Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования «Кубанский государственный университет»

Кафедра вычислительных технологий

**ОТЧЕТ**

о выполнении лабораторной работы №5

по дисциплине «Обработка больших данных»

Выполнил: ст. гр. 39/1

Кочнев В.Ю.

Проверил: преподаватель

Яхонтов А.А.

Краснодар

2025

Тема: Задачи классификации и кластеризации.

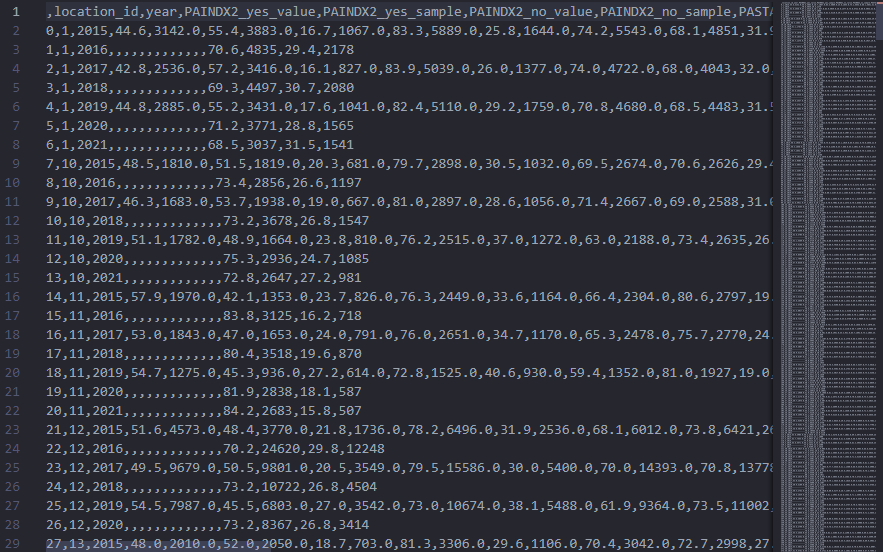
Цель: Закрепить знания, об алгоритмах классификации и кластеризации данных, ознакомиться с некоторыми функциями языка R, осуществляющими этот вид анализа, принципами их работы. Научиться визуализировать результаты работы функций кластерного анализа и классификаторов, интерпретировать полученные результаты.

Задание:

1. Выполнить дескриптивный анализ данных (здесь приветствуются дополнительные исследования).
2. Оценить оптимальное число кластеров, для этого построить диаграмму "Метод силуэта", “Метод локтя”, "Статистику разрыва" и Алгоритм консенсуса.
3. Выполнить иерархическую кластеризацию вашего набора данных, построив **дендрограмму**. Подробно обосновать Ваш выбор числа групп.
4. Построить диаграмму со столбчатыми диаграммами (рис. 5.8) и боксплотами групп (рис. 5.12). Провести сравнительный анализ полученных групп.
5. Выполнить кластеризацию своего датасета по k-means (рис.5.9, 5.10).
6. Выполнить построение scatterplot (рис. 5.13) с помощью функций plot или pairs.
7. Построить трехмерную кластеризацию по scatterplot3d (5.16)
8. В целом: выполнить шаги 1-3,5 анализа для своего набора данных (если какие-то из шагов нерелевантны вашему набору данных, объяснить почему).

Ход работы:

1) Дан датасет, необходимо провести дескриптивный анализ данных



PA <- read.csv("BRFSS Physical Acticity.csv")

location <- read.csv("BRFSS location\_id KEY.csv")

locations\_id <- PA[,2]

locations\_id <- as.numeric(levels(factor(locations\_id)))

locations\_id

dataset\_st <- PA[c(2,4:length(PA))]

# Замена пропущенных данных на среднее значение по каждой локации

dataset <- data.frame()

for (i in 1:length(locations\_id)){

df <- subset(dataset\_st, dataset\_st$location\_id == locations\_id[i])

for (i in (1:length(df))){

df[,i][is.na(df[,i])] = mean(df[,i], na.rm = TRUE)

}

dataset<-rbind(dataset, df)

}

# Оставим только положительные ответы в опросах(положительные минус отрицательные)

dataset

dataset <- dataset[rowSums(is.na(dataset)) == 0,]

dataset <- dataset[2:length(dataset)]

dataset <- dataset[c(1:(length(dataset)/2))\*2]

dataset <- dataset[c(1:(length(dataset)/2))\*2-1] - dataset[c(1:(length(dataset)/2))\*2]

maxsets <- apply(dataset, 2, max, na.rm = TRUE)

minsets <- apply(dataset, 2, min, na.rm = TRUE)

meanSets <- apply(dataset, 2, mean, na.rm = TRUE)

meanSets

# Приближение значений

psichical\_activity <- scale(dataset, center = minsets, scale = maxsets - minsets)

