

Advantage of two scans in small populations demonstrated with a transgender dataset.

Freya Acar

Introduction

We have a dataset where FreeSurfer was used to determine cortical and subcortical brain anatomy in cis- and transgender population. For every participant two scans were administered (T1 and T2) and an average of both scans was computed. In this report we analyze the data from one scan and the average of both scans. Later on we show the advantage and increase in power obtained by administering 2 scans. One participant (P22) was removed from the analysis because no anatomical data was available for this participant.

Actual data

We first read in the data. In data.all all available measurements are stored (descriptive and anatomical) while in data.hyp the anatomical results for every region of interest stored. Both files contain data of T1, T2 and the average.

```
# Read in data
data.all <- read.csv("../1.Data/Behzad_all.csv", sep=";", dec=",")
data.hyp <- read.csv("../1.Data/Behzad_hyp.csv", sep=";", dec=",")

# Check data
dim(data.hyp)
```

```
## [1] 140 80
```

```
dim(data.all)
```

```
## [1] 140 820
```

The regions we are interested in are the cerebellum, caudate, putamen, nucleus accumbens, thalamus, fusiform, pre-central gyrus, post-central gyrus, frontal poles and inferior parietal gyrus. Here we list the variables we selected from FreeSurfer that comply with these regions.

```
# Regions of interest
names(data.hyp[,55:80])

## [1] "Tavg_LeftCerebellumWhiteMatter"
## [2] "Tavg_LeftCerebellumCortex"
## [3] "Tavg_RightCerebellumWhiteMatter"
## [4] "Tavg_RightCerebellumCortex"
## [5] "Tavg_L_caudalanteriorcingulate_volume"
## [6] "Tavg_L_caudalmiddlefrontal_volume"
## [7] "Tavg_L_fusiform_volume"
## [8] "Tavg_L_inferiorparietal_volume"
## [9] "Tavg_L_postcentral_volume"
## [10] "Tavg_L_precentral_volume"
## [11] "Tavg_L_frontalpole_volume"
## [12] "Tavg_R_caudalanteriorcingulate_volume"
## [13] "Tavg_R_caudalmiddlefrontal_volume"
```

```
## [14] "Tavg_R_fusiform_volume"
## [15] "Tavg_R_inferiorparietal_volume"
## [16] "Tavg_R_postcentral_volume"
## [17] "Tavg_R_precentral_volume"
## [18] "Tavg_R_frontalpole_volume"
## [19] "Tavg_LeftThalamusProper"
## [20] "Tavg_LeftCaudate"
## [21] "Tavg_LeftPutamen"
## [22] "Tavg_LeftAccumbensarea"
## [23] "Tavg_RightThalamusProper"
## [24] "Tavg_RightCaudate"
## [25] "Tavg_RightPutamen"
## [26] "Tavg_RightAccumbensarea"
```

Descriptives

In this section population parameters are presented.

Gender

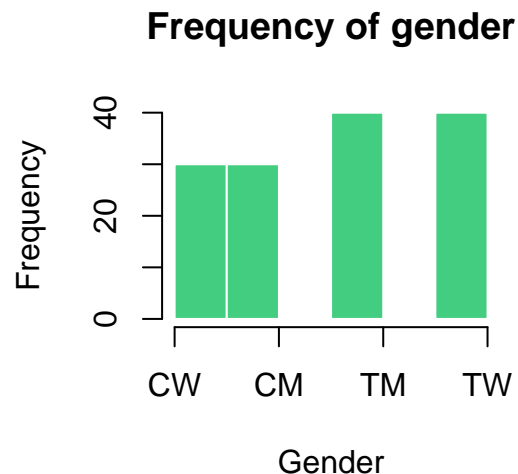
There were 60 cisgender and 80 transgender participants. One participant (P22) was removed from the analysis because no anatomical data was available for this participant.

```
ng1 <- sum(data.all[,2]==1)
ng2 <- sum(data.all[,2]==2)
ng3 <- sum(data.all[,2]==3)
ng4 <- sum(data.all[,2]==4)
```

```
paste("Cisgender women = ", ng1, ", cisgender men = ", ng2,
      ", transgender men = ", ng3, ", transgender women = ", ng4, sep="")
```

```
## [1] "Cisgender women = 30, cisgender men = 30, transgender men = 40, transgender women = 40"
```

```
hist(data.all[,2], breaks = 5, xlab = "Gender", xaxt = "n", col = "seagreen3",
      border = "white", main = "Frequency of gender")
axis(1,at=c(1:4),labels=c("CW", "CM", "TM", "TW"))
```

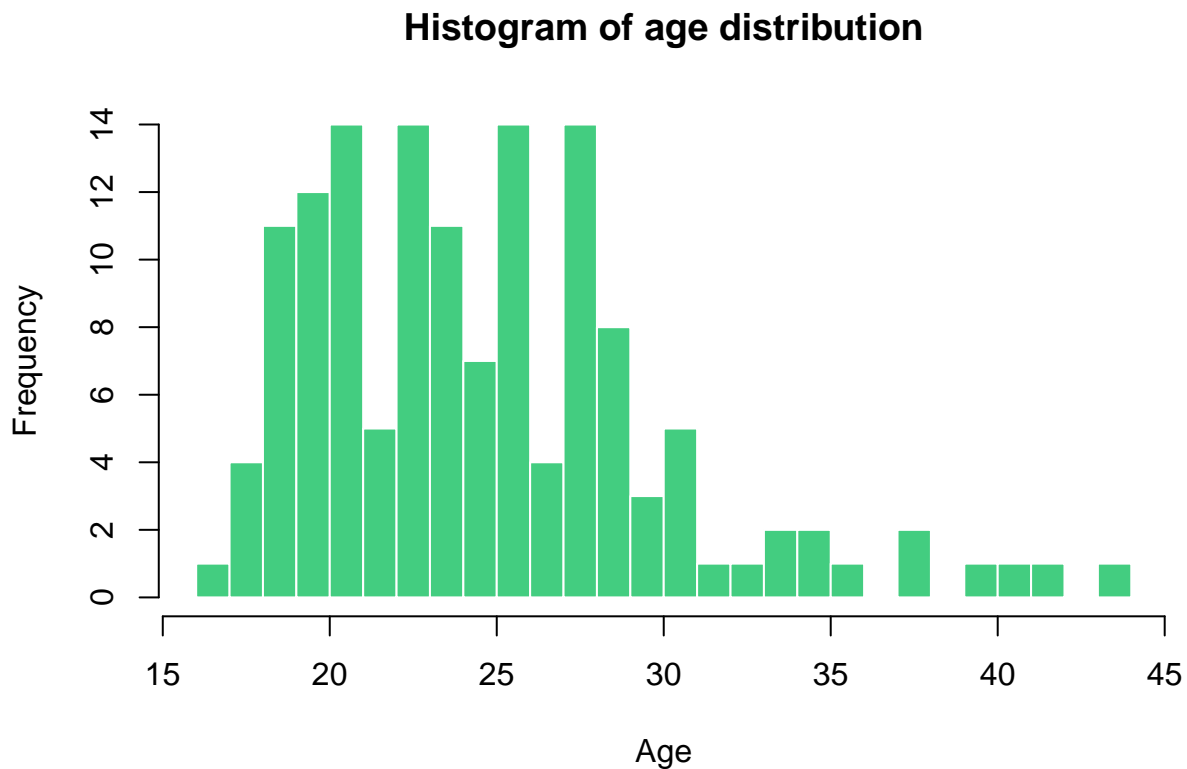


Age

```
summary(data.all[,4])
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      16.00  21.00   24.00   25.18  28.00   44.00
```

```
hist(data.all[,4], xlab = "Age", col = "seagreen3", border = "white",
      main = "Histogram of age distribution", breaks = length(unique(data.all[,4])))
```



