Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №4

з дисципліни МОПЕ

на тему:

# «Проведення трьохфакторного експерименту

# при використанні рівняння регресії з урахуванням ефекту взаємодії.»

Виконав:

студент групи ІВ-82

Захарчук Д. С.

Залікова книжка № 8210

Варіант: 209

Перевірив:

Регіда П.Г.

Київ 2020



**Код програми:**

library(dplyr)

library(gsubfn)

# Main functions

inverse <- function(f, lower = -100, upper = 100) {

function(y) {

uniroot(function(x) {

f(x) - y

}, lower = lower, upper = upper)[1]

}

}

get.Gkr <- function(prob, f1, f2) {

isf <- inverse(function(x) 1 - pf(x, f1, (f2 - 1) \* f1))

f\_crit <- isf((1 - prob) / f2)$root

Gkr <- f\_crit / (f\_crit + f2 - 1)

}

get.X\_ranges <- function(X\_names) {

X\_ranges <- c(X1\_min, X1\_max, X2\_min, X2\_max, X3\_min, X3\_max) %>%

array(dim = c(2, 3),

dimnames = list(c('min', 'max'), X\_names)) %>%

t()

}

get.Y\_range <- function(X\_ranges) {

X\_max\_mean <- mean(X\_ranges[, 2])

X\_min\_mean <- mean(X\_ranges[, 1])

Y\_max <- 200 + X\_max\_mean

Y\_min <- 200 + X\_min\_mean

c(Y\_min, Y\_max)

}

get.X\_norms <- function(X\_names, interaction = F) {

if (! interaction) {

X1\_norm <- c(-1, -1, 1, 1)

X2\_norm <- c(-1, 1, -1, 1)

X3\_norm <- c(-1, 1, 1, -1)

} else {

X1\_norm <- c(-1, -1, -1, -1, 1, 1, 1, 1)

X2\_norm <- c(-1, -1, 1, 1, -1, -1, 1, 1)

X3\_norm <- c(-1, 1, -1, 1, -1, 1, -1, 1)

}

N <- length(X1\_norm)

X\_norm <- c(X1\_norm, X2\_norm, X3\_norm) %>%

array(dim = c(N, 3), dimnames = list(c(), X\_names)) %>%

t()

}

get.X\_abs <- function(X\_norm, X\_ranges, k, N, X\_names) {

X\_abs <- sapply(1:k, function(i) {

sapply(X\_norm[i,], function(elem) {

ifelse(elem == 1, X\_ranges[i, 2], X\_ranges[i, 1])

})

}) %>% array(dim = c(N, 3), dimnames = list(c(), X\_names)) %>% t()

return(X\_abs)

}

get.Y\_exp <- function(m, N, Y\_min, Y\_max) {

Y\_names <- paste('Y', 1:m, sep='')

Y\_exp <- runif(n = m \* N, min = Y\_min, max = Y\_max) %>%

matrix(nrow = N, ncol = m, dimnames = list(c(), Y\_names))

return(Y\_exp)

}

get.df <- function(X, Y, interaction) {

l1 <- length(Y[1, ])

matrix <- cbind(t(X), Y)

df <- as.data.frame(matrix)

l2 <- length(df[1, ])

if (interaction) {

df$X1\_X2 <- df$X1 \* df$X2

df$X1\_X3 <- df$X1 \* df$X3

df$X2\_X3 <- df$X2 \* df$X3

df$X1\_X2\_X3 <- df$X1 \* df$X2 \* df$X3

df <- df[, c('X1', 'X2', 'X3', 'X1\_X2',

'X1\_X3', 'X2\_X3', 'X1\_X2\_X3',

'Y1', 'Y2', 'Y3')]

}

df$Y\_means <- rowMeans(matrix[, (l2-l1+1):l2])

df

}

get.lm <- function(df, interaction = F) {

ifelse(interaction,

return(lm(Y\_means ~ X1 \* X2 \* X3, data = df)),

return(lm(Y\_means ~ X1 + X2 + X3, data = df)))

}

check.lm <- function(lm, Y, newdata) {

preds <- predict(lm, newdata)

result <- data.frame(True = Y, Predicted = preds)

}

check.Cohren <- function(df, Y, m, N, prob) {

f1 <- m - 1

f2 <- N

Gkr <- get.Gkr(prob, f1, f2)

Y\_vars <- diag(var(cbind(df$Y\_means, Y)))

Gp <- max(Y\_vars) / sum(Y\_vars)

c(ifelse(Gkr > Gp, T, F), Y\_vars)

}

check.Student <- function(m, N, Y\_vars, Y\_means, X) {

f3 <- (m - 1) \* N

t\_krit <- qt(0.975, df = f3)