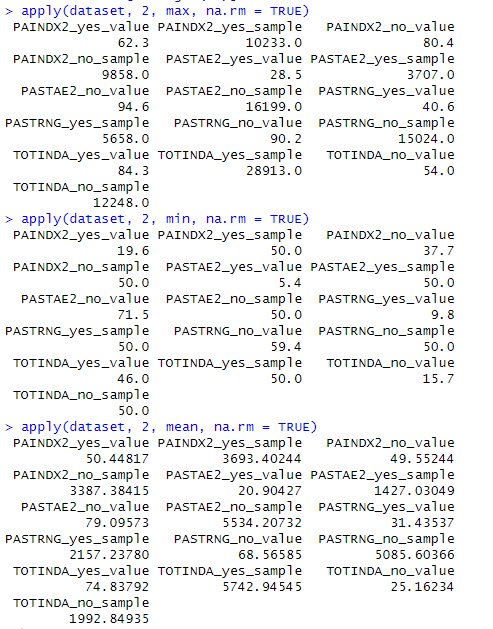
psichical\_activity

# Матрица попарных расстояний

dist.psichical\_activity <- dist(psichical\_activity)

# Проводим кластерный анализ

clust.psichical\_activity <- hclust(dist.psichical\_activity, "ward.D")



# Замена пропущенных данных на среднее значение

for (i in (1:length(dataset))){

dataset[,i][is.na(dataset[,i])] = mean(dataset[,i], na.rm = TRUE)

}

# Приближение значений

psichical\_activity <- scale(dataset, center = minsets, scale = maxsets - minsets)

psichical\_activity

# Матрица попарных расстояний

dist.psichical\_activity <- dist(psichical\_activity)

# Проводим кластерный анализ

clust.psichical\_activity <- hclust(dist.psichical\_activity, "ward.D")

2) Оцениваю оптимальное кол-во кластеров

— Метод силуэта

Метод Silhouette измеряет качество кластеризации и определяет, насколько хорошо каждая точка лежит в пределах своего кластера. Число кластеров *k* в этом случае определяется по средней ширине силуэта каждого кластера (average silhouette width), а точнее по отношению:

*si*=(*b*(*i*)−*a*(*i*))/max[*b*(*i*),*a*(*i*)],

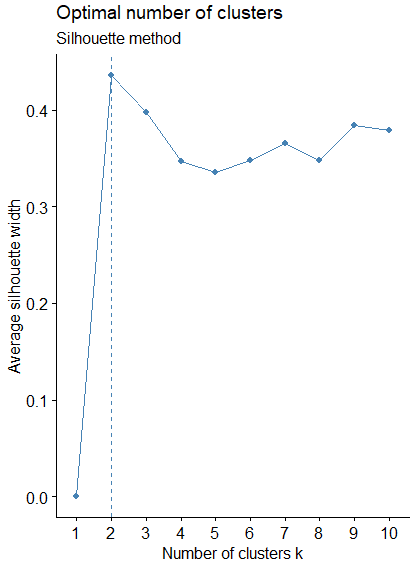
где *a*(*i*) - среднее расстояние между объектами *i*-го кластера, *b*(*i*) - среднее расстояние от объектов *i*-го кластера до другого кластера, самого близкого к *i*-му.

library("factoextra")

help(fviz\_nbclust)

fviz\_nbclust(psichical\_activity, kmeans, method = "silhouette")+

labs(subtitle = "Silhouette method")



Оптимальное кол-во кластеров 2

— Метод локтя

Метод Локтя рассматривает общую сумму квадратов внутри кластера (WSS) как функцию количества кластеров.

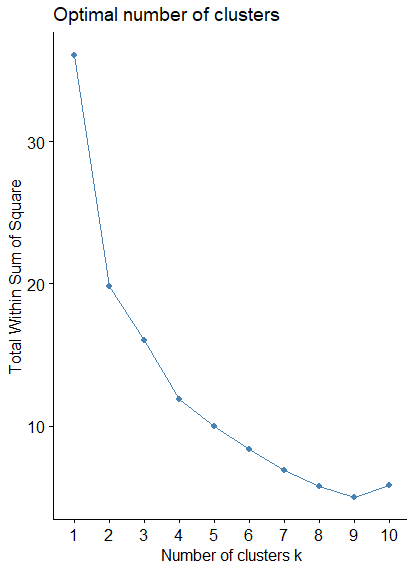
Метод основан на минимизации внутригруппового разброса *Wtotal*. Внутригрупповой разброс *Wtotal*=∑*kW*(*Ck*)→min.

