对成像数据进行分析

通过spm\_fNIRS 工具箱中的 spm\_fnirs\_con2d 函数，对所有的数据转换成标准的 NIfTI-1 格式。随后使用 RNifti 包进行读取。

#install.packages("devtools")  
#devtools::install\_github("jonclayden/RNifti")  
library(RNifti)  
library(tidyverse)

## Loading tidyverse: ggplot2  
## Loading tidyverse: tibble  
## Loading tidyverse: tidyr  
## Loading tidyverse: readr  
## Loading tidyverse: purrr  
## Loading tidyverse: dplyr

## Conflicts with tidy packages ----------------------------------------------

## filter(): dplyr, stats  
## lag(): dplyr, stats

library(stringr)  
library(R.matlab)

## R.matlab v3.6.1 (2016-10-19) successfully loaded. See ?R.matlab for help.

##   
## Attaching package: 'R.matlab'

## The following objects are masked from 'package:base':  
##   
## getOption, isOpen

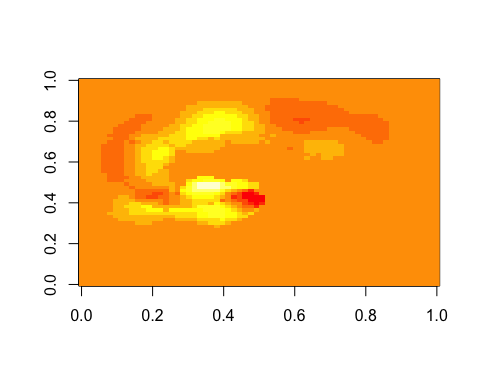
extract\_fNIRS\_beta <- function(filename) {  
 beta\_data <- readMat(filename)$S[1,,]$cbeta  
 return(beta\_data)  
}  
  
# 计算多个个体nii文件的均值矩阵  
mean\_group\_nii <- function(nii\_list) {  
 dim\_single <- dim(nii\_list[[1]])  
 file\_num <- length(nii\_list)  
 dim\_array <- c(dim\_single, file\_num)  
 total\_array <- array(dim = dim\_array)  
 for(i in 1:length(nii\_list)){  
 total\_array[,,i] <- matrix(as.numeric(nii\_list[[i]]), nrow = dim\_single[1], ncol = dim\_single[2])  
 }  
  
 mean\_matrix <- apply(total\_array, MARGIN = 1:2, mean)  
  
 return(mean\_matrix)  
}

## 实验1

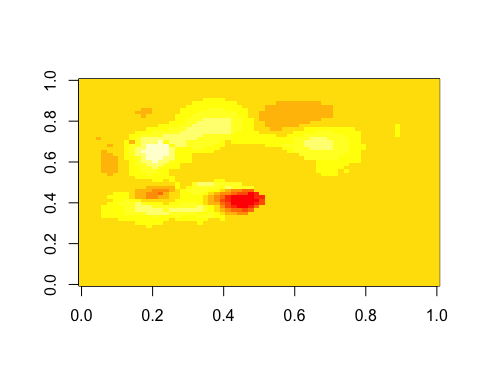
### 成像数据

读取数据进行分析

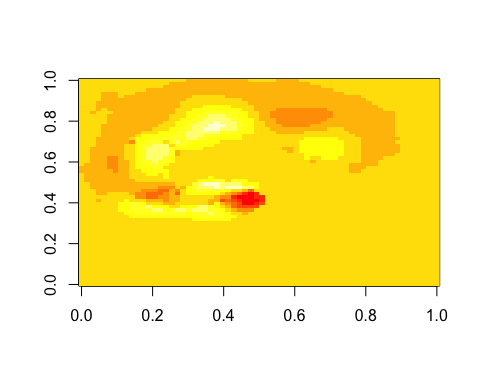
test1\_nii <- dir("../../../Dropbox/毕业设计/实验数据/仪器数据/SPM处理结果/test1/", include.dirs = T, recursive = T)  
test1\_nii <- test1\_nii[stringr::str\_detect(test1\_nii, pattern = "nii")]  
test1\_nii <- paste0("../../../Dropbox/毕业设计/实验数据/仪器数据/SPM处理结果/test1/", test1\_nii)  
test1\_con\_01 <- test1\_nii[stringr::str\_detect(test1\_nii, pattern = "con\_0001.nii")]  
test1\_con\_02 <- test1\_nii[stringr::str\_detect(test1\_nii, pattern = "con\_0002.nii")]  
test1\_con\_03 <- test1\_nii[stringr::str\_detect(test1\_nii, pattern = "con\_0003.nii")]  
  
test1\_con\_01\_images <- readNifti(test1\_con\_01)  
test1\_con\_02\_images <- readNifti(test1\_con\_02)  
test1\_con\_03\_images <- readNifti(test1\_con\_03)  
  
image(mean\_group\_nii(test1\_con\_01\_images))