S2b <- mean(Y\_vars) / (N \* m)

Sb <- sqrt(S2b)

b <- rep(1, N)

betas <- sapply(1:N, function(i) rowMeans(Y\_means \* cbind(b, X)[i,]))

tp <- abs(betas) / Sb

d <- sum(ifelse(t\_krit < tp, 1, 0))

print('Number of significant coefs:')

print(d)

significance <- tp > t\_krit

print(significance)

list(significance, d, S2b)

}

check.Fisher <- function(m, N, d, Y, preds, S2b) {

f3 <- (m - 1) \* N

f4 <- N - d + 1

S2ad <- (m / f4) \* sum((preds - Y) ^ 2)

Fp <- S2ad / S2b

print('Fp:')

print(Fp)

f\_krit <- qf(0.95, df1 = f4, df2 = f3)

print('Fkr:')

print(f\_krit)

ifelse(Fp < f\_krit, T, F)

}

check.Gohren <- function(

X\_norm, X\_abs, Y\_exp, X\_names, interaction, m, N, prob, Y\_min, Y\_max

) {

repeat {

print('Regression for norm values:')

list[df\_norm\_data, norm\_regr] <- run.regression(X\_norm, Y\_exp, X\_names, interaction)

print('Regression for absolute values:')

list[df\_data, abs\_regr] <- run.regression(X\_abs, Y\_exp, X\_names, interaction)

# Cohren's criteria

print('Cohren\'s criteria:')

list[cohren.criteria, Y\_vars] <- check.Cohren(df\_data, Y\_exp, m, N, prob)

if (cohren.criteria) {

return(list(df\_data, abs\_regr, norm\_regr, Y\_vars, Y\_exp, m))

} else {

print('Unstable variances! Change m = m + 1')

m <- m + 1

Y\_exp <- cbind(Y\_exp, runif(n = N, min = Y\_min, max = Y\_max))

dimnames(Y\_exp) <- list(paste('Y', 1:length(Y\_exp[, 1]), sep=''))

}

}

}

run.regression <- function(X, Y, X\_names, interaction) {

# Regression:

df <- get.df(X, Y, interaction)

regression <- get.lm(df, interaction)

print(regression)

# Let's put Xs:

results <- check.lm(regression, df$Y\_means, df[X\_names])

print('Regression results:')

print(results)

list(df, regression)

}

run.experiment <- function(

k = 3, m = 3, prob = 0.95, interaction = F

) {

X\_names <- c('X1', 'X2', 'X3')

X\_ranges <- get.X\_ranges(X\_names)

print('X\_ranges:')

print(X\_ranges)

Y\_range <- get.Y\_range(X\_ranges)

Y\_min <- Y\_range[1]

Y\_max <- Y\_range[2]

print('Y\_range:')

print(Y\_range)

X\_norm <- get.X\_norms(X\_names, interaction)

print('X\_norm:')

print(X\_norm)

N <- length(X\_norm[1, ])

X\_abs <- get.X\_abs(X\_norm, X\_ranges, k, N, X\_names)

print('X\_abs:')

print(X\_abs)

Y\_exp <- get.Y\_exp(m, N, Y\_min, Y\_max)

print('Y\_exp:')

print(Y\_exp)

list[df\_data, abs\_regr, norm\_regr, Y\_vars, Y\_exp, m] <- check.Gohren(

X\_norm, X\_abs, Y\_exp, X\_names, interaction, m, N, prob, Y\_min, Y\_max

)

print('Stable variances!')

print('Student criteria:')

list[significance, d, S2b] <- check.Student(

m, N, Y\_vars, df\_data$Y\_means,

df\_data[, !names(df\_data) %in% c('Y1', 'Y2', 'Y3', 'Y\_means')])

# Deleting unsignificant coefs

abs\_regr$coefficients <- abs\_regr$coefficients \* significance

print('After deleting unsignificant coefs:')

print(abs\_regr)

check.lm(abs\_regr, df\_data$Y\_means, df\_data[X\_names])

print('Fisher criteria:')

preds <- predict(abs\_regr, df\_data[X\_names])

f.criteria <- check.Fisher(m, N, d, df\_data$Y\_means, preds, S2b)

if (isTRUE(f.criteria)) {

print('Regression is adequate to original function!')

return(T)

} else {

print('Regression is unadequate to original function!')

return(F)

}

}

# Variant 209

X1\_min <- -30

X1\_max <- 0

X2\_min <- 10

X2\_max <- 60

X3\_min <- 10

X3\_max <- 35

# Let's run experiments

num\_tries <- 1

# Without interaction effect

run.experiment()

# With interacti on effect

for (try in 1:num\_tries) {

run.experiment(interaction = T)

}