# Метод локтя

library("cluster")

fviz\_nbclust(psichical\_activity, kmeans, method = "wss")+

labs(subtitle = "Locktya method")



Оптимальное кол-во кластеров 4

— Статистика разрыва

Статистика разрыва сравнивает общую внутрикластерную дисперсию для разных значений k с их ожидаемыми значениями для распределения без кластеризации.

Этот метод генерируются на основе ресэмплинга и имитационных процедур Монте-Карло. Пусть *E*∗*n*{log(*W*∗*k*)} обозначает оценку средней дисперсии *W*∗*k*, полученной бутстреп-методом, когда *k* кластеров образованы случайными наборами объектов из исходной выборки размером *n.*

Тогда статистика

определяет отклонение наблюдаемой дисперсии *Wk*



от ее ожидаемой величины при справедливости нулевой гипотезы о том, что исходные данные образуют только один кластер.

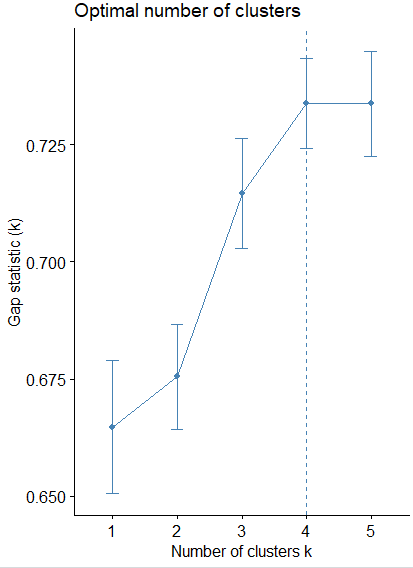
При сравнительном анализе последовательности значений *Gapn*(*k*),*k*=2,…,*Kmax*

наибольшее значение статистики соответствует наиболее полезной группировке, дисперсия которой максимально меньше внутригрупповой дисперсии кластеров, собранных из случайных объектов исходной выборки:

# Метод статистика разрыва

gap\_stat <- clusGap(psichical\_activity, FUN = kmeans, nstart = 5, K.max =5, B = 5)

fviz\_gap\_stat(gap\_stat)



Оптимальное кол-во кластеров 1

— Алгоритм на основе консенсуса

# Алгоритм консенсуса

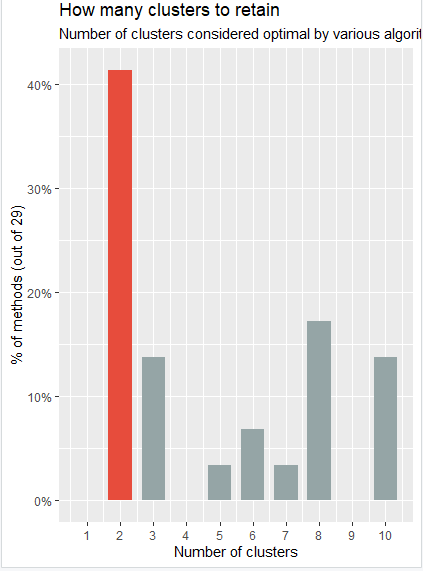
library ("parameters")

n\_clust <- n\_clusters(dataset,

package = c("easystats", "nbClust", "mclust"),

standardize = FALSE)

