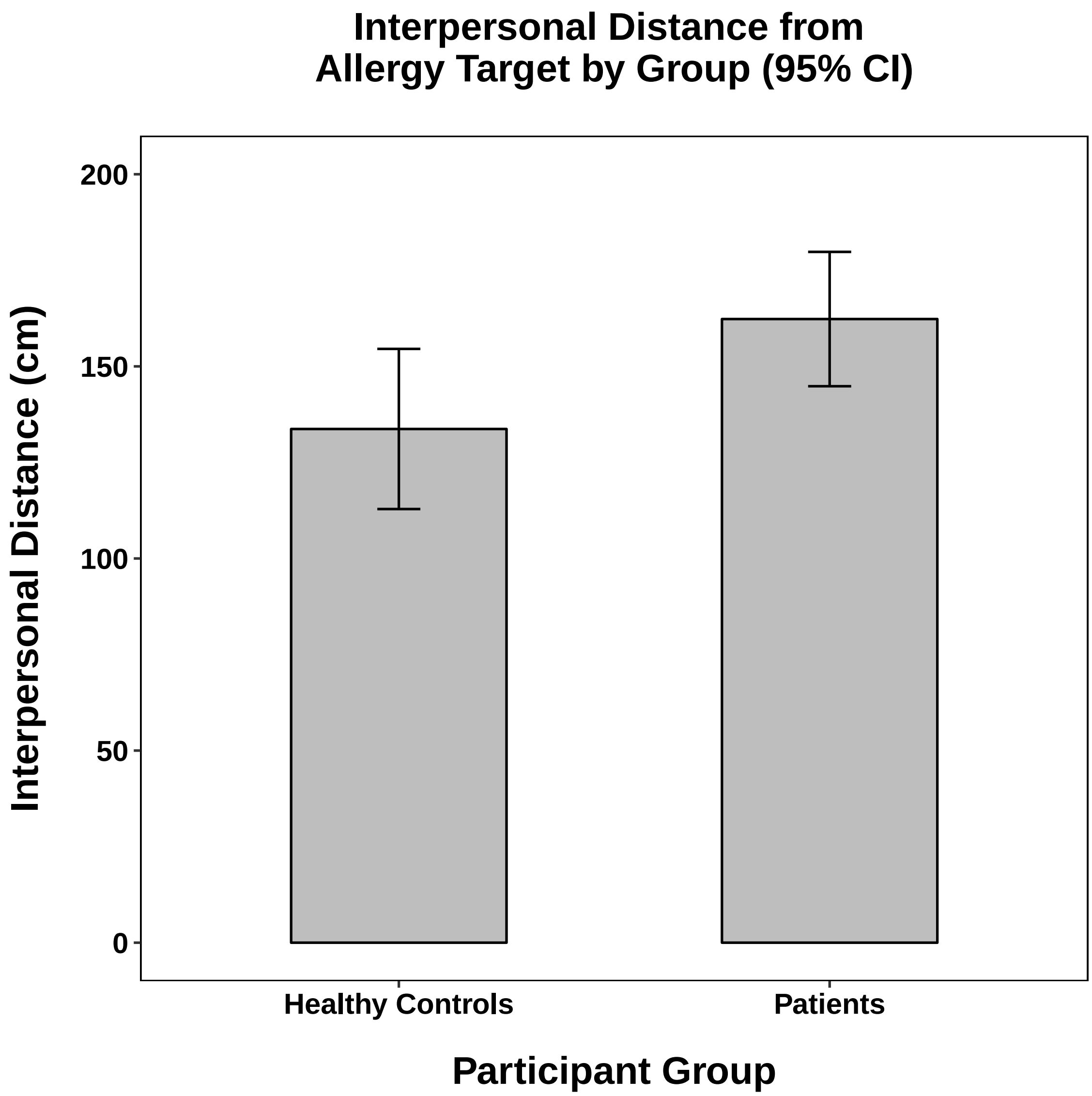
```
##                   Distance_Diff Liking_Diff Eye_Contact_Diff
## Distance_Diff           1.0000     -0.2710          -0.3148
## Liking_Diff            -0.2710      1.0000           0.3744
## Eye_Contact_Diff       -0.3148      0.3744           1.0000
```

9 Means and Confidence Intervals

Displayed here are bar graphs of the condition means with 95% confidence intervals.