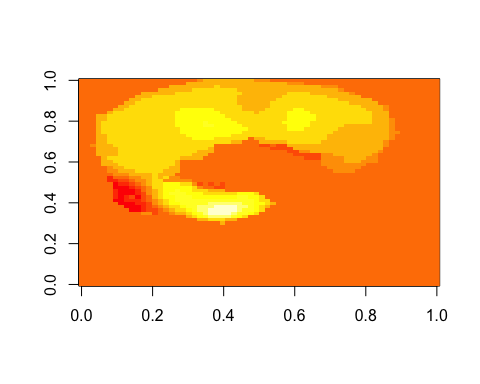
image(mean\_group\_nii(test1\_con\_02\_images))



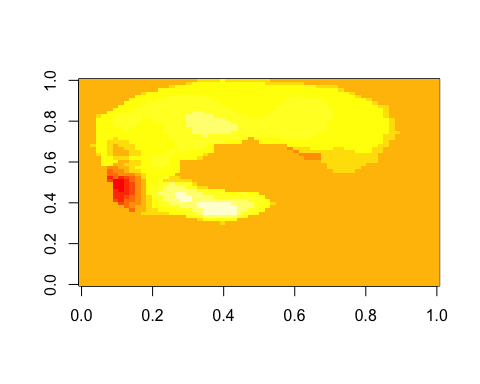
image(mean\_group\_nii(test1\_con\_03\_images))



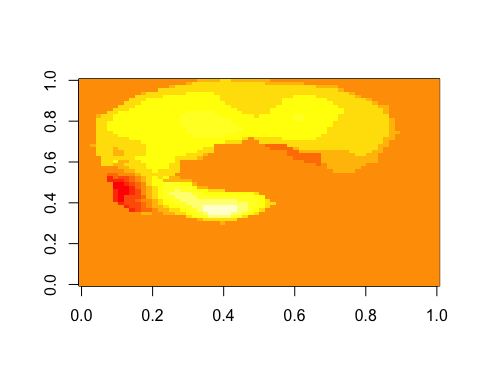
doctor\_nii <- dir("../../../Dropbox/毕业设计/实验数据/仪器数据/SPM处理结果/doctor/", include.dirs = T, recursive = T)  
doctor\_nii <- doctor\_nii[stringr::str\_detect(doctor\_nii, pattern = "nii")]  
doctor\_nii <- paste0("../../../Dropbox/毕业设计/实验数据/仪器数据/SPM处理结果/doctor/", doctor\_nii)  
doctor\_con\_01 <- doctor\_nii[stringr::str\_detect(doctor\_nii, pattern = "con\_0001.nii")]  
doctor\_con\_02 <- doctor\_nii[stringr::str\_detect(doctor\_nii, pattern = "con\_0002.nii")]  
doctor\_con\_03 <- doctor\_nii[stringr::str\_detect(doctor\_nii, pattern = "con\_0003.nii")]  
  
doctor\_con\_01\_images <- readNifti(doctor\_con\_01)  
doctor\_con\_02\_images <- readNifti(doctor\_con\_02)  
doctor\_con\_03\_images <- readNifti(doctor\_con\_03)  
  
image(mean\_group\_nii(doctor\_con\_01\_images))



image(mean\_group\_nii(doctor\_con\_02\_images))



image(mean\_group\_nii(doctor\_con\_03\_images))



### 组间的假设检验

#普通专业被试的 Block2 的 beta 值  
test1\_doctor\_files <- dir("/Users/eXrld/Dropbox/毕业设计/实验数据/仪器数据/SPM处理结果/test1", include.dirs = T, recursive = T)  
test1\_doctor\_con2 <- test1\_doctor\_files[stringr::str\_detect(test1\_doctor\_files, pattern = "con\_0002.mat")]  
test1\_doctor\_con2 <- paste0("/Users/eXrld/Dropbox/毕业设计/实验数据/仪器数据/SPM处理结果/test1/", test1\_doctor\_con2)  
  
test1\_normal\_beta <- matrix(NA, ncol = length(extract\_fNIRS\_beta(test1\_doctor\_con2[1])), nrow = length(test1\_doctor\_con2))  
for(i in 1:length(test1\_doctor\_con2)){  
 test1\_normal\_beta[i,] <- extract\_fNIRS\_beta(test1\_doctor\_con2[i])  
}  
  