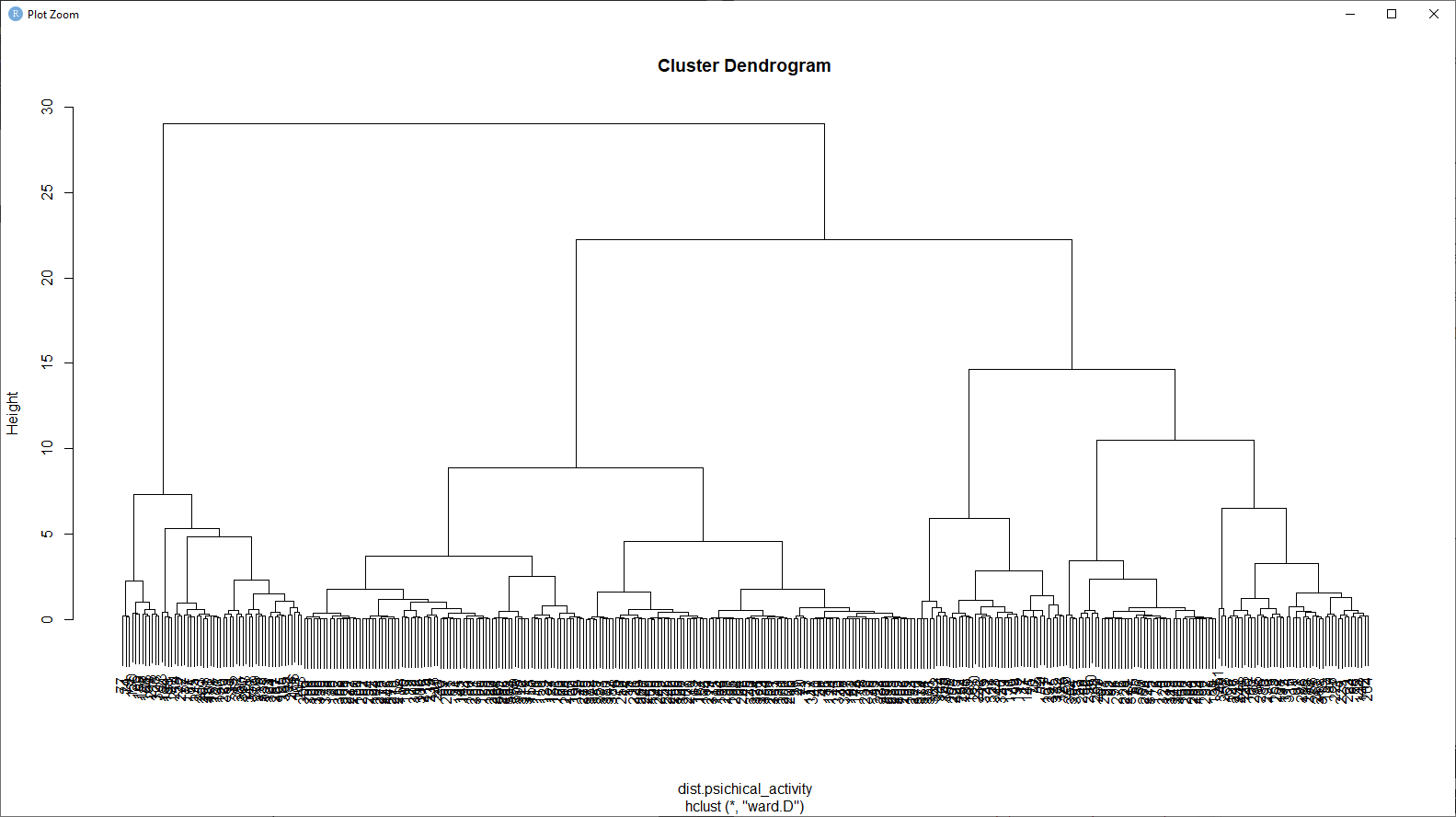
plot(n\_clust)



