**Задача 1**

Начальные условия  
n = 160

A0 = 3

A1 = 1

A2 = -6

B0 = 5

B1 = 7

B2 = -2

Перед выполнением задания создадим ряды значений и введём начальные параметры

n <- 160

a0 <- 3

a1 <- 1

a2 <- -6

b0 <- 5

b1 <- 7

b2 <- -2

eps <- rnorm(n, 0, 4)

psi <- rnorm(n, 0, 9)

x <- a0 + a1 \* eps + a2 \* psi

y <- b0 + b1 \* eps + b2 \* psi

1а) Теперь найдём матожидание, дисперсию смещённую и не смещённую и среднеквадратичное отклонение и другие дополнительные статистические параметры

# матожидание

mean(x)

mean(y)

# дисперсия несмещенная

var(x)

var(y)

# дисперсия смещенная

var(x) \* (n - 1) / (n)

var(y) \* (n - 1) / (n)

# среднеквадратическое отклонения

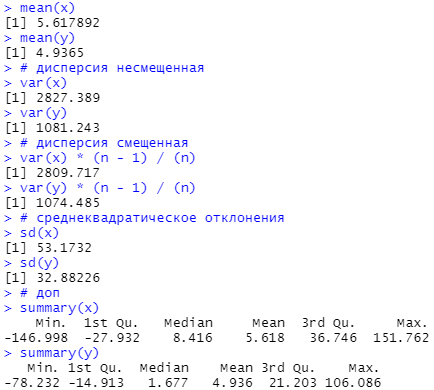
sd(x)

sd(y)

# доп

summary(x)

summary(y)



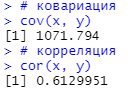
1б) Теперь найдём ковариацию и корреляцию переменных

# ковариация

cov(x, y)

# корреляция

cor(x, y)



1в) Теперь найдём выборочное среднее, дисперсию и среднеквадратичное отклонение

# mean\_x = 3 + 1 \* 0 + -6 \* 0 = 3

# mean\_y = 5 + 7 \* 0 + -2 \* 0 = 5

# std\_x = sqrt(4\*std(eps)^2 + 9 \* std(psi)^2)

# std\_y = sqrt(4\*std(eps)^2 + 9 \* std(psi)^2)

# var\_x = std\_x^2

# var\_y = std\_y^2

1г) Далее построим 95% доверительный интервал

temp\_data <- cor.test(x, y)

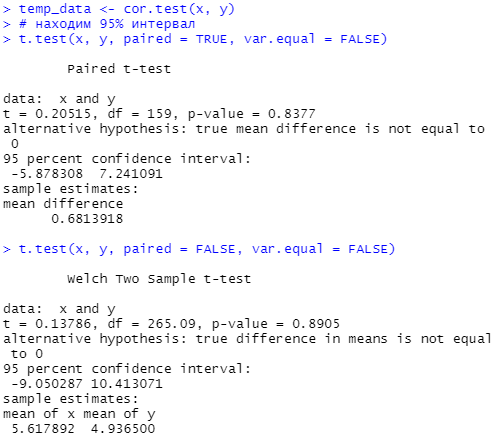
# находим 95% интервал

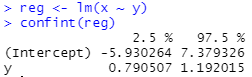
t.test(x, y, paired = TRUE, var.equal = FALSE)

t.test(x, y, paired = FALSE, var.equal = FALSE)

reg <- lm(x ~ y)

confint(reg)





На скриншоте видны доверительные интервалы

**Задание 2**

Начальное условие XRP, USDT

Считываем наши данные, проверяем их и строим две новых таблицы с необходимыми данными

df <- read.csv("C:\\Users\\a.khorunzhenko\\Downloads\\crypto.csv", sep =",")

View(df)

dfus <- data.frame(date = df['date'], x= df['USDT'] )

View(dfus)

dfxr <- data.frame(date = df['date'], x= df['XRP'] )

2a) Вычисляем два ряда доходностей в процентах, положительность и отрицательность доходности и вычисляем кол-во знаков

plus\_dfus <- c()

sign\_dfus <- c()

plus\_dfxr <- c()

sign\_dfxr <- c()

counter <- 0

counter\_plus\_one <- 0

for(t in 1:495){

  if(t>1){

    plus\_dfus[t] = (dfus$USDT[t] - dfus$USDT[t-1]) / dfus$USDT[t]

    if(plus\_dfus[t]>0){

      sign\_dfus[t] = TRUE

    }

    else {

      sign\_dfus[t] = FALSE

    }

    plus\_dfxr[t] = (dfxr$XRP[t] - dfxr$XRP[t-1]) / dfxr$XRP[t]