Age of the participants ranged from 16 to 44. If we look at the distribution of age in the cis- and transgender group we see that the range is similar in both groups.

```
# Cisgender group
```

```
summary(data.all[data.all[,2]<3,4])
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      19.00  23.00   25.00   25.92  29.00   41.00
```

```
# Transgender group
```

```
summary(data.all[data.all[,2]>2,4])
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      16.00  20.75   24.00   24.62  26.25   44.00
```

Social-Economic Status

```
table(data.all[,5])
```

```
##
##  1  2  3
```

```
## 19 87 34
```

Level of education

```
table(data.all[,6])
```

```
##  
## 2 3 4 5  
## 22 76 28 14
```

Handedness

```
table(data.all[,7])
```

```
##  
## 1 2  
## 129 11
```

Gender identity

```
summary(data.all[,8])
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
##    0.000  0.000   5.000   5.036 10.000 10.000
```

```
table(data.all[,8])
```

```
##  
## 0 1 2 3 7 8 9 10  
## 49 4 13 4 1 9 16 44
```

```
# Cisgender women
```

```
summary(data.all[data.all[,2]==1,8])
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
##    7.000  9.000   9.500   9.267 10.000 10.000
```

```
# Cisgender men
```

```
summary(data.all[data.all[,2]==2,8])
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
##     0.0     0.0     0.0     0.9     2.0     3.0
```

```
# Transgender women
```

```
summary(data.all[data.all[,2]==3,8])
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
##    0.000  0.000   0.000   0.375  0.000   3.000
```

```
# Transgender men
```

```
summary(data.all[data.all[,2]==4,8])
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
##    8.000  9.000 10.000   9.625 10.000 10.000
```

Sexual orientation

```
summary(data.all[,9])

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.000  0.000   5.000   4.957 10.000 10.000

table(data.all[,9])

##
##  0  1  2  3  7  8  9 10
## 50 12  7  1  1  9 14 46

# Cisgender women
summary(data.all[data.all[,2]==1,9])

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.0000  0.0000  0.0000  0.5333  1.0000  2.0000

# Cisgender men
summary(data.all[data.all[,2]==2,9])

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     7.000   9.250 10.000   9.533 10.000 10.000

# Transgender women
summary(data.all[data.all[,2]==3,9])

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     8.000   9.000 10.000   9.475 10.000 10.000

# Transgender men
summary(data.all[data.all[,2]==4,9])

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.000   0.000   0.000   0.325   0.000   3.000
```

Mental illnesses

There are no available measures for the cisgender group. #####Somatization

```
table(data.all[,22])

##
## #NULL!      0      1
##     60     67     13
```

Obsessive-compulsive disorder

```
table(data.all[,23])

##
## #NULL!      0      1
##     60     58     22
```

Among the cisgender participants 1 had a history of obsession.

Interpersonal sensitivity

```
table(data.all[,24])
```

```
##  
## #NULL!      0      1  
##      60     57     23
```

Depression

```
table(data.all[,25])
```

```
##  
## #NULL!      0      1      2  
##      60     42     37     1
```

Among the cisgender participants 3 had a history of depression.

Anxiety

```
table(data.all[,26])
```

```
##  
## #NULL!      0      1      2  
##      60     54     25     1
```

Among the cisgender participants 1 had a history of a general anxiety disorder.

Hostility

```
table(data.all[,27])
```

```
##  
## #NULL!      0      1      2  
##      60     64     15     1
```

Phobic anxiety

```
table(data.all[,28])
```

```
##  
## #NULL!      0      1  
##      60     67     13
```

Paranoia

```
table(data.all[,29])
```

```
##  
## #NULL!      0      1  
##      60     46     34
```

Psychotism

```
table(data.all[,30])
```

```
##
## #NULL!      0      1
##      60      67      13
```

Global severity

```
table(data.all[,31])
```

```
##
## #NULL!      0      1
##      60      59      21
```

Cisgender group

Past psychiatric condition Out of 60 cisgender participants 3 had a history of depression, 1 reported a general anxiety disorder and 1 had a history of obsession.

Past medical condition Out of 60 cisgender participants 3 reported migraine, 1 reported left ear surgery, 1 participant had suffered from heart palpitations, 1 participants reported a history of meningitis, 1 participant reported they had asthma as a child and 1 participants reported favism.

Analysis of 1 measurement

```
bg <- 3
nd <- 28
ln <- nd-bg
# Object to save p-values of ANOVA
pan.one <- array(data=NA, dim = ln)

# Compute ANOVA for every predictor and save p-value
for(i in bg:nd){
  tempan <- aov(data.hyp[,i] ~ as.factor(data.hyp[,2]))
  pan.one[i-bg+1] <- unlist(summary(tempan))[9]
}

# FDR correction on p-values to correct for multiple testing
pancorr.one <- p.adjust(pan.one, method = "bonferroni")
round(pancorr.one,3)

## [1] 0.918 1.000 0.000 0.006 0.025 0.746 0.132 0.016 1.000 0.000 0.006
## [12] 0.312 0.029 0.012 0.008 0.000 0.018 0.000 0.000 0.039 0.116 1.000
## [23] 0.000 0.032 0.000 1.000

sum(pancorr.one < 0.05)

## [1] 17

names(data.hyp[,which(pancorr.one < 0.05) + bg - 1])

## [1] "T1_L_fusiform_volume"
## [2] "T1_L_inferiorparietal_volume"
## [3] "T1_L_postcentral_volume"
## [4] "T1_R_caudalanteriorcingulate_volume"
## [5] "T1_R_fusiform_volume"
## [6] "T1_R_inferiorparietal_volume"
```

```
## [7] "T1_R_precentral_volume"
## [8] "T1_R_frontalpole_volume"
## [9] "T1_LeftCerebellumWhiteMatter"
## [10] "T1_LeftCerebellumCortex"
## [11] "T1_RightCerebellumWhiteMatter"
## [12] "T1_RightCerebellumCortex"
## [13] "T1_LeftThalamusProper"
## [14] "T1_LeftCaudate"
## [15] "T1_RightThalamusProper"
## [16] "T1_RightCaudate"
## [17] "T1_RightPutamen"
```

