**Практическое занятие №6**

**Анализ нескольких групп ( параметрическая и непараметрическая однофакторная ANOVA)**

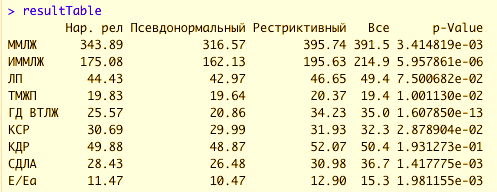
В файле ThreeGroups.RData находятся данные пациентов с нарушением диастолической функции (ДФ) сердечной деятельности:

* Масса миокарда левого желудочка (ММЛЖ),
* Индекс ММЛЖ,
* Левое предсердие (ЛП),
* Толщина межжелудочковой перегородки (ТМЖП),
* Конечно-диастолический размер (КДР) левого желудочка,
* Конечно-систолический размер (КСР) левого желудочка,
* Систолическое давление в легочной артерии (СДЛА),
* Тип диастолической дисфункции (ДД).

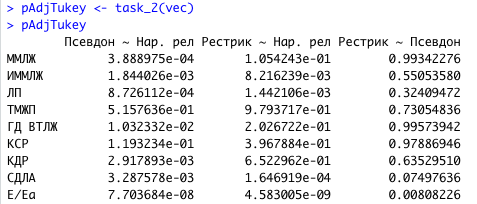
Задание.

Используя методические рекомендации к практическому занятию №5, выполнить следующие действия:

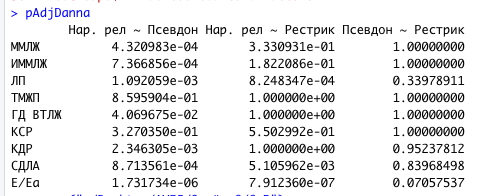
1. Исследовать вид распределения переменных.
2. Заполнить таблицу соответствующими описательными характеристиками групп.
3. Провести сравнительный анализ трех групп. Заполнить последний столбик таблицы и сделать выводы о наличии значимых различий средних.

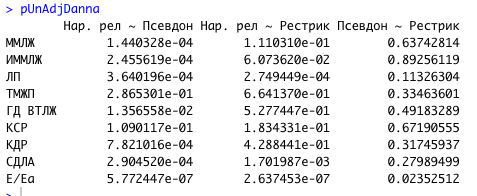


1. Провести множественные сравнения. Для параметрического дисперсионного анализа использовать критерий Тьюки.

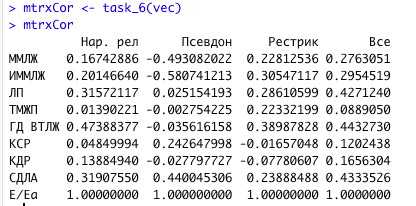


1. Для непараметрического – критерий Данна (самостоятельно найти и установить пакет R, который выполняет тест Данна). Отметить некоторым образом показатели и пары групп, которые различаются при парных сравнениях в таблице.





1. Исследовать корреляцию Е/Еа с остальными показателями (по группам и общую), таблицу для представления результатов разработать самостоятельно.



1. Оформить отчет по результатам анализа, а также предоставить скрипт для выполнения анализа. Описать назначение всех параметров вычислений.

**Приложение А**

vec <- c(4, 5, 6, 7, 9, 10, 11, 12, 13)

# 1-3

task\_1 <- function(v) {

mean\_type <- c()

mean\_full <- c()

p\_value <- c()

for (i in v) {

temp\_mean\_type <- round(tapply(ThreeGroups[, i], ThreeGroups$DD, mean), 2)

temp\_mean\_full <- round(mean(ThreeGroups[, i]), 2)

test\_shapiro <- shapiro.test(ThreeGroups[, i])

mean\_full <- c(mean\_full, temp\_mean\_full)

mean\_type <- rbind(mean\_type, temp\_mean\_type)

p\_value <- c(p\_value, test\_shapiro$p.value)

}

result\_table <- cbind(mean\_full, mean\_type)

result\_table <- cbind(result\_table, p\_value)

colnames(result\_table) <- c(

"Нар. рел", "Псевдон",

"Рестрик", "Все", "p-Value"