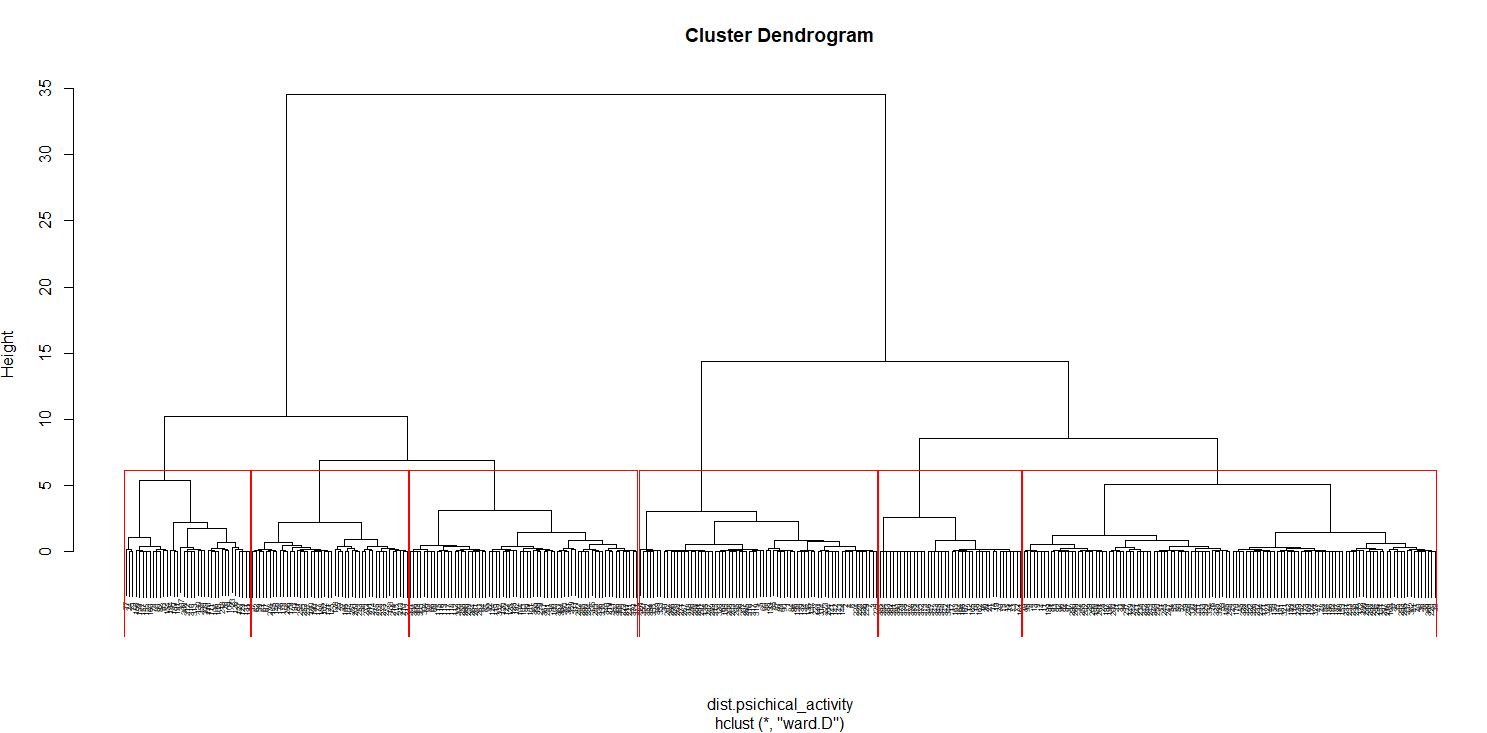
Оптимальное кол-во кластеров 2

3) Выполнить кластеризацию построив дендраграмму

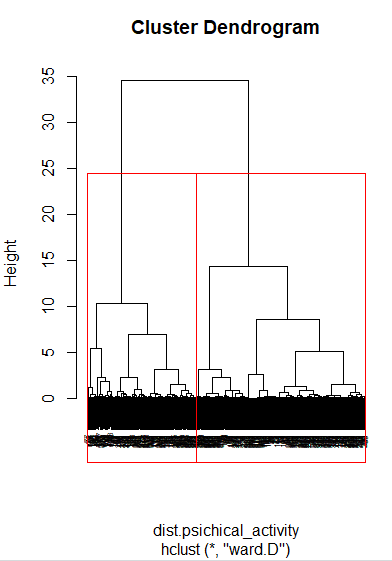
Обычная дендраграмма



Разделим её на 6 кластеров



Разделим на выдавшие нам методами кол-во кластеров (2)



3) Столбчатая диаграмма и боксплот

# Разделение на группы

groups <- cutree(clust.psichical\_activity, k =2)

g1 <- colMeans(psichical\_activity[groups == 1, 1:4])

g2 <- colMeans(psichical\_activity[groups == 2, 1:4])

# Постройка столбчатой диаграммы

df <- data.frame(g1,g2)

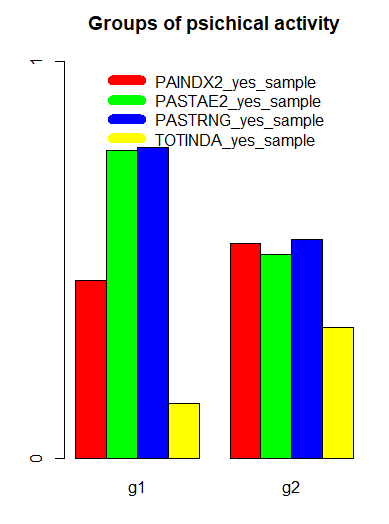
df1 <- t(df)

df <- t(df1)

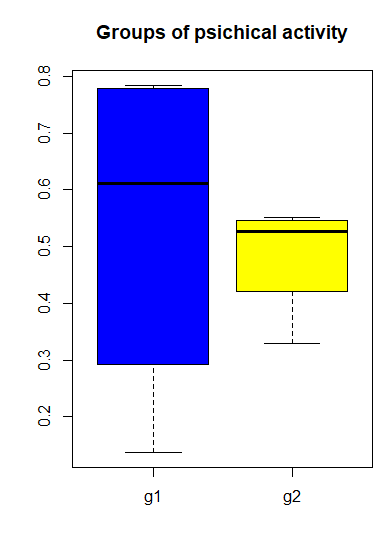
barplot(df,ylim=c(0,1), col=c("red","green","blue","yellow"), beside = TRUE, axes = FALSE, main = "Groups of psichical activity")

axis(2, at = 0:2, labels = 0:2)

legend("top", legend = rownames(df), col=c("red","green","blue","yellow"), lwd=10, bty = "n")



boxplot(df, main = "Groups of psichical activity", col=c("blue","yellow"))