#医学专业被试的 Block2 的 beta 值  
test1\_doctor\_files <- dir("/Users/eXrld/Dropbox/毕业设计/实验数据/仪器数据/SPM处理结果/doctor", include.dirs = T, recursive = T)  
test1\_doctor\_con2 <- test1\_doctor\_files[stringr::str\_detect(test1\_doctor\_files, pattern = "con\_0002.mat")]  
test1\_doctor\_con2 <- paste0("/Users/eXrld/Dropbox/毕业设计/实验数据/仪器数据/SPM处理结果/doctor/", test1\_doctor\_con2)  
  
test1\_doctor\_beta <- matrix(NA, ncol = length(extract\_fNIRS\_beta(test1\_doctor\_con2[1])), nrow = length(test1\_doctor\_con2))  
for(i in 1:length(test1\_doctor\_con2)){  
 test1\_doctor\_beta[i,] <- extract\_fNIRS\_beta(test1\_doctor\_con2[i])  
}  
  
test1\_ttest\_value <- data.frame(matrix(0, nrow = 45, ncol = 3))  
names(test1\_ttest\_value) <- c("Ch", "p.value", "sig.")  
for(i in 1:ncol(test1\_doctor\_beta)){  
 test1\_ttest\_value[i,1] <- i  
 test1\_ttest\_value[i,2] <- t.test(test1\_normal\_beta[,i], test1\_doctor\_beta[,i])$p.value  
 if(t.test(test1\_normal\_beta[,i], test1\_doctor\_beta[,i])$p.value < 0.05){  
 test1\_ttest\_value[i,3] <- "\*\*"  
 }  
}  
test1\_ttest\_value

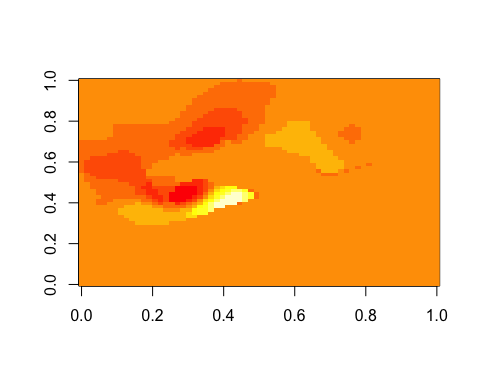
## Ch p.value sig.  
## 1 1 0.08661732 0  
## 2 2 0.05064175 0  
## 3 3 0.09208531 0  
## 4 4 0.03855873 \*\*  
## 5 5 0.08594881 0  
## 6 6 0.08700083 0  
## 7 7 0.27338055 0  
## 8 8 0.04119775 \*\*  
## 9 9 0.11026676 0  
## 10 10 0.11654180 0  
## 11 11 0.09284784 0  
## 12 12 0.12863494 0  
## 13 13 0.02291570 \*\*  
## 14 14 0.04864954 \*\*  
## 15 15 0.09842162 0  
## 16 16 0.10510523 0  
## 17 17 0.10634044 0  
## 18 18 0.07791025 0  
## 19 19 0.12701393 0  
## 20 20 0.00623695 \*\*  
## 21 21 0.08662543 0  
## 22 22 0.22678496 0  
## 23 23 0.13481355 0  
## 24 24 0.04011048 \*\*  
## 25 25 0.90128329 0  
## 26 26 0.09398913 0  
## 27 27 0.60416995 0  
## 28 28 0.65152125 0  
## 29 29 0.37269339 0  
## 30 30 0.09879713 0  
## 31 31 0.65950946 0  
## 32 32 0.89979998 0  
## 33 33 0.46646834 0  
## 34 34 0.90631588 0  
## 35 35 0.02434314 \*\*  
## 36 36 0.11999021 0  
## 37 37 0.11678129 0  
## 38 38 0.12047151 0  
## 39 39 0.36908784 0  
## 40 40 0.18310846 0  
## 41 41 0.18222903 0  
## 42 42 0.83436438 0  
## 43 43 0.31000105 0  
## 44 44 0.35617908 0  
## 45 45 0.44069249 0