Analysis of the average

```
bg <- 55
nd <- 80
ln <- nd-bg+1
fac <- c(rep("CW",30), rep("CM",30), rep("TM",40), rep("TW",40)) # factor for participant group

# Object to save p-values of ANOVA
pan.avg <- array(data=NA, dim = ln)

# Compute ANOVA for every predictor and save p-value
for(i in bg:nd){
  tempan <- aov(data.hyp[,i] ~ as.factor(data.hyp[,2]))
  pan.avg[i-bg+1] <- unlist(summary(tempan))[9]

  # construct boxplot for every region
  boxplot(data.hyp[,i] ~ fac, main = names(data.hyp)[i])
}

# FDR correction on p-values to correct for multiple testing
pancorr.avg <- p.adjust(pan.avg, method = "bonferroni")
round(pancorr.avg,3)
```

```
## [1] 0.049 0.000 0.040 0.000 0.698 1.000 0.001 0.076 0.089 1.000 1.000
## [12] 0.015 1.000 0.009 0.053 1.000 0.193 0.035 0.000 0.073 0.008 1.000
## [23] 0.000 0.146 0.022 0.292
```

```
sum(pancorr.avg < 0.05)
```

```
## [1] 12
```

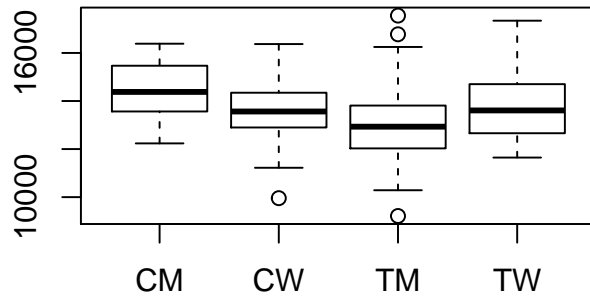
```
names(data.hyp[,which(pancorr.avg < 0.05) + bg - 1])
```

```
## [1] "Tavg_LeftCerebellumWhiteMatter"
## [2] "Tavg_LeftCerebellumCortex"
## [3] "Tavg_RightCerebellumWhiteMatter"
## [4] "Tavg_RightCerebellumCortex"
## [5] "Tavg_L_fusiform_volume"
## [6] "Tavg_R_caudalanteriorcingulate_volume"
## [7] "Tavg_R_fusiform_volume"
## [8] "Tavg_R_frontalpole_volume"
## [9] "Tavg_LeftThalamusProper"
```

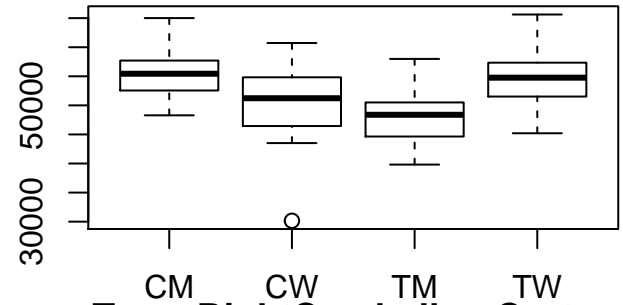


```
## [10] "Tavg_LeftPutamen"
## [11] "Tavg_RightThalamusProper"
## [12] "Tavg_RightPutamen"
```