    if(plus\_dfxr[t]>0){

      sign\_dfxr[t] = TRUE

    }

    else {

      sign\_dfxr[t] = FALSE

    }

  }

  else{

    plus\_dfus[t] = 0

    plus\_dfxr[t] = 0

    sign\_dfxr[t] = FALSE

    sign\_dfus[t] = FALSE

  }

  if(sign\_dfxr[t] == sign\_dfus[t]){

    counter <- counter + 1

  }

  if(sign\_dfxr[t] == FALSE && sign\_dfus[t] == TRUE){

    counter\_plus\_one <- counter\_plus\_one + 1

  }

}

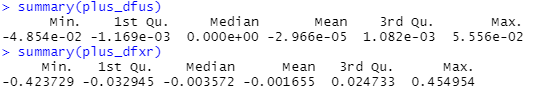
Каунтер получаем равным 239. В процентном соотношении получаем 239/495.

Каунтер для положительных получаем 104. В процентном соотношении получаем 104/495.

Теперь продемонстрируем, что распределения доходностей имеют толстые хвосты

summary(plus\_dfus)

summary(plus\_dfxr)



Далее строим регрессию доходности первой валюты от знака второй валюты и получаем значение

install.packages("lmtest")

library(lmtest)

install.packages("sandwich")

library(sandwich)

lmtest::bptest(reg)

lmtest::coefci(reg, vcov = sandwich::vcovHC)



##

psych::describe(plus\_dfus)

hist(plus\_dfus)

#curve(dnorm(plus\_dfus, mean(plus\_dfus), sd(plus\_dfus)), add = TRUE,

#      col = 3, lwd = 3)

qqnorm(plus\_dfus)

qqline(plus\_dfus, col = "red")

shapiro.test(plus\_dfus)

##

psych::describe(plus\_dfxr)

hist(plus\_dfxr)

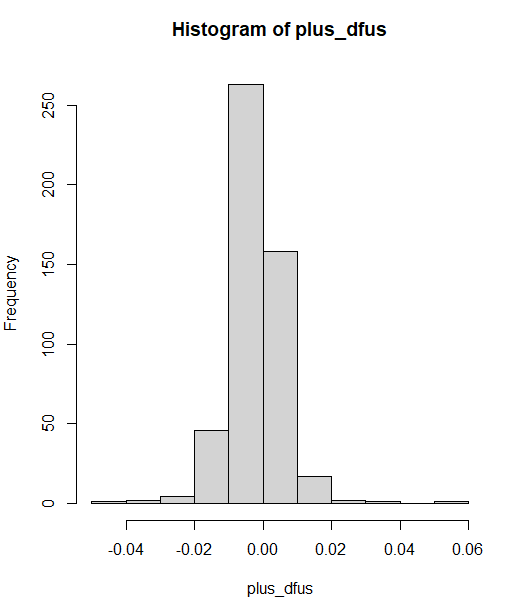
#curve(dnorm(plus\_dfxr, mean(plus\_dfxr), sd(plus\_dfxr)), add = TRUE,

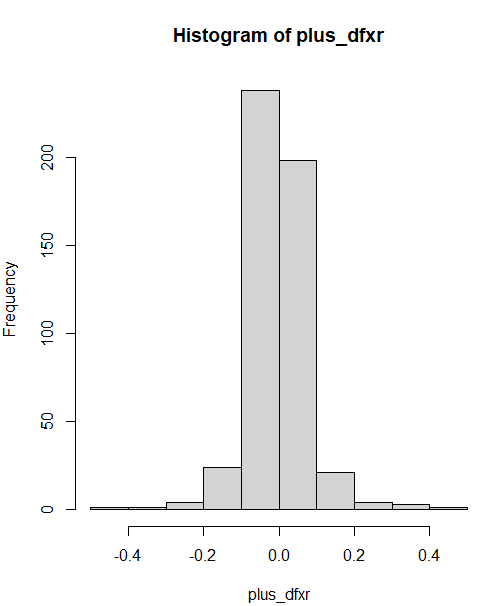
#      col = 3, lwd = 3)

qqnorm(plus\_dfxr)

qqline(plus\_dfxr, col = "red")

shapiro.test(plus\_dfxr)





Последний пункт не совсем понятен + не нашёл в заданиях и решениях что-то близкое по смыслу, сделал как мог, без отладки это wiki делать сложно и практически невозможно

install.packages("gamlss")

library(gamlss)

model3 <- gamlss(reg, ~ sign\_dfus)

summary(model3)