## 实验2

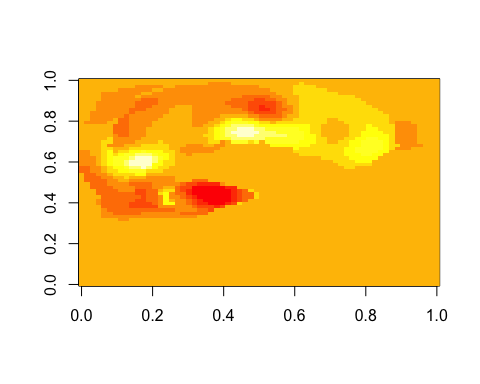
### 读取数据

对要分析的数据进行读取

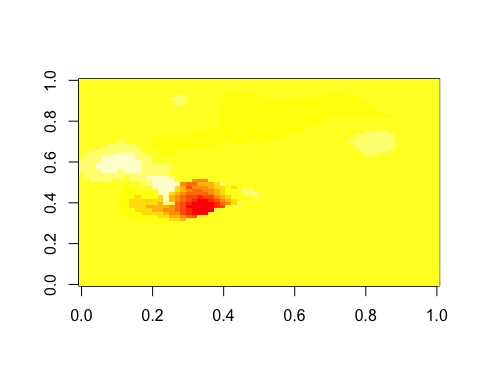
test2\_nii <- dir("../../../Dropbox/毕业设计/实验数据/仪器数据/SPM处理结果/test2/", include.dirs = T, recursive = T)  
  
test2\_nii <- test2\_nii[stringr::str\_detect(test2\_nii, pattern = "nii")]  
now\_path <- getwd()  
#setwd("../../../Dropbox/毕业设计/实验数据/仪器数据/SPM处理结果/test2/")  
  
test2\_nii <- paste0("../../../Dropbox/毕业设计/实验数据/仪器数据/SPM处理结果/test2/", test2\_nii)  
  
con\_01 <- test2\_nii[stringr::str\_detect(test2\_nii, pattern = "con\_0001.nii")]  
con\_02 <- test2\_nii[stringr::str\_detect(test2\_nii, pattern = "con\_0002.nii")]  
con\_03 <- test2\_nii[stringr::str\_detect(test2\_nii, pattern = "con\_0003.nii")]  
con\_04 <- test2\_nii[stringr::str\_detect(test2\_nii, pattern = "con\_0004.nii")]  
con\_05 <- test2\_nii[stringr::str\_detect(test2\_nii, pattern = "con\_0005.nii")]  
  
con\_01\_images <- readNifti(con\_01)  
con\_02\_images <- readNifti(con\_02)  
con\_03\_images <- readNifti(con\_03)  
con\_04\_images <- readNifti(con\_04)  
con\_05\_images <- readNifti(con\_05)  
  
mean\_group\_nii <- function(nii\_list) {  
 dim\_single <- dim(nii\_list[[1]])  
 file\_num <- length(nii\_list)  
 dim\_array <- c(dim\_single, file\_num)  
 total\_array <- array(dim = dim\_array)  
 for(i in 1:length(nii\_list)){  
 total\_array[,,i] <- matrix(as.numeric(nii\_list[[i]]), nrow = dim\_single[1], ncol = dim\_single[2])  
 }  
   
 mean\_matrix <- apply(total\_array, MARGIN = 1:2, mean)  
   
 return(mean\_matrix)  
}  
  
image(mean\_group\_nii(con\_01\_images))



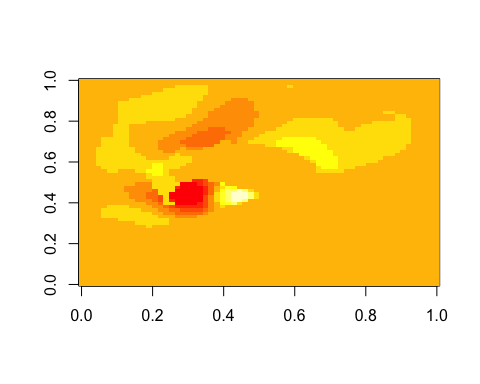
image(mean\_group\_nii(con\_02\_images))