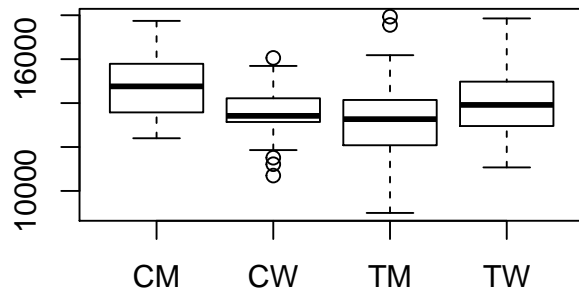
Tavg_LeftCerebellumWhiteMatter



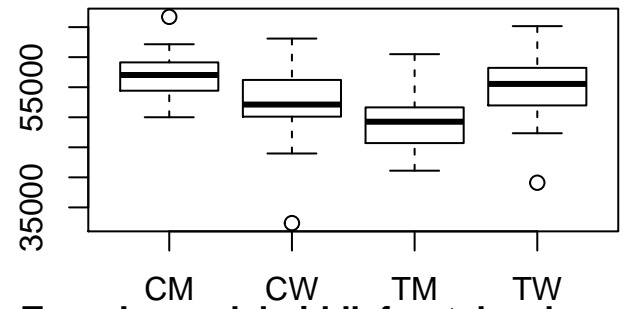
Tavg_LeftCerebellumCortex



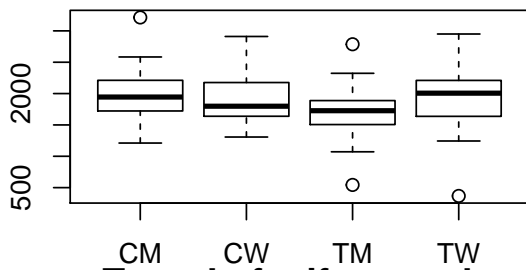
Tavg_RightCerebellumWhiteMatter



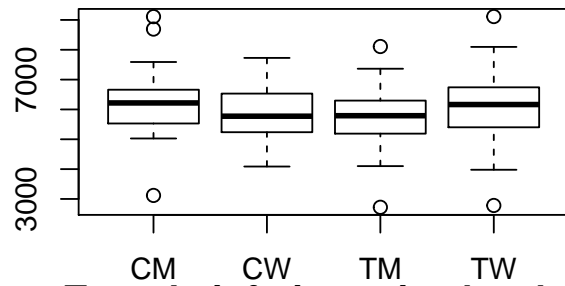
Tavg_RightCerebellumCortex



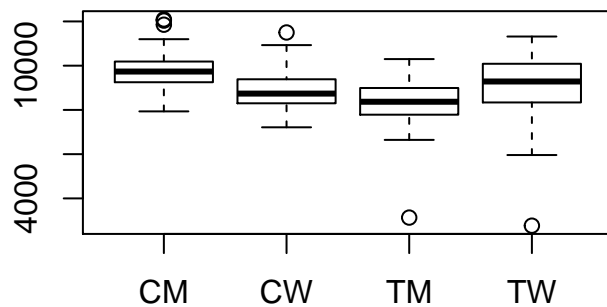
Tavg_L_caudalanteriorcingulate_volume



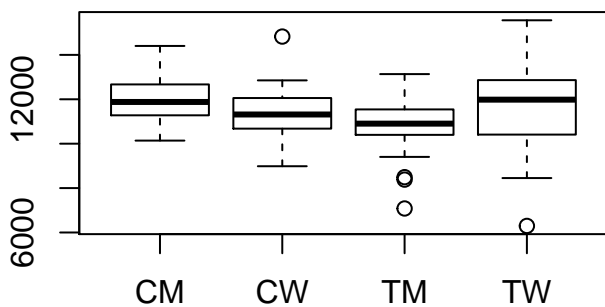
Tavg_L_caudalmiddlefrontal_volume



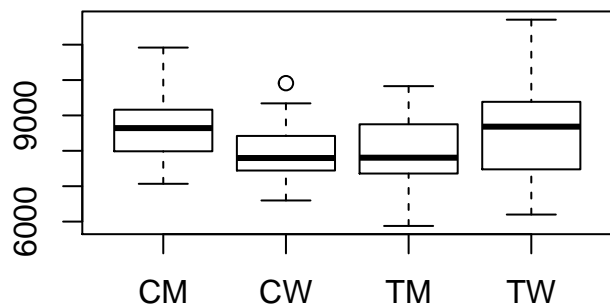
Tavg_L_fusiform_volume



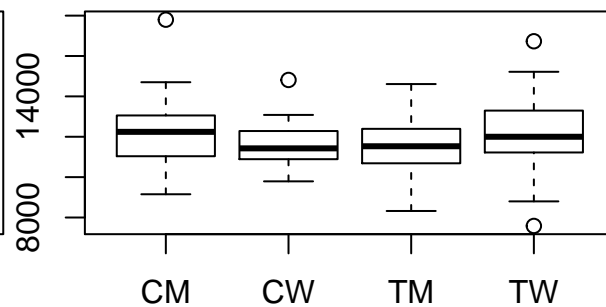
Tavg_L_inferiorparietal_volume



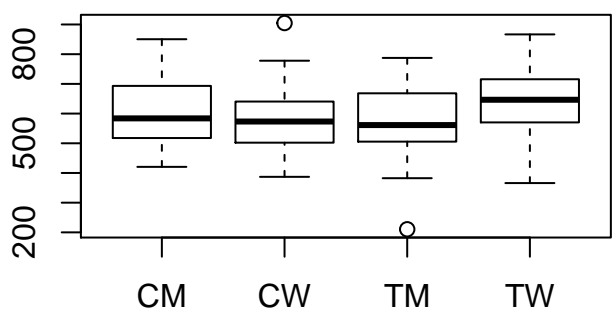
Tavg_L_postcentral_volume



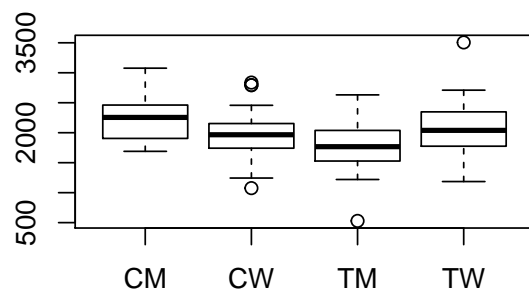
Tavg_L_precentral_volume



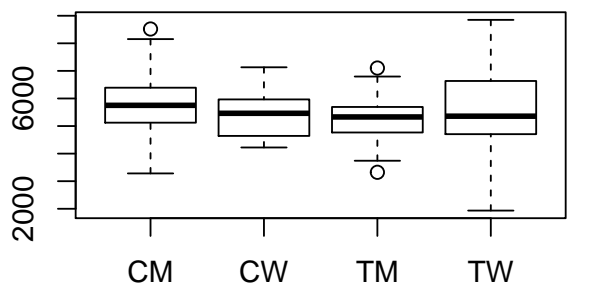
Tavg_L_frontalpole_volume



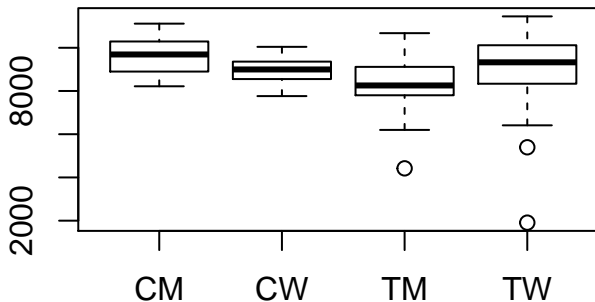
Tavg_R_caudalanteriorcingulate_volume



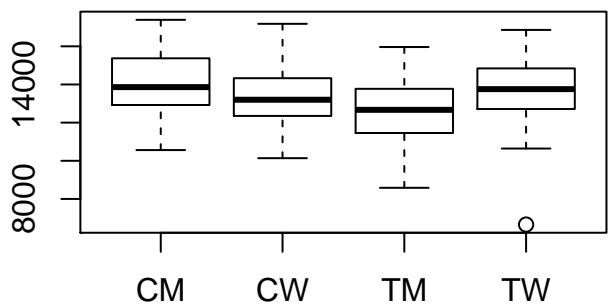
Tavg_R_caudalmiddlefrontal_volume



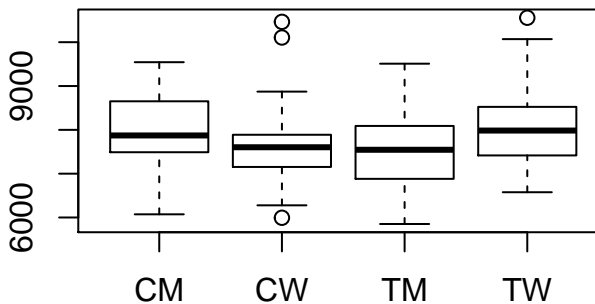
Tavg_R_fusiform_volume

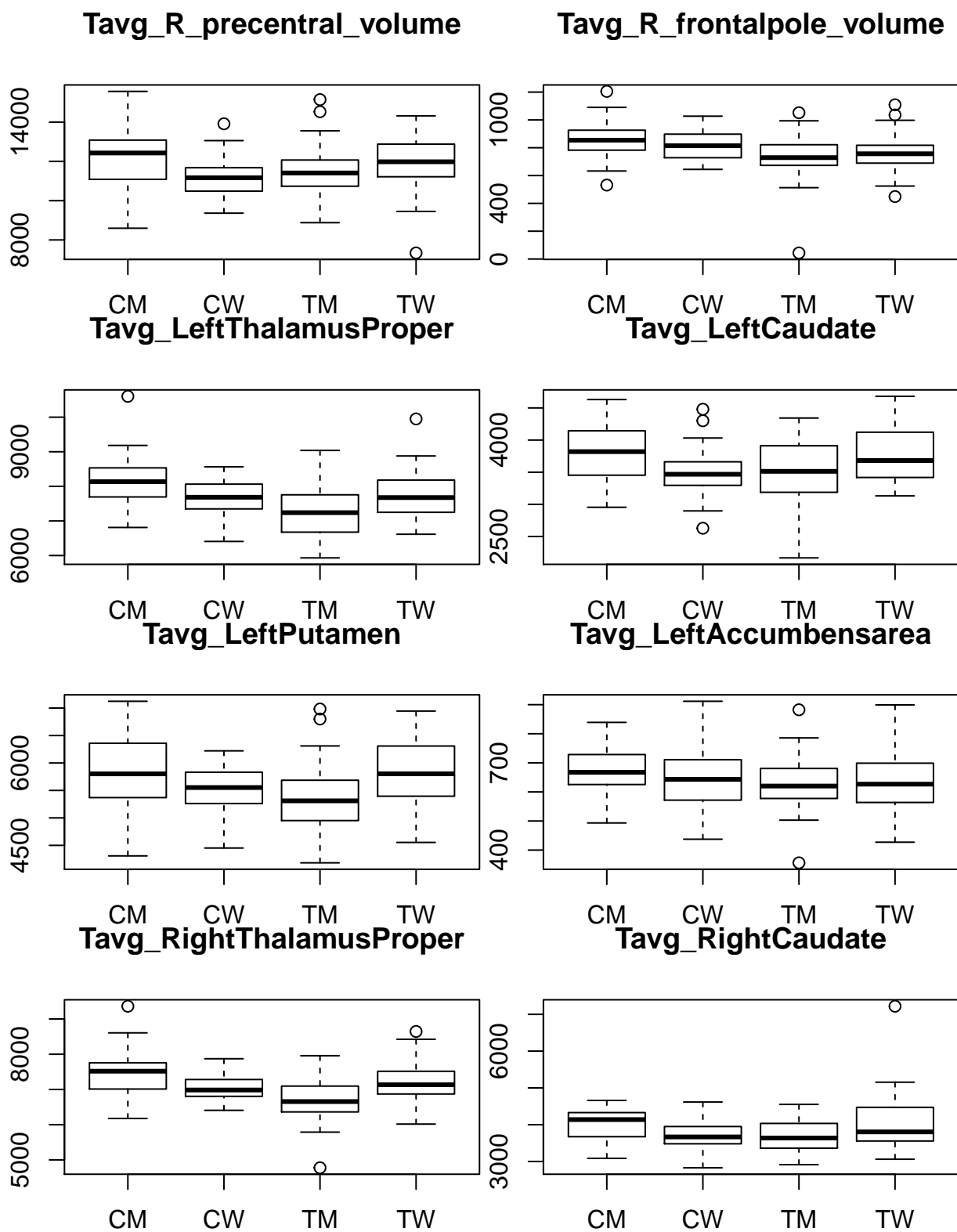


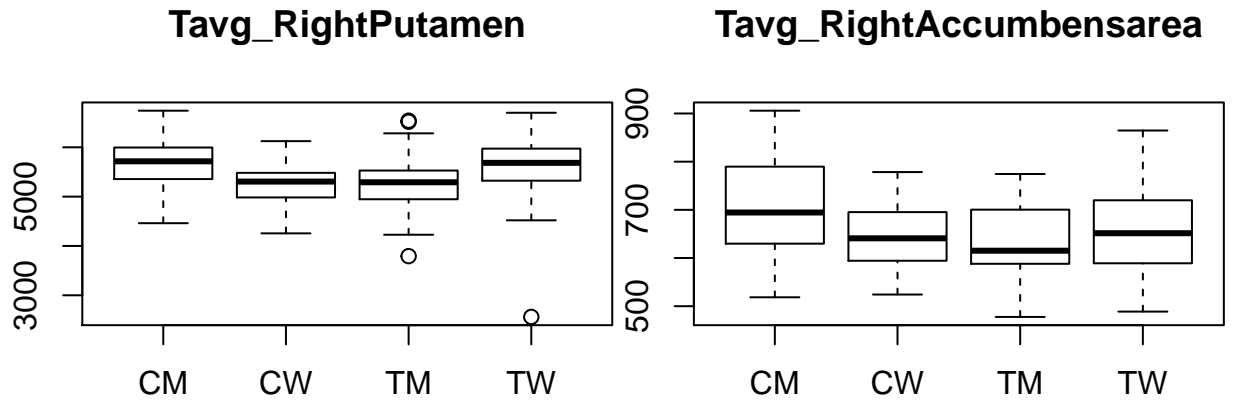
Tavg_R_inferiorparietal_volume



Tavg_R_postcentral_volume

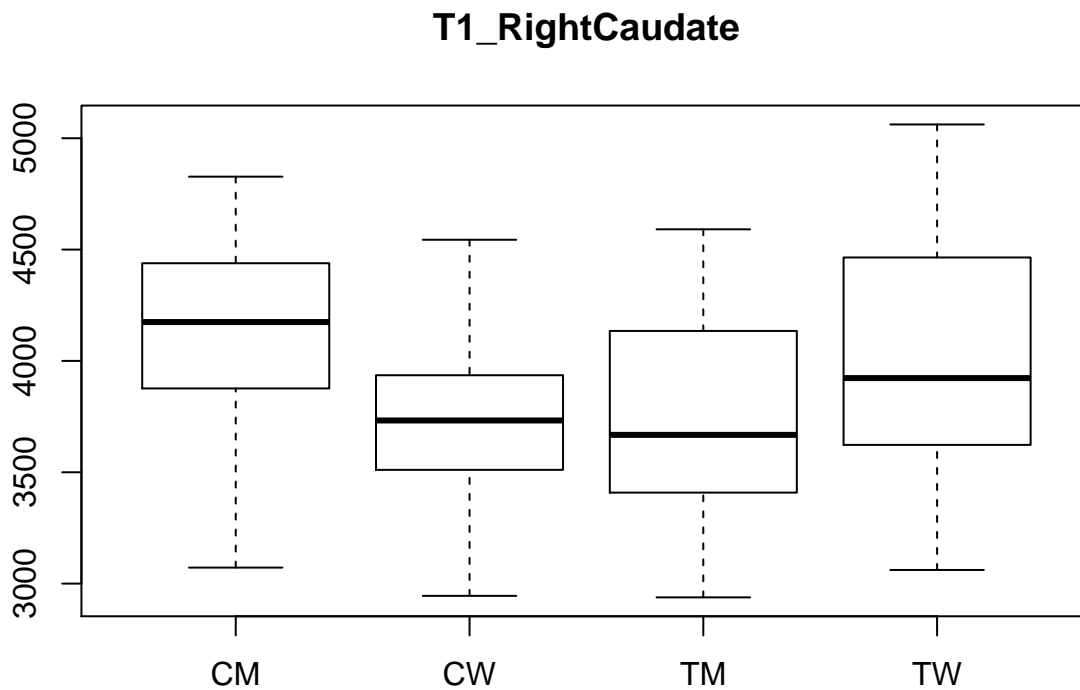






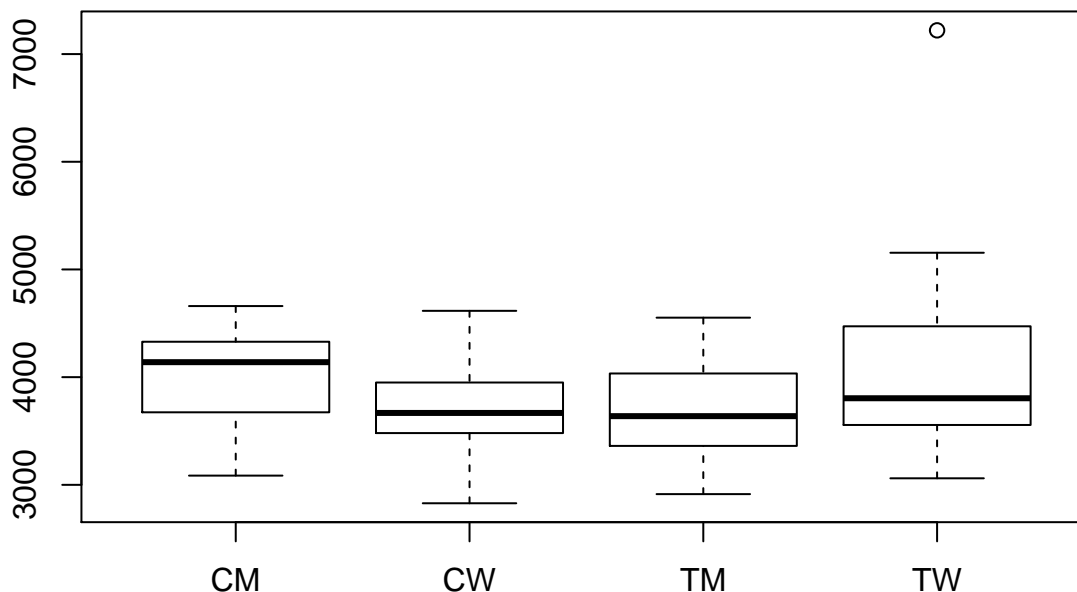
Why are there less regions for which the difference is statistically significant when the average is used compared to when one measure is used? To investigate this we look at the difference between the boxplots for one statistically significant region.

```
boxplot(data.hyp[,26] ~ fac, main = names(data.hyp)[26])
```



```
boxplot(data.hyp[,78] ~ fac, main = names(data.hyp)[78])
```

Tavg_RightCaudate