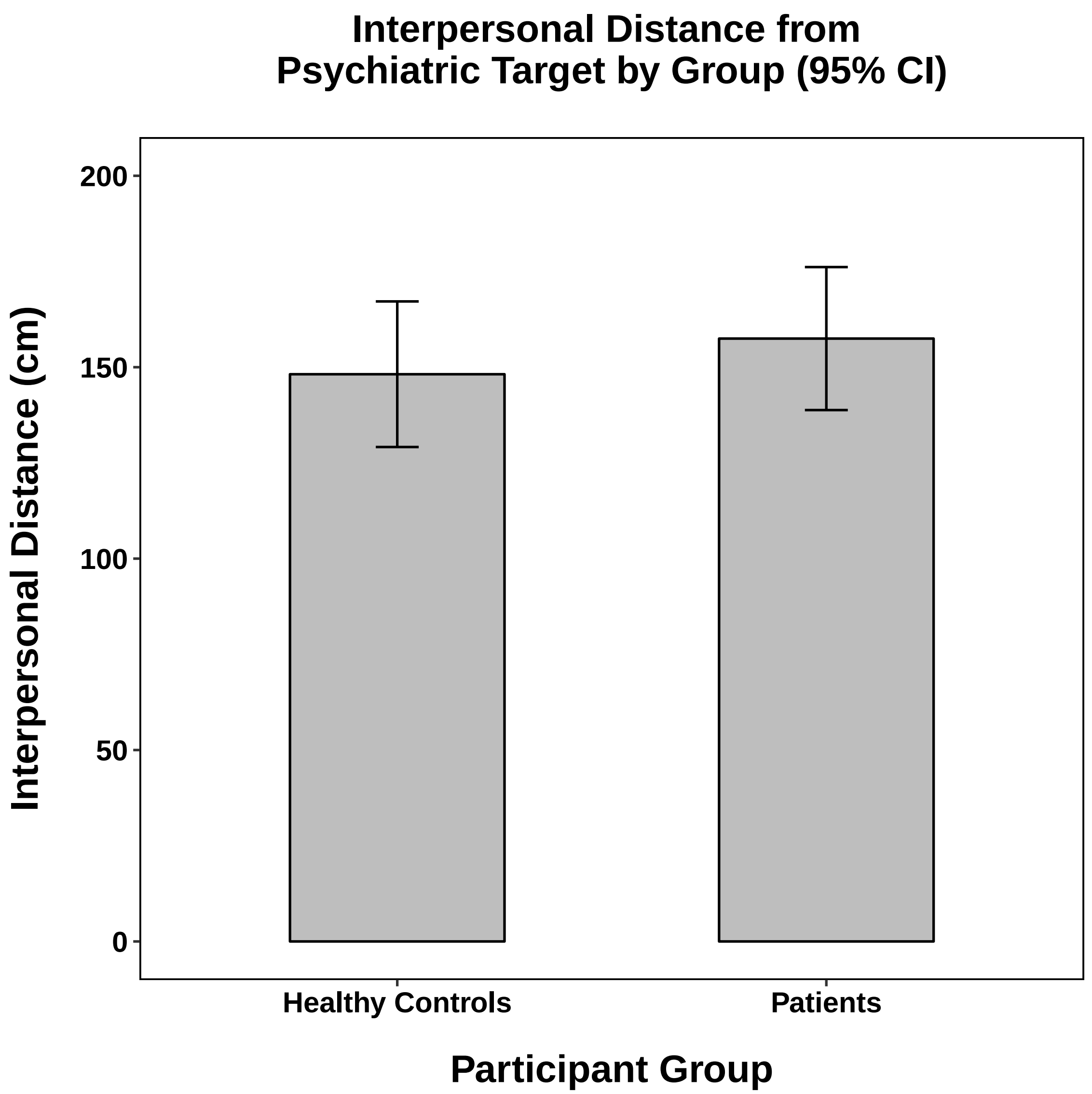
```
D <- describeBy(Double[, 1:6], group = Double$Group2)

plot_data <- matrix(NA, nrow = 2, ncol = 12)

for (i in 1:2) {
    for (j in 1:6) {
        plot_data[i, j] <- D[[i]]$mean[j]
        plot_data[i, j + 6] <- qt(0.975, D[[i]]$n[j]) * D[[i]]$sd[j]/sqrt(D[[i]]$n[j])
    }
}
plot_data <- as.data.frame(plot_data)
names(plot_data) <- c("ID_A_mean", "ID_B_mean", "Like_A_mean", "Like_B_mean",
    "Eye_A_mean", "Eye_B_mean", "ID_A_CI", "ID_B_CI", "Like_A_CI",
    "Like_B_CI", "Eye_A_CI", "Eye_B_CI")
plot_data$Group <- factor(c("Healthy Controls", "Patients"))
plot_data$Group_F <- factor(plot_data$Group, levels = c("Healthy Controls",
    "Patients"), labels = c("Healthy Controls", "Patients"))
```

```
p1 <- ggplot(plot_data, aes(x = as.factor(Group_F), y = ID_A_mean)) +
    geom_bar(position = position_dodge(), stat = "identity", color = "black",
        width = 0.5, fill = "grey") + geom_errorbar(aes(ymin = ID_A_mean -
    ID_A_CI, ymax = ID_A_mean + ID_A_CI), width = 0.1, position = position_dodge(0.5)) +
    scale_y_continuous(breaks = c(seq(0, 200, 50))) + coord_cartesian(ylim = c(0,
    200)) + xlab("Participant Group") + ylab("Interpersonal Distance (cm)") +
    theme(text = element_text(size = 14, family = "sans", color = "black",
        face = "bold"), axis.text.y = element_text(colour = "black",
        size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
        size = 12, face = "bold", angle = 0, hjust = 0.5), axis.title.x = element_text(margin = margin(1
        0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
        15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
        plot.title = element_text(size = 16, face = "bold", margin = margin(0,
            0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
            linetype = 1, color = "black"), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), panel.border = element_rect(fill = NA,
            size = 0.5), plot.background = element_rect(fill = "white"),
        plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
        legend.title = element_blank()) + ggtitle("Interpersonal Distance from \nAllergy Target by Group
print(p1)
```



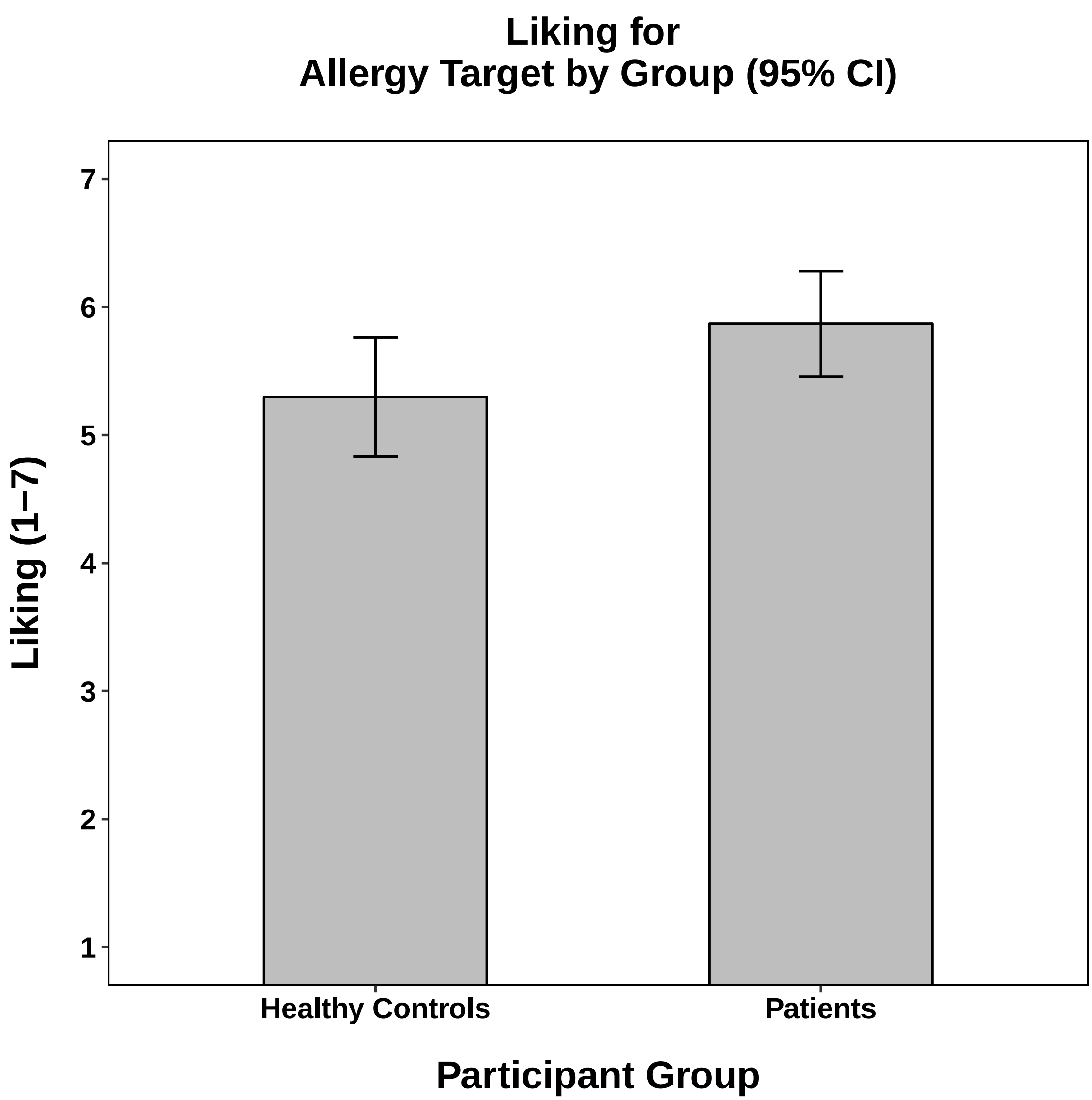

```
p2 <- ggplot(plot_data, aes(x = as.factor(Group_F), y = ID_B_mean)) +
    geom_bar(position = position_dodge(), stat = "identity", color = "black",
        width = 0.5, fill = "grey") + geom_errorbar(aes(ymin = ID_B_mean -
    ID_B_CI, ymax = ID_B_mean + ID_B_CI), width = 0.1, position = position_dodge(0.5)) +
    scale_y_continuous(breaks = c(seq(0, 200, 50))) + coord_cartesian(ylim = c(0,
    200)) + xlab("Participant Group") + ylab("Interpersonal Distance (cm)") +
    theme(text = element_text(size = 14, family = "sans", color = "black",
        face = "bold"), axis.text.y = element_text(colour = "black",
        size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
        size = 12, face = "bold", angle = 0, hjust = 0.5), axis.title.x = element_text(margin = margin(1
        0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
        15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
        plot.title = element_text(size = 16, face = "bold", margin = margin(0,
            0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
```

```
            linetype = 1, color = "black"), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), panel.border = element_rect(fill = NA,
            size = 0.5), plot.background = element_rect(fill = "white"),
        plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
        legend.title = element_blank()) + ggtitle("Interpersonal Distance from \nPsychiatric Target by (
print(p2)
```

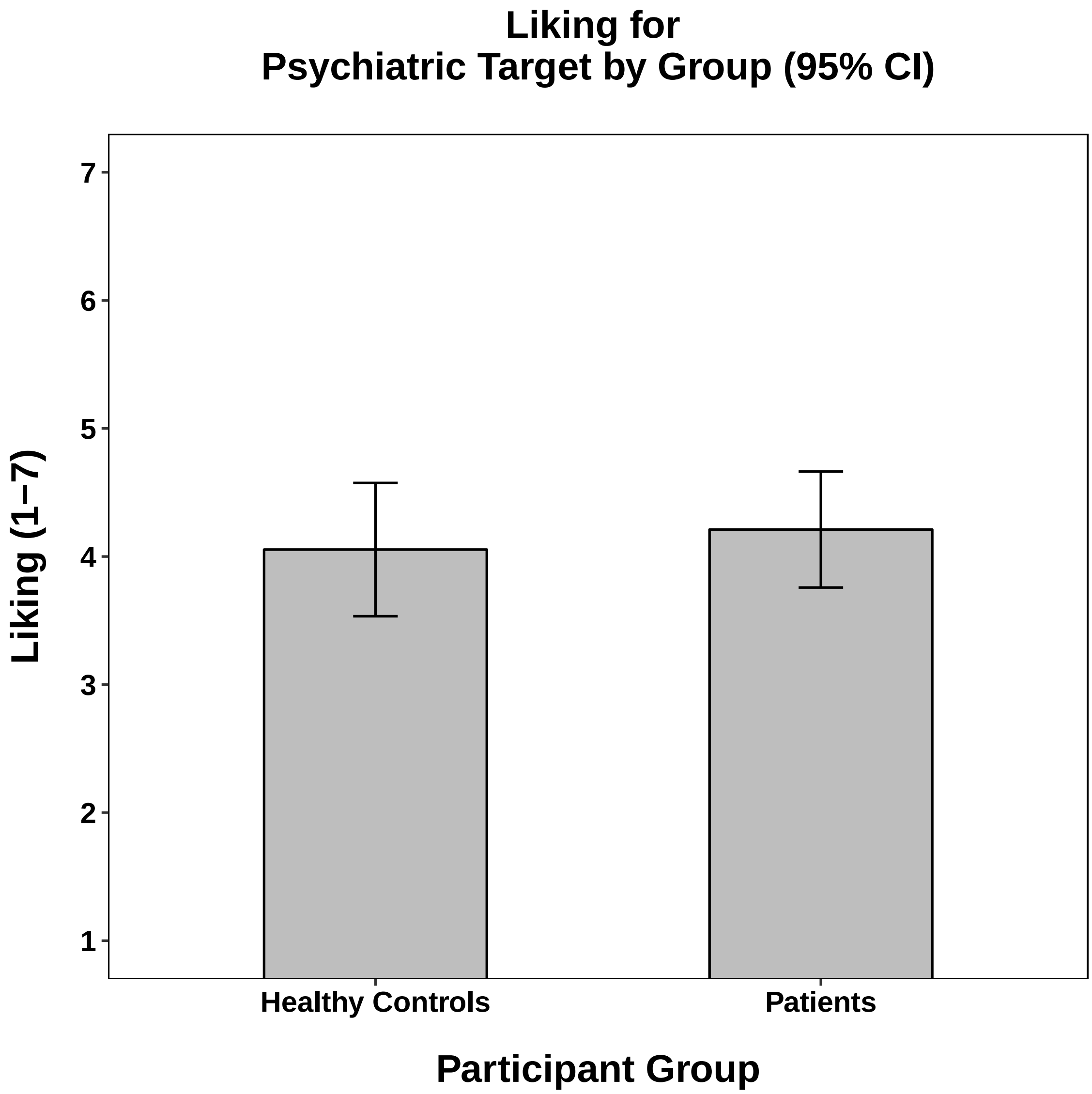



```
p3 <- ggplot(plot_data, aes(x = as.factor(Group_F), y = Like_A_mean)) +
    geom_bar(position = position_dodge(), stat = "identity", color = "black",
        width = 0.5, fill = "grey") + geom_errorbar(aes(ymin = Like_A_mean -