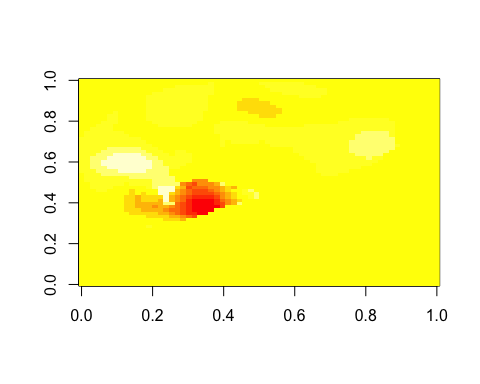
image(mean\_group\_nii(con\_03\_images))



image(mean\_group\_nii(con\_04\_images))



image(mean\_group\_nii(con\_05\_images))



随后对各成像数据进行统计分析，首先取出三种条件下各通道的 beta 值。

test2\_beta <- dir("/Users/eXrld/Dropbox/毕业设计/实验数据/仪器数据/SPM处理结果/test2/", include.dirs = T, recursive = T)   
  
test2\_con\_01 <- paste0("/Users/eXrld/Dropbox/毕业设计/实验数据/仪器数据/SPM处理结果/test2/",test2\_beta[stringr::str\_detect(test2\_beta,pattern = "con\_0001.mat")])  
test2\_con\_02 <- paste0("/Users/eXrld/Dropbox/毕业设计/实验数据/仪器数据/SPM处理结果/test2/",test2\_beta[stringr::str\_detect(test2\_beta,pattern = "con\_0002.mat")])  
test2\_con\_03 <- paste0("/Users/eXrld/Dropbox/毕业设计/实验数据/仪器数据/SPM处理结果/test2/",test2\_beta[stringr::str\_detect(test2\_beta,pattern = "con\_0003.mat")])  
  
test2\_con\_01\_beta <- matrix(0, ncol = length(extract\_fNIRS\_beta(test2\_con\_01[1])), nrow = length(test2\_con\_01))  
test2\_con\_02\_beta <- matrix(0, ncol = length(extract\_fNIRS\_beta(test2\_con\_01[1])), nrow = length(test2\_con\_01))  
test2\_con\_03\_beta <- matrix(0, ncol = length(extract\_fNIRS\_beta(test2\_con\_01[1])), nrow = length(test2\_con\_01))  
  
for(i in 1:length(test2\_con\_01)){  
 test2\_con\_01\_beta[i,] <- extract\_fNIRS\_beta(test2\_con\_01[i])  
}  
  
for(i in 1:length(test2\_con\_01)){  
 test2\_con\_02\_beta[i,] <- extract\_fNIRS\_beta(test2\_con\_02[i])  
}  
  
for(i in 1:length(test2\_con\_01)){  
 test2\_con\_03\_beta[i,] <- extract\_fNIRS\_beta(test2\_con\_03[i])  
}

取出各通道 beta 值后，通过方差分析比较其差异

test2\_con\_01\_beta <- cbind(test2\_con\_01\_beta, rep(1, 11))  
test2\_con\_02\_beta <- cbind(test2\_con\_02\_beta, rep(2, 11))  
test2\_con\_03\_beta <- cbind(test2\_con\_03\_beta, rep(3, 11))  
  
test2\_beta\_table <- data.frame(rbind(test2\_con\_01\_beta, test2\_con\_02\_beta, test2\_con\_03\_beta))  
names(test2\_beta\_table)[46] <- "Group"  
names(test2\_beta\_table)[1:45] <- paste0("Ch", 1:45)  
test2\_aov\_table <- data.frame(matrix(0, ncol = 3, nrow = 45))  
names(test2\_aov\_table) <- c("Ch", "P.value", "Sig.")  
for(i in 1:45){  
test2\_aov\_table$Ch[i] <- i  
test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[,i] ~ Group, data = test2\_beta\_table))[[1]]$`Pr(>F)`  
if(summary(aov(test2\_beta\_table[,i] ~ Group, data = test2\_beta\_table))[[1]]$`Pr(>F)` < 0.05){  
 test2\_aov\_table$Sig.[i] <- "\*\*\*"  
 }  
}

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used

## Warning in test2\_aov\_table$P.value[i] <- summary(aov(test2\_beta\_table[, :  
## number of items to replace is not a multiple of replacement length

## Warning in if (summary(aov(test2\_beta\_table[, i] ~ Group, data =  
## test2\_beta\_table))[[1]]$`Pr(>F)` < : the condition has length > 1 and only  
## the first element will be used