Post-hoc tests

We conduct post-hoc t-tests on the statistically significant regions to determine which group differences cause the effect.

```
# create an object with all possible combinations
allcomb <- combn(c(1:4),2)

# create object to save results
pt.one <- array(data = NA, dim = c(length(pan.one),dim(allcomb)[2]))
pt.avg <- array(data = NA, dim = c(length(pan.avg),dim(allcomb)[2]))

# Left Cerebellum White Matter
# 1 measure
bg <- 3
nd <- 28
for(r in bg:nd)
  for(i in 1:dim(allcomb)[2]){
    pt.one[r-bg+1,i] <- unlist(t.test(data.hyp[data.hyp[,2]==allcomb[1,i],r], data.hyp[data.hyp[,2]==
  })
}

round(pt.one,3)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] 0.812 0.028 0.980 0.040 0.762 0.007
## [2,] 0.149 0.507 0.291 0.029 0.599 0.057
## [3,] 0.002 0.008 0.289 0.000 0.034 0.000
## [4,] 0.023 0.026 0.469 0.000 0.158 0.004
## [5,] 0.002 0.504 0.039 0.000 0.386 0.010
## [6,] 0.047 0.524 0.135 0.015 0.515 0.040
## [7,] 0.155 0.887 0.003 0.166 0.178 0.002
## [8,] 0.012 0.142 0.449 0.000 0.066 0.017
```

```
## [9,] 0.099 0.430 0.261 0.027 0.576 0.079
## [10,] 0.002 0.003 0.258 0.000 0.130 0.001
## [11,] 0.027 0.095 0.178 0.000 0.255 0.001
## [12,] 0.145 0.371 0.089 0.011 0.891 0.003
## [13,] 0.002 0.830 0.004 0.006 0.369 0.015
## [14,] 0.001 0.938 0.104 0.001 0.029 0.121
## [15,] 0.006 0.129 0.934 0.000 0.002 0.113
## [16,] 0.001 0.098 0.002 0.000 0.686 0.000
## [17,] 0.007 0.174 0.490 0.000 0.022 0.037
## [18,] 0.001 0.038 0.010 0.000 0.251 0.000
## [19,] 0.001 0.033 0.102 0.000 0.029 0.000
## [20,] 0.004 0.973 0.012 0.003 0.481 0.011
## [21,] 0.023 0.467 0.070 0.005 0.361 0.011
## [22,] 0.196 0.874 0.912 0.135 0.183 0.973
## [23,] 0.021 0.008 0.310 0.000 0.129 0.001
## [24,] 0.001 0.858 0.021 0.001 0.254 0.034
## [25,] 0.000 0.401 0.000 0.000 0.179 0.005
## [26,] 0.531 0.051 0.540 0.014 0.247 0.283
```