)

rownames(result\_table) <- c(

"ММЛЖ", "ИММЛЖ", "ЛП", "ТМЖП",

"ГД ВТЛЖ", "КСР", "КДР", "СДЛА", "Е/Еа"

)

return(result\_table)

}

result\_table <- task\_1(vec)

result\_table

# 4

task\_4 <- function(v) {

p\_adj <- c()

for (i in v) {

test <- aov(ThreeGroups[, i] ~ ThreeGroups$DD, data = ThreeGroups)

tukey\_test <- TukeyHSD(test)

p\_value <- tukey\_test[[1]]

p\_adj <- rbind(p\_adj, p\_value[, 4])

}

colnames(p\_adj) <- c(

"Псевдон ~ Нар. рел",

"Рестрик ~ Нар. рел",

"Рестрик ~ Псевдон"

)

rownames(p\_adj) <- c(

"ММЛЖ", "ИММЛЖ", "ЛП",

"ТМЖП", "ГД ВТЛЖ", "КСР",

"КДР", "СДЛА", "Е/Еа"

)

return(p\_adj)

}

p\_adj\_tukey <- task\_2(vec)

p\_adj\_tukey

# 5

install.packages("FSA")

library("FSA")

task\_5\_1 <- function(v) {

p\_adj\_danna <- c()

for (i in v) {

test <- dunnTest(

ThreeGroups[, i] ~ ThreeGroups$DD,

data = ThreeGroups, method = "bonferroni"

)

p\_adj\_danna <- rbind(p\_adj\_danna, test[[2]]$P.adj)

}

colnames(p\_adj\_danna) <- c(

"Нар. рел ~ Псевдон",

"Нар. рел ~ Рестрик",

"ПсевÍдон ~ Рестрик"

)

rownames(p\_adj\_danna) <- c(

"ММЛЖ", "ИММЛЖ", "ЛП",

"ТМЖП", "ГД ВТЛЖ", "КСР",

"КДР", "СДЛА", "Е/Еа"

)

return(p\_adj\_danna)

}

task\_5\_2 <- function(v) {

p\_un\_adj\_danna <- c()

for (i in v) {

test <- dunnTest(

ThreeGroups[, i] ~ ThreeGroups$DD,

data = ThreeGroups, method = "bonferroni"

)

p\_un\_adj\_danna <- rbind(p\_un\_adj\_danna, test[[2]]$P.unadj)

}

colnames(pUnAdjDanna) <- c(

"Нар. рел ~ Псевдон",

"Нар. рел ~ Рестрик",

"Псевдон ~ Рестрик"

)

rownames(pUnAdjDanna) <- c(

"ММЛЖ", "ИММЛЖ", "ЛП",

"ТМЖП", "ГД ВТЛЖ", "КСР",

"КДР", "СДЛА", "Е/Еа"

)

return(p\_un\_adj\_danna)

}

p\_adj\_danna <- task\_5\_1(vec)

p\_adj\_danna

p\_un\_adj\_danna <- task\_5\_2(vec)

p\_un\_adj\_danna

# 6

task\_6 <- function(v) {

mtrx\_cor <- c()

vec\_nar <- which(ThreeGroups$DD == "нар рел")

vec\_pcev <- which(ThreeGroups$DD == "рестрик")

vec\_rest <- which(ThreeGroups$DD == "псевдон")

for (i in v) {

a <- cor(ThreeGroups[vec\_nar, 13],

ThreeGroups[vecNar, i],

method = "pearson"

)

b <- cor(ThreeGroups[vec\_pcev, 13],

ThreeGroups[vecPcev, i],

method = "pearson"

)

c <- cor(ThreeGroups[vec\_rest, 13],

ThreeGroups[vecRest, i],

method = "pearson"

)

d <- cor(ThreeGroups[, 13],

ThreeGroups[, i],

method = "pearson"

)

mtrx\_cor <- rbind(mtrx\_cor, c(a, b, c, d))

}

colnames(mtrx\_cor) <- c(

"Нар. рел", "Псевдон",

"Рестрик", "Все"

)

rownames(mtrx\_cor) <- c(

"ММЛЖ", "ИММЛЖ", "ЛП",

"ТМЖП", "ГД ВТЛЖ", "КСР",

"КДР", "СДЛА", "Е/Еа"

)

return(mtrx\_cor)

}

mtrx\_cor <- task\_6(vec)

mtrx\_cor