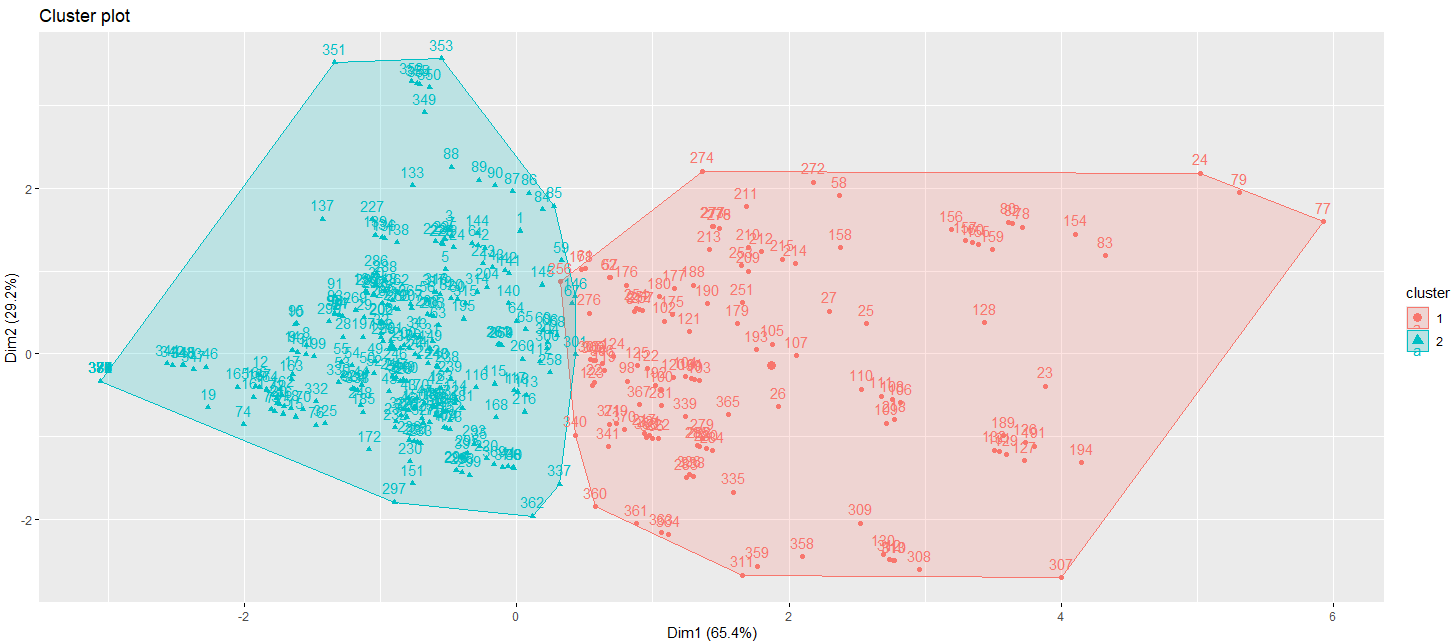


5)

# Кластеризация k-means

km.res <- kmeans(psichical\_activity, 2, nstart =10)

fviz\_cluster(km.res, psichical\_activity)



6)

pairs(psichical\_activity, main = "psichical activity",pch = 19, cex = 0.8, col = c("#00AFBB", "#E7B800", "#FC4E07"))