```
# Average
```

```
bg <- 55
```

```
nd <- 80
```

```
for(r in bg:nd)
```

```
  for(i in 1:dim(allcomb)[2]){
```

```
    pt.avg[r-bg+1,i] <- unlist(t.test(data.hyp[data.hyp[,2]==allcomb[1,i],r], data.hyp[data.hyp[,2]==
```

```
  })
```

```
round(pt.avg,3)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] 0.015 0.153 0.471 0.000 0.058 0.029
## [2,] 0.001 0.047 0.005 0.000 0.363 0.000
## [3,] 0.001 0.373 0.195 0.000 0.055 0.047
## [4,] 0.000 0.022 0.016 0.000 0.162 0.000
## [5,] 0.769 0.021 0.894 0.011 0.868 0.013
## [6,] 0.233 0.423 0.352 0.054 0.759 0.087
## [7,] 0.002 0.023 0.651 0.000 0.025 0.017
## [8,] 0.024 0.065 0.308 0.000 0.443 0.012
## [9,] 0.001 0.894 0.024 0.002 0.497 0.031
## [10,] 0.080 0.946 0.085 0.093 0.835 0.102
## [11,] 0.446 0.686 0.073 0.223 0.315 0.018
## [12,] 0.012 0.100 0.559 0.000 0.046 0.023
## [13,] 0.131 0.461 0.279 0.035 0.679 0.088
## [14,] 0.001 0.004 0.845 0.000 0.065 0.037
## [15,] 0.059 0.100 0.184 0.001 0.529 0.003
## [16,] 0.200 0.920 0.036 0.122 0.406 0.012
## [17,] 0.004 0.198 0.002 0.068 0.766 0.075
## [18,] 0.260 0.010 0.039 0.001 0.005 0.443
## [19,] 0.006 0.015 0.580 0.000 0.024 0.004
## [20,] 0.008 0.753 0.003 0.017 0.939 0.009
## [21,] 0.007 0.418 0.007 0.002 0.568 0.001
## [22,] 0.346 0.483 0.915 0.068 0.261 0.521
## [23,] 0.004 0.012 0.082 0.000 0.152 0.001
## [24,] 0.002 0.685 0.015 0.004 0.972 0.028
## [25,] 0.001 0.710 0.005 0.003 0.588 0.014
```

```
## [26,] 0.030 0.349 0.699 0.003 0.066 0.187
```