    Like_A_CI, ymax = Like_A_mean + Like_A_CI), width = 0.1, position = position_dodge(0.5)) +
    scale_y_continuous(breaks = c(seq(1, 7, 1))) + coord_cartesian(ylim = c(1,
    7)) + xlab("Participant Group") + ylab("Liking (1-7)") + theme(text = element_text(size = 14,
```

```
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0, hjust = 0.5), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.border = element_rect(fill = NA,
        size = 0.5), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Liking for \nAllergy Target by Group (95% CI)")
print(p3)
```

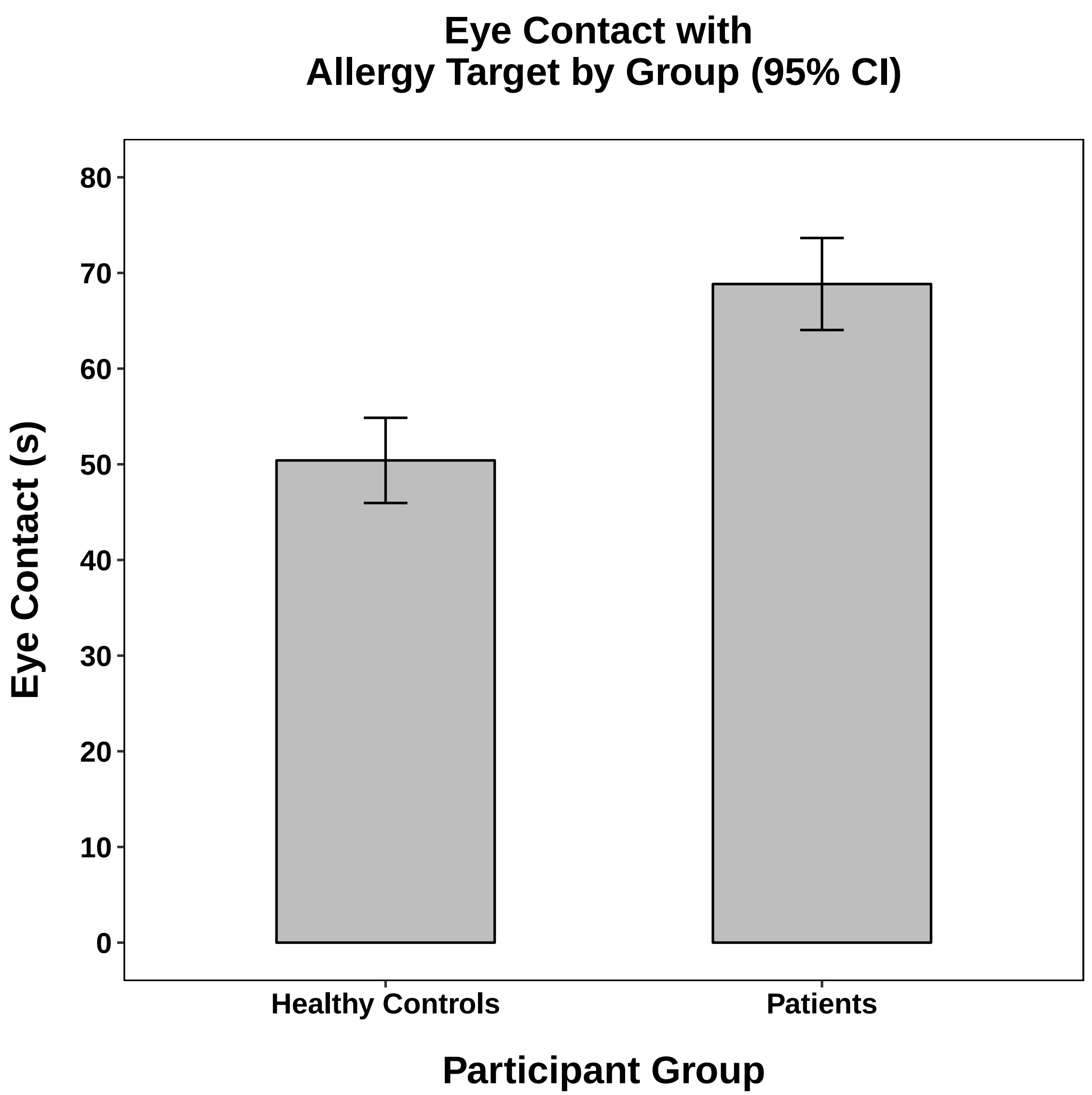


```
p4 <- ggplot(plot_data, aes(x = as.factor(Group_F), y = Like_B_mean)) +
    geom_bar(position = position_dodge(), stat = "identity", color = "black",
        width = 0.5, fill = "grey") + geom_errorbar(aes(ymin = Like_B_mean -
    Like_B_CI, ymax = Like_B_mean + Like_B_CI), width = 0.1, position = position_dodge(0.5)) +
    scale_y_continuous(breaks = c(seq(1, 7, 1))) + coord_cartesian(ylim = c(1,
    7)) + xlab("Participant Group") + ylab("Liking (1-7)") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0, hjust = 0.5), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.border = element_rect(fill = NA,
        size = 0.5), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Liking for \nPsychiatric Target by Group (95% CI)")
print(p4)
```



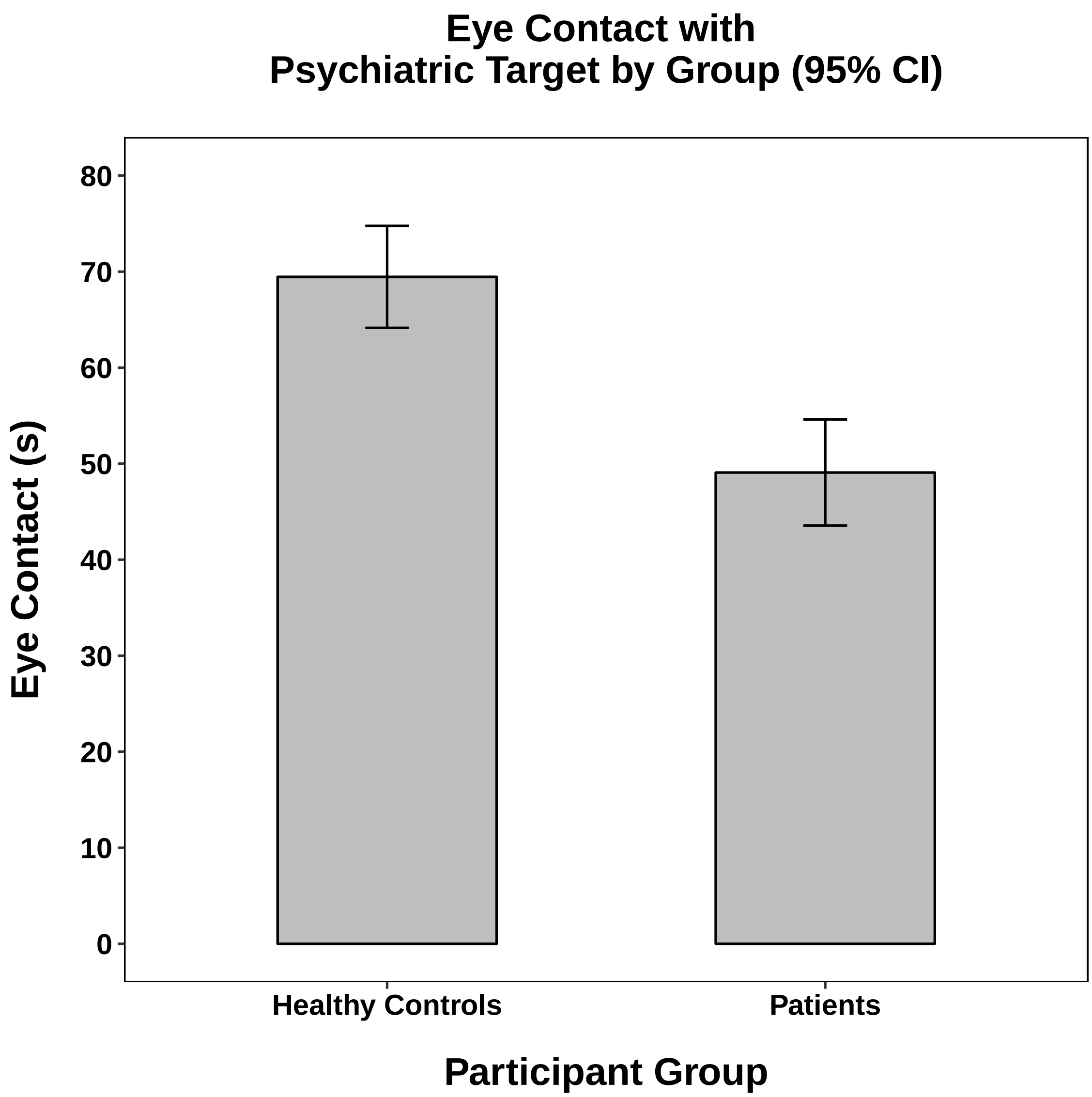

```
p5 <- ggplot(plot_data, aes(x = as.factor(Group_F), y = Eye_A_mean)) +
    geom_bar(position = position_dodge(), stat = "identity", color = "black",
        width = 0.5, fill = "grey") + geom_errorbar(aes(ymin = Eye_A_mean -
    Eye_A_CI, ymax = Eye_A_mean + Eye_A_CI), width = 0.1, position = position_dodge(0.5)) +
    scale_y_continuous(breaks = c(seq(0, 80, 10))) + coord_cartesian(ylim = c(0,
    80)) + xlab("Participant Group") + ylab("Eye Contact (s)") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0, hjust = 0.5), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
```

```
    panel.grid.minor = element_blank(), panel.border = element_rect(fill = NA,