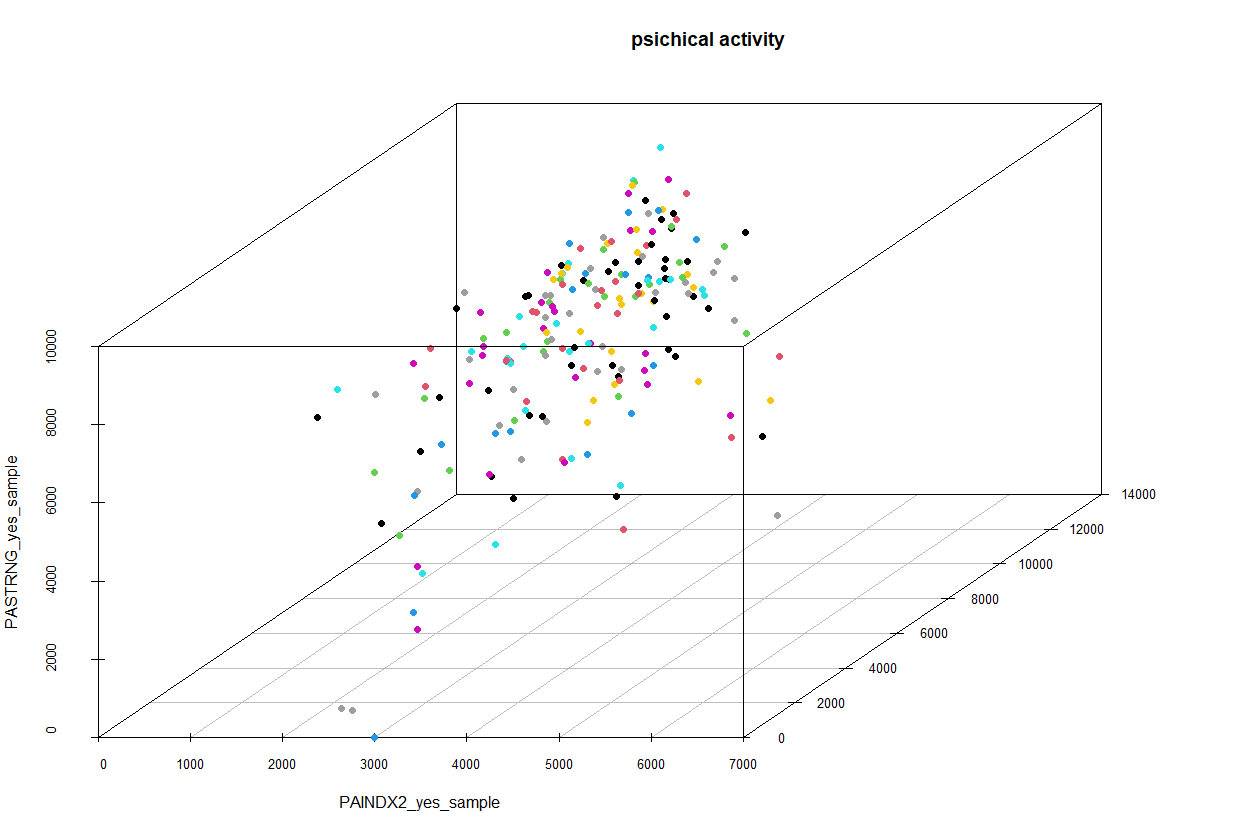
7)

colors <- c("#00AFBB", "#E7B800", "#FC4E07")

scatterplot3d(dataset, main= "psichical activity",pch = 16, color=colors)

legend(s3d$xyz.convert(7.5, 3, 4.5), legend = c(1:count\_group),

col = colors, pch = 16)



**Часть 2**

1) Классификация с помощью наивного Байесовского классификатора

# Наивный Байесовский подход

library("klaR")

my\_data <- dataset

answers <- factor(groups)

my\_data <- data.frame(my\_data, answers)

train.ind <- sample(1:nrow(dataset), ceiling(nrow(dataset)\*2/3), replace=FALSE)

train.ind

train\_data <- my\_data[train.ind,]

train\_data

naive\_phis <- NaiveBayes(train\_data$answers ~ ., data = train\_data)

naive\_phis$tables

layout(matrix(c(1,2,3,4), 2, 2, byrow = TRUE))

plot(naive\_phis,lwd = 2, legendplot=TRUE)

layout(1)

test\_data <- my\_data[-train.ind,]

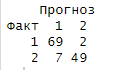
test\_data <- test\_data[,-ncol(test\_data)]

test\_data

answer\_test <- my\_data[-train.ind,]$answers

pred <- predict(naive\_phis, test\_data)

table(Факт = answer\_test, Прогноз = pred$class)



2) Анализ точности

Acc <- mean(pred$class == answer\_test)

Acc



92%

3) Классификация с помощью дерева решений

# Деревья решений

library("party")

set.seed(1234)

nrow(train\_data)

nrow(test\_data)

nrow(my\_data)

colnames(dataset)

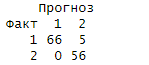
myFormula <- answers ~ PAINDX2\_yes\_sample + PASTAE2\_yes\_sample + PASTRNG\_yes\_sample + TOTINDA\_yes\_sample

ctree\_phis <- ctree(myFormula, data = train\_data)

pred\_ctree <- predict(ctree\_phis, test\_data)