Correlation between T1 and T2

```
bg <- 36
nd <- 108
ln <- nd-bg+1

# Object to save correlations
corrall <- array(data=NA, dim = ln)

# Compute ANOVA for every predictor and save p-value
for(i in bg:nd){
  corrall[i-bg+1] <- cor(x = data.all[,i], y = data.all[,i + ln + 1])
}

summary(corrall)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.6334  0.7950  0.8243  0.8216  0.8466  0.9985
```

```
bg <- 3
nd <- 28
ln <- nd-bg+1

# Object to save correlations
corrhyp <- array(data=NA, dim = ln)

# Compute ANOVA for every predictor and save p-value
for(i in bg:nd){
  corrhyp[i-bg+1] <- cor(x = data.hyp[,i], y = data.hyp[,i + ln])
}

summary(corrhyp)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.7046  0.9050  0.9334  0.9056  0.9498  0.9963
```

Simulations

RESULTS OF THE SIMULATIONS STILL NEED TO FINISH, these take a long time to run. ## Intro
Code

First we need to define the parameters of our simulations.

```
# variance/sd epsilon
seps <- 1

# Number of simulations
asim <- 10

# Effect size
delta <- 0.8
```

```

# Number of participants
n <- 30
n.1 <- n/2      # in the first group
n.2 <- n/2      # in the second group

# Level of statistical significance
alpha <- 0.05

# Correlation between first and second measurement
rho <- seq(0.01,0.99,0.01)

```

Then we prepare objects to store our results

```

# Number of simulations
pow.mean1<-vector("numeric",length(rho))
pow.mean2<-vector("numeric",length(rho))
pow.mean3<-vector("numeric",length(rho))
pow.mean4<-vector("numeric",length(rho))

```

```
## Loading required package: Matrix
```

```
##
```

```
## Attaching package: 'lmerTest'
```

```
## The following object is masked from 'package:lme4':
```

```
##
```

```
##      lmer
```

```
## The following object is masked from 'package:stats':
```

```
##
```

```
##      step
```

```
# Loop over preset correlations between measure 1 and measure 2
```

```
for(i in 1:length(rho)){
```

```
  # Create objects to store power in for every simulations
```

```
  pow.1<-vector("numeric",asim)
```

```
  pow.2<-vector("numeric",asim)
```

```
  pow.3<-vector("numeric",asim)
```

```
  pow.4<-vector("numeric",asim)
```

```
  for(k in 1:asim){
```

```
    # Scenario 1: lower bound of power curve
```

```
    # two groups with equal amount of subjects, groups differ with an effect size delta
```

```
    # Construct a vector that determines in which group each subject falls
```

```
    x<-c(rep(1,n.1),rep(0,n.2))
```

```
    # Vector with observations in the set of participants
```

```
    y<-rnorm(n,0,seps)
```

```
    # Add an effect size to the first group
```

```
    y[1:n.1]<-y[1:n.1]+delta
```

```
    # Boolean of whether an effect is detected, this is later used to compute the power
```

```
    pow.1[k]<-summary(lm(y~x))$coef[2,4]<alpha
```

```
    # Scenario 2: upper bound of power curve
```

```
    # two groups with equal amount of subjects, twice as many as scenario 1, groups differ with an effect size delta
```



```

# Construct a vector that determines in which group each subject falls
x2<-c(rep(1,(n.1*2)),rep(0,(n.2*2)))

# Vector with observations in the set of participants
y2<-rnorm(n*2,0,seps)
# Add an effect size to the first group
y2[1:(n.1*2)]<-y2[1:(n.1*2)]+delta

# Boolean of whether an effect is detected, this is later used to compute the power
pow.2[k]<-summary(lm(y2~x2))$coef[2,4]<alpha

# Scenario 3: two measurements for every subject, same amount of subjects as in scenario 1
# two groups with equal amount of subjects, correlation between measurements, groups differ with an effect
# Construct a vector that determines in which group each subject falls
x3<-c(rep(1,n.1),rep(0,n.2),rep(1,n.1),rep(0,n.2))

# Vector with first observation of every participant
y3<-rnorm(n,0,seps)
# Factor to multiply second set of observations with to obtain results in line with predefined correlation
alpac<-sqrt(rho[i]^2/(1-rho[i]^2)*seps)
# Construct second set of observations that are correlated with first set (y3)
y3.2u<-alpac*y3+rnorm(n)
y3.2<-y3.2u/sqrt(var(y3.2u))
# Add effect size to the first group of participants
y3[1:n.1]<-y3[1:n.1]+delta
y3.2[1:n.1]<-y3.2[1:n.1]+delta
# Combine both observations in 1 vector
y3o<-c(y3,y3.2)

# Define subject numbers
subject<-rep(1:n,2)
# Construct mixed model
mm<-lmer(y3o ~ x3 + (1 | subject))
# Boolean of whether an effect is detected, this is later used to compute the power
pow.3[k]<-summary(mm)$coef[2,5]<alpha

# Scenario 4: What if we work with the average?
y3m<-(y3+y3.2)/2
# Boolean of whether an effect is detected, this is later used to compute the power
pow.4[k]<-summary(lm(y3m~x))$coef[2,4]<alpha
}
pow.mean1[i]<-mean(pow.1)
pow.mean2[i]<-mean(pow.2)
pow.mean3[i]<-mean(pow.3)
pow.mean4[i]<-mean(pow.4)
}