        size = 0.5), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Eye Contact with \nAllergy Target by Group (95% CI)")
print(p5)
```




```
p6 <- ggplot(plot_data, aes(x = as.factor(Group_F), y = Eye_B_mean)) +
    geom_bar(position = position_dodge(), stat = "identity", color = "black",
        width = 0.5, fill = "grey") + geom_errorbar(aes(ymin = Eye_B_mean -
    Eye_B_CI, ymax = Eye_B_mean + Eye_B_CI), width = 0.1, position = position_dodge(0.5)) +
    scale_y_continuous(breaks = c(seq(0, 80, 10))) + coord_cartesian(ylim = c(0,
    80)) + xlab("Participant Group") + ylab("Eye Contact (s)") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
```

```
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0, hjust = 0.5), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.border = element_rect(fill = NA,
        size = 0.5), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Eye Contact with \nPsychiatric Target by Group (95% CI)")
print(p6)
```



```
p1 <- ggplot(plot_data, aes(x = as.factor(Group_F), y = ID_A_mean)) +
    geom_bar(position = position_dodge(), stat = "identity", color = "black",
        width = 0.5, fill = "grey") + geom_errorbar(aes(ymin = ID_A_mean -
    ID_A_CI, ymax = ID_A_mean + ID_A_CI), width = 0.1, position = position_dodge(0.5)) +
    scale_y_continuous(breaks = c(seq(0, 200, 50))) + coord_cartesian(ylim = c(0,
    200)) + xlab("") + ylab("Distance \n(cm)") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 8, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 8, face = "bold", angle = 0, hjust = 0.5), axis.title.x = element_blank(),
    axis.title.y = element_text(margin = margin(0, 10, 0, 0), size = 10),
    axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 10, face = "bold", margin = margin(0,
        0, 10, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.border = element_rect(fill = NA,
        size = 0.5), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(0.3, 0.3, 0.3, 0.3), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Allergy Target \nby Group (95% CI)")