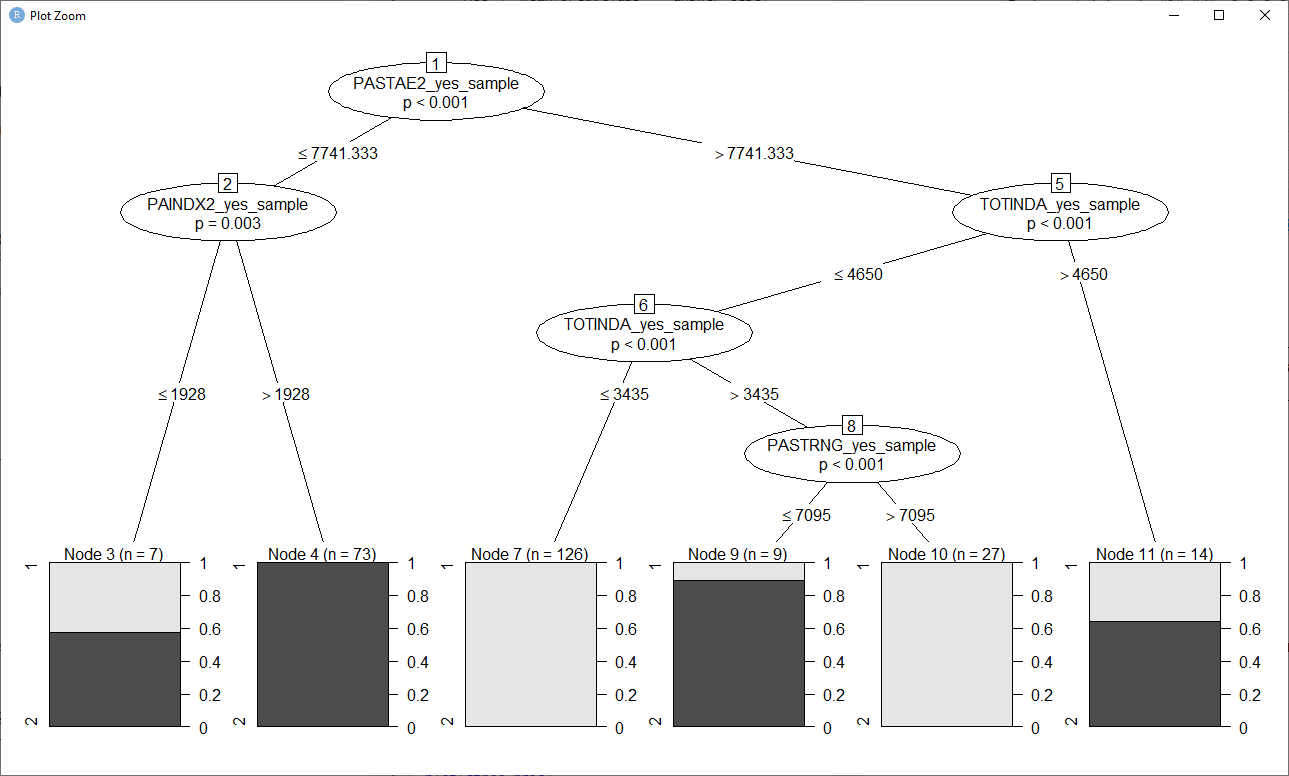
pred\_ctree

table(Факт = answer\_test, Прогноз = pred\_ctree)



4) График дерева решений

plot(ctree\_phis)



5) Анализ точности



96%

Точность дерева решений выше чем у наивного Байесовского классификатора

6) Классификация с помощью алгоритма случайного леса

# Случайный лес

# install.packages("randomForest")

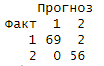
library("randomForest")

# 1

rf <- randomForest(train\_data$answers ~. , data = train\_data, ntree = 100, proximity= TRUE)

pred\_rf <- predict(rf, test\_data)

table(Факт = answer\_test, Прогноз = pred\_rf)



98%

Точность случайного леса выше чем у классификатора деревом решений на 2%

7) Сравнение результатов случайного леса и Байесовского классификатора

Случайный лес выдал на 6% больше верных решений чем Байесовский классификатор

Вывод: В ходе лабораторной работы я узнал подробно об алгоритмах классификации и кластеризации данных, ознакомился с некоторыми функциями языка R, осуществляющими этот вид анализа, принципами их работы. Научился визуализировать результаты работы функций кластерного анализа и классификаторов, интерпретировал полученные результаты. Различные методы кластерного анализа дают очень разные решения, поэтому лучше пользоваться тем что собирают данные из нескольких кластеризаторов и выдаёт наиболее вероятное кол-во кластеров.