Задача : Создать библиотеку для статистической обработки данных.

Есть бибилиотека (cmp\_function.r), в ней описаны все функции, которые необходимы для обработки, только для двух групп. В примере (example\_cmp\_function.r) представлена таблица mtcars, которая разбивается на 2 групп машин, по 9 показателю(трансмиссия – автоматическая и мануальная). И дальше идет сравнение этих 2 групп по остальным показателям. Таблица 1 : столбец A – показатель, по которому идет сравнение

Столбец B – количество машин в 1 группе

Столбец С – видимо какое-то среднее значение или медиана из всех содержащихся в группе машин + доверительный интервал

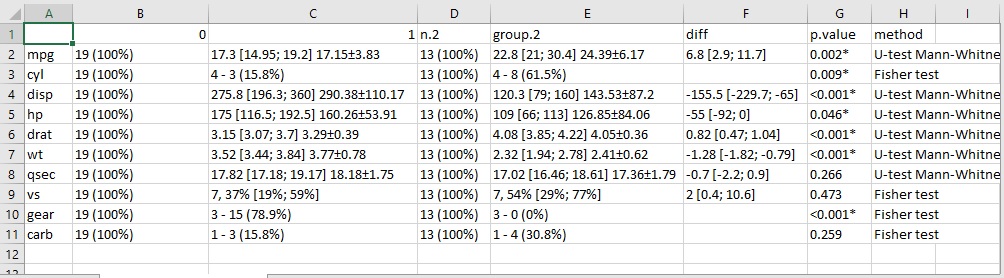
Столбец D – количество машин в группе 2

Столбец E - видимо какое-то среднее значение или медиана из всех содержащихся в группе машин и доверительный интервал (не точно)

Столбец F – отношение шансов

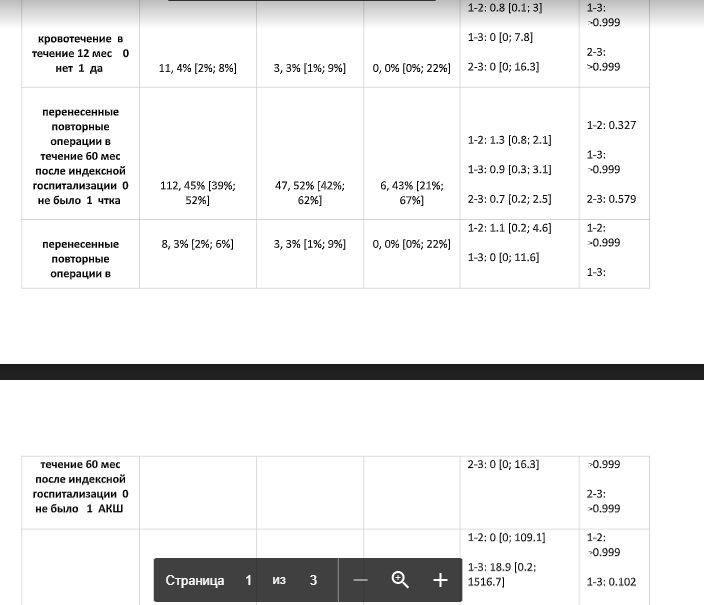
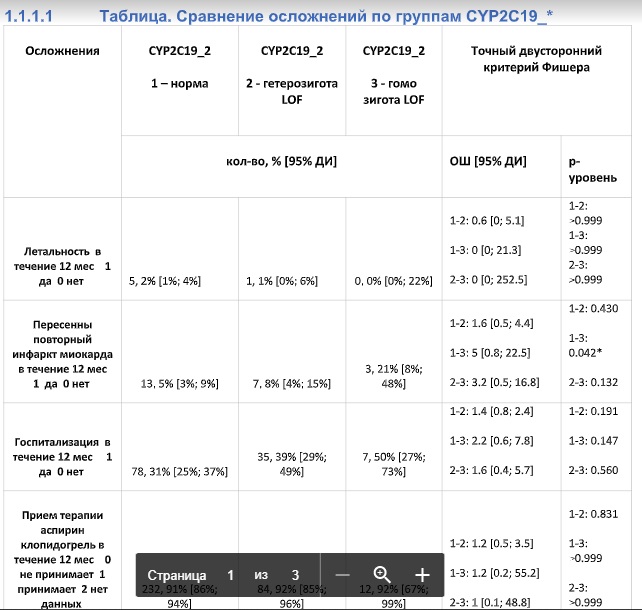
Столбец G – p-уровень – главный показатель в данной задаче. Там, где \* - статистически значимая разница в показателях групп. Т.е. Влияние трансмиссий на каждый показатель со \* оказывает важную роль, т.е. показатель точно зависит от вида трансмиссии.

Столбец H – метод сравнения(в итоговой программе не обязателен)



Задача сделать сравнение не для двух групп, а для трех.

Т.е. еще столбец 3 группы, столбец сравнения для 1-2 1-3 2-3 , можно и с выводом в excel файле, можно и в файл docx, в таком виде это должно быть:



<https://docs.google.com/viewerng/viewer?url=https://trello-attachments.s3.amazonaws.com/5e54b349dd992f56b0a692fb/5e8ffc36e5fb5325aa225c9c/d29c59096e60ba2a0b3496695e291d9f/%25D0%259F%25D1%2580%25D0%25B8%25D0%25BC%25D0%25B5%25D1%2580_%25D1%2581%25D1%2580%25D0%25B0%25D0%25B2%25D0%25BD%25D0%25B5%25D0%25BD%25D0%25B8%25D1%258F_3_%25D0%25B3%25D1%2580%25D1%2583%25D0%25BF%25D0%25BF.docx> таблица полностью