```

```
p2 <- ggplot(plot_data, aes(x = as.factor(Group_F), y = ID_B_mean)) +
    geom_bar(position = position_dodge(), stat = "identity", color = "black",
        width = 0.5, fill = "grey") + geom_errorbar(aes(ymin = ID_B_mean -
    ID_B_CI, ymax = ID_B_mean + ID_B_CI), width = 0.1, position = position_dodge(0.5)) +
    scale_y_continuous(breaks = c(seq(0, 200, 50))) + coord_cartesian(ylim = c(0,
    200)) + xlab("") + ylab("") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 8, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 8, face = "bold", angle = 0, hjust = 0.5), axis.title.x = element_blank(),
    axis.title.y = element_text(margin = margin(0, 10, 0, 0), size = 10),
    axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 10, face = "bold", margin = margin(0,
        0, 10, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.border = element_rect(fill = NA,
        size = 0.5), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(0.3, 0.3, 0.3, 0.3), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Psychiatric Target \nby Group (95% CI)")
```

```
p3 <- ggplot(plot_data, aes(x = as.factor(Group_F), y = Like_A_mean)) +
    geom_bar(position = position_dodge(), stat = "identity", color = "black",
        width = 0.5, fill = "grey") + geom_errorbar(aes(ymin = Like_A_mean -
    Like_A_CI, ymax = Like_A_mean + Like_A_CI), width = 0.1, position = position_dodge(0.5)) +
    scale_y_continuous(breaks = c(seq(1, 7, 1))) + coord_cartesian(ylim = c(1,
    7)) + xlab("") + ylab("Liking \n(1-7)") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 8, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 8, face = "bold", angle = 0, hjust = 0.5), axis.title.x = element_blank(),
    axis.title.y = element_text(margin = margin(0, 10, 0, 0), size = 10),
    axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_blank(), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.border = element_rect(fill = NA,