```

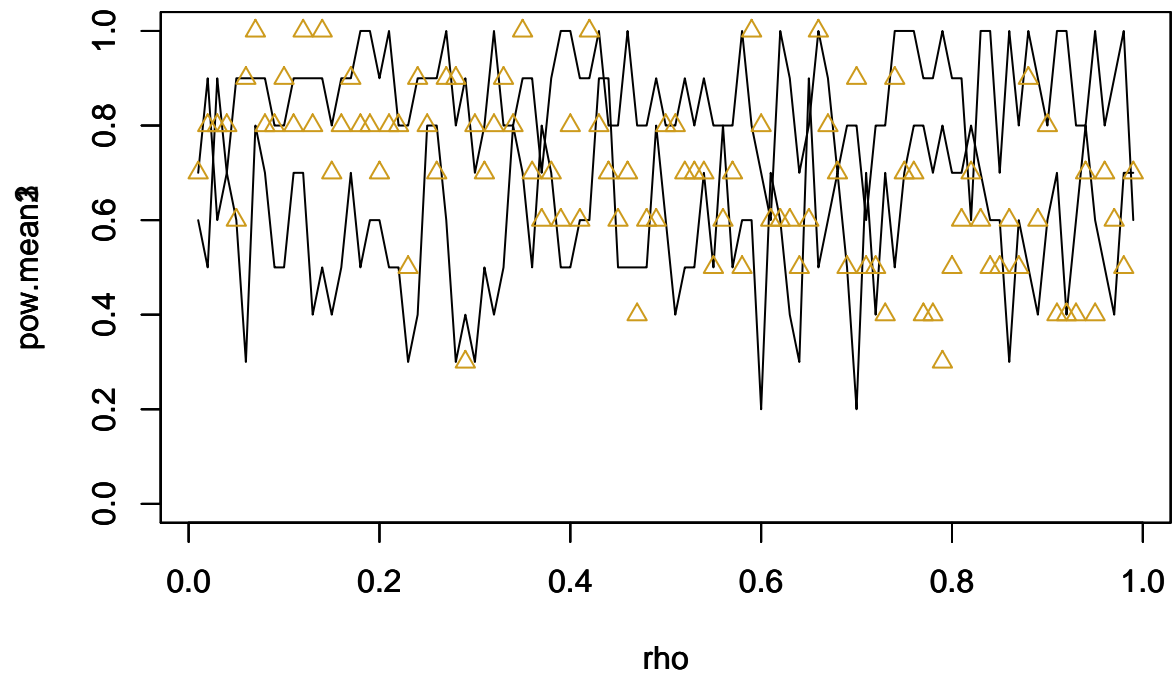
Results

```

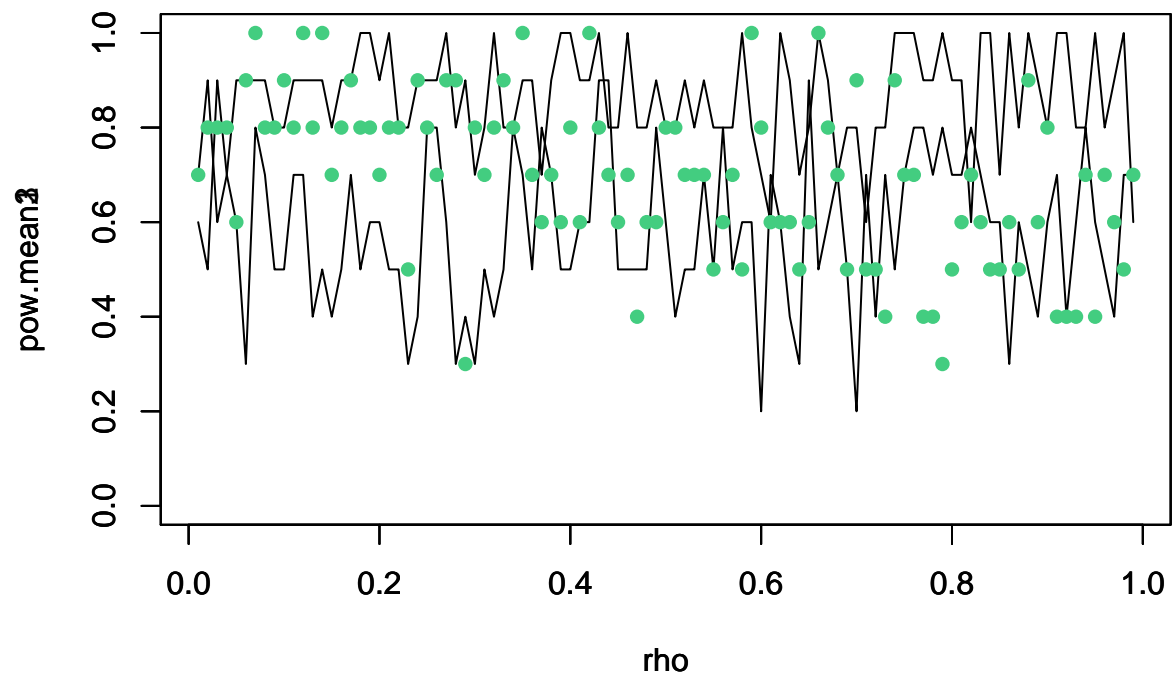
# power of taking both measures into account
plot(rho,pow.mean1,ylim=c(0,1), type="l")
par(new = TRUE)

```

```
plot(rho,pow.mean2,ylim=c(0,1), type="l")
par(new = TRUE)
plot(rho,pow.mean3,ylim=c(0,1), type="p", col = "goldenrod3", pch = 2)
```



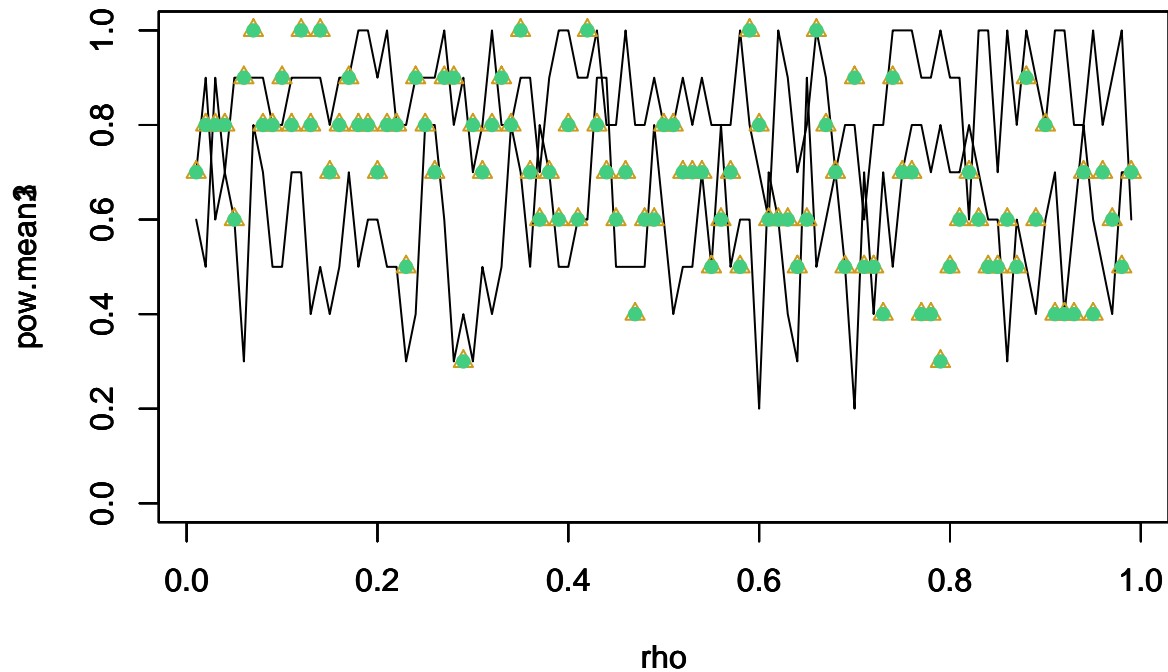
```
# power of using the average
plot(rho,pow.mean1,ylim=c(0,1), type="l")
par(new = TRUE)
plot(rho,pow.mean2,ylim=c(0,1), type="l")
par(new = TRUE)
plot(rho,pow.mean3,ylim=c(0,1), type="p", col = "seagreen3", pch = 16)
```



If we overlay both plots we see that there is a lot of overlap. This can also be demonstrated by looking at the results that are exactly the same.

```
table(pow.mean3==pow.mean4)
```

```
##
## FALSE  TRUE
##      1    98
# Overlap of both plots
plot(rho,pow.mean1,ylim=c(0,1), type="l")
par(new = TRUE)
plot(rho,pow.mean2,ylim=c(0,1), type="l")
par(new = TRUE)
plot(rho,pow.mean3,ylim=c(0,1), type="p", col = "goldenrod3", pch = 2)
par(new = TRUE)
plot(rho,pow.mean3,ylim=c(0,1), type="p", col = "seagreen3", pch = 16)
```



Discussion