Есть еще такой код, который должен помочь

# 6

# сравнение динамики осложнений во всех подгруппах групп В, C, D

# Таблицы

```{r}

#

# ----- COMPARE MANY GROUPS IN ONE TIME POINT -----------------------

#

#

# ----- Step 1. Create list of comparing groups -----------------------

#

cmp.groups = list( 'CYP2C19\_2' = list(col = 'CYP2C19\_2'),

'CYP2C19\_2\_LOF' = list(col = '2+3 vs. 1: CYP2C19\_2, 1 == 0, 2+3 == 1'),

'CYP2C19\_3' = list(col = 'CYP2C19\_3'),

'CYP2C19\_3\_LOF' = list(col = '2+3 vs. 1: CYP2C19\_3, 1 == 0, 2+3 == 1'),

'CYP2C19\_17' = list(col = 'CYP2C19\_17'),

'CYP2C19\_17\_LOF' = list(col = '2+3 vs. 1: CYP2C19\_17, 1 == 0, 2+3 == 1'),

'CYP2C19\_2\_3\_norm\_vs\_LOF' = list(col = '@ 2+3 vs. 1: CYP2C19\_2 OR CYP2C19\_3'),

'CYP2C19\_2\_17\_norm\_vs\_LOF' = list(col = '@ 2+3 vs. 1: CYP2C19\_2 OR CYP2C19\_17'),

'CYP2C19\_3\_17\_norm\_vs\_LOF' = list(col = '@ 2+3 vs. 1: CYP2C19\_3 OR CYP2C19\_17'),

'CYP2C19\_2\_3\_norm\_vs\_gomo\_LOF' = list(col = '@ 3 vs. 1: CYP2C19\_2 OR CYP2C19\_3'),

'CYP2C19\_2\_17\_norm\_vs\_gomo\_LOF' = list(col = '@ 3 vs. 1: CYP2C19\_2 OR CYP2C19\_17'),

'CYP2C19\_3\_17\_norm\_vs\_gomo\_LOF' = list(col = '@ 3 vs. 1: CYP2C19\_3 OR CYP2C19\_17')

)

#

# ----- Step 1. Create list of comparing groups -----------------------

#

#

# ----- Step 2. Compare complications in groups -----------------------

#

ls = list(list(id = 33, type = 'bin'))

ls = append.descr(ls, ids = c(34,36,38,39,42,44,79,81,85,86), type = 'bin')

ls = append.descr(ls, ids = c(49,51,53,55,80,82), type = 'bin')

#

# ----- Tukey test -----

#

res.table = NULL

for( groups in names(cmp.groups)){

data.temp = data.Proc

id.col = cmp.groups[[groups]][['col']]

data.temp[[id.col]] = as.character(data.temp[[id.col]])

t.points = as.character(sort(unique(data.temp[[id.col]])))

len.t.points = length(t.points)

for( i in 1:(len.t.points-1)){

for(j in (i+1):len.t.points){

cat("\n")

cat("dimension:")

cat(dim(data.temp))

cat("\n")

group.No.1 = t.points[i]

group.No.2 = t.points[j]

cat( group)

cat(": ")

cat(group.No.1)

cat(" ")

cat(group.No.2)

cat("\n")

#group.No.1 = as.character(unique(data.temp$t.point)[i])

#group.No.2 = as.character(unique(data.temp$t.point)[j])

F.NAME = paste0(

as.character(groups),

"\_\_",

gsub('[\\. ]','\_', group.No.1),

"\_",

gsub('[\\. ]','\_', group.No.2)

)

#debug(cmp.2group.all.vars)

res.cmp = cmp.2group.all.vars(id.group = id.col,

group.labels = c(group.No.1, group.No.2),

columns.list = ls,

data = data.temp,

PATH = path.dir.out,

FILENAME = F.NAME,

#paired = F,

result = NULL

)

res.cmp = cbind(names = rownames(res.cmp),res.cmp)

res.cmp = res.cmp %>%

select(1,2,3,4,5,6,7) %>%

rename\_at(c(1,2,3,4,5,6,7),

~c('names',

paste0('n (%): ',group.No.1),

paste0('values: ',group.No.1),

paste0('n (%): ',group.No.2),

paste0('values: ',group.No.2),

paste0('difference: ', group.No.1, '-',group.No.2),

paste0('p-value: ', group.No.1, '-', group.No.2)

)

)

if( (i == 1) & (j == 2)){

res.table = res.cmp

} else{

res.table = left\_join(res.table, res.cmp,by='names')

}

}

}

p.table = res.table %>%

select(starts\_with('p-value: '))

p.common = NULL

for (ii in 1:dim(p.table)[2] )

{

xx = paste0(gsub('p-value: ','',colnames(p.table)[ii]),

': ',

p.table[[ii]],

'\n'

)

if(ii == 1){

p.common = xx

}else{

p.common = paste0(p.common, xx)

}

}

diff.table = res.table %>%

select(starts\_with('difference: '))

diff.common = NULL

for (ii in 1:dim(diff.table)[2] )

{

xx = paste0(gsub('difference: ','',colnames(diff.table)[ii]),

': ',

diff.table[[ii]],

'\n'

)

if(ii == 1){

diff.common = xx

}else{

diff.common = paste0(diff.common, xx)

}

}

res.table = cbind(res.table, diff.common, p.common)

F.NAME = paste0( path.dir.out,

'COMMON\_TABLE\_',

gsub('\\. ', '\_', as.character(groups)),

'.csv'

)

cat(F.NAME)

res.table %>% write.csv2(file=F.NAME)

}

#

# ----- COMPARE MANY GROUPS IN ONE TIME POINT -----------------------

#

Подсказка от препода :

Разница будет в group.labels: для сравнения 2 групп это 2 метки, для многих групп - больше 2 меток.