```

```
        size = 0.5), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(0.3, 0.3, 0.3, 0.3), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Liking for \nAllergy Target by Group (95% CI)")
```

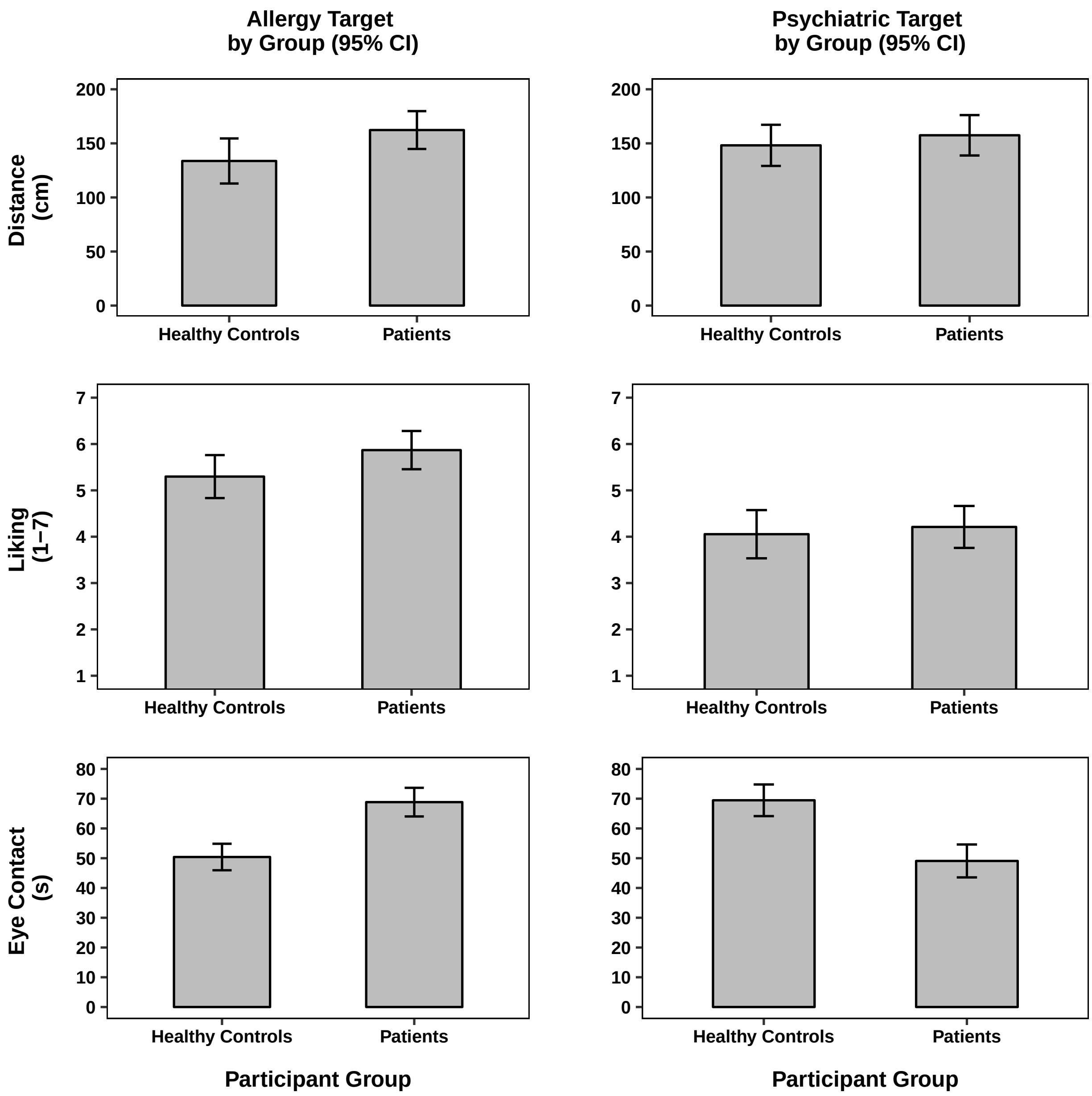
```
p4 <- ggplot(plot_data, aes(x = as.factor(Group_F), y = Like_B_mean)) +
    geom_bar(position = position_dodge(), stat = "identity", color = "black",
        width = 0.5, fill = "grey") + geom_errorbar(aes(ymin = Like_B_mean -
    Like_B_CI, ymax = Like_B_mean + Like_B_CI), width = 0.1, position = position_dodge(0.5)) +
    scale_y_continuous(breaks = c(seq(1, 7, 1))) + coord_cartesian(ylim = c(1,
    7)) + xlab("Participant Group") + ylab("") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 8, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 8, face = "bold", angle = 0, hjust = 0.5), axis.title.x = element_blank(),
    axis.title.y = element_text(margin = margin(0, 10, 0, 0), size = 10),
    axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_blank(), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.border = element_rect(fill = NA,
        size = 0.5), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(0.3, 0.3, 0.3, 0.3), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Liking for \nPsychiatric Target by Group (95% CI)")
```

```
p5 <- ggplot(plot_data, aes(x = as.factor(Group_F), y = Eye_A_mean)) +
    geom_bar(position = position_dodge(), stat = "identity", color = "black",
        width = 0.5, fill = "grey") + geom_errorbar(aes(ymin = Eye_A_mean -
    Eye_A_CI, ymax = Eye_A_mean + Eye_A_CI), width = 0.1, position = position_dodge(0.5)) +
    scale_y_continuous(breaks = c(seq(0, 80, 10))) + coord_cartesian(ylim = c(0,
    80)) + xlab("Participant Group") + ylab("Eye Contact \n(s)") +
    theme(text = element_text(size = 14, family = "sans", color = "black",
        face = "bold"), axis.text.y = element_text(colour = "black",
        size = 8, face = "bold"), axis.text.x = element_text(colour = "black",
        size = 8, face = "bold", angle = 0, hjust = 0.5), axis.title.x = element_text(margin = margin(1
        0, 0, 0), size = 10), axis.title.y = element_text(margin = margin(0,
        10, 0, 0), size = 10), axis.line.x = element_blank(), axis.line.y = element_blank(),
        plot.title = element_blank(), panel.background = element_rect(fill = "white",
            linetype = 1, color = "black"), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), panel.border = element_rect(fill = NA,
            size = 0.5), plot.background = element_rect(fill = "white"),
        plot.margin = unit(c(0.3, 0.3, 0.3, 0.3), "cm"), legend.position = "bottom",
        legend.title = element_blank()) + ggtitle("Eye Contact with \nAllergy Target by Group (95% CI)")
```

```
p6 <- ggplot(plot_data, aes(x = as.factor(Group_F), y = Eye_B_mean)) +
    geom_bar(position = position_dodge(), stat = "identity", color = "black",
        width = 0.5, fill = "grey") + geom_errorbar(aes(ymin = Eye_B_mean -
    Eye_B_CI, ymax = Eye_B_mean + Eye_B_CI), width = 0.1, position = position_dodge(0.5)) +
    scale_y_continuous(breaks = c(seq(0, 80, 10))) + coord_cartesian(ylim = c(0,
    80)) + xlab("Participant Group") + ylab("") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 8, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 8, face = "bold", angle = 0, hjust = 0.5), axis.title.x = element_text(margin = margin(10,
    0, 0, 0), size = 10), axis.title.y = element_text(margin = margin(0,
```

```r
    10, 0, 0), size = 10), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_blank(), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.border = element_rect(fill = NA,
        size = 0.5), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(0.3, 0.3, 0.3, 0.3), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Eye Contact with \nPsychiatric Target by Group (95% CI)")
```

```r
grid.arrange(p1, p2, p3, p4, p5, p6, nrow = 3)
```




```r
Sys.time() - how_long
```

```
## Time difference of 3.797 mins
```