Задание 1. Прогноз интенсивности дорожного движения в США (с прошлых

занятий). Гетероскедастичность

Для начала устанавливаем путь к папке с нашими данными и загружаем данные. Пропускаем первые пять строк поскольку они не содержат данных.

setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

df <- read.delim("StreetLight2020.tsv", skip = 5)

Проверяем наши данные на корректность

head(df)

Строим регрессию

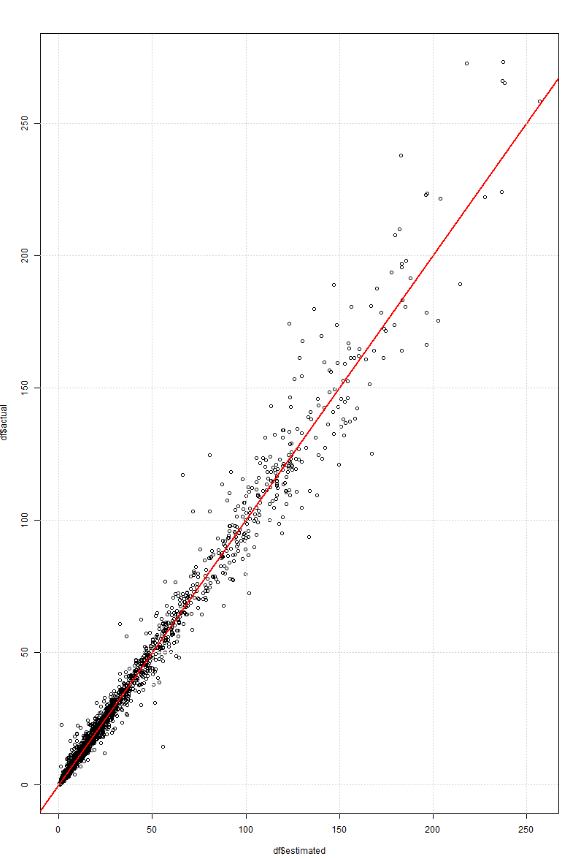
reg1 <- lm(df$actual ~ df$estimated)

И строим график наших данные

plot(df$actual ~ df$estimated)

abline(c(0, 1), lwd = 2, col = "red")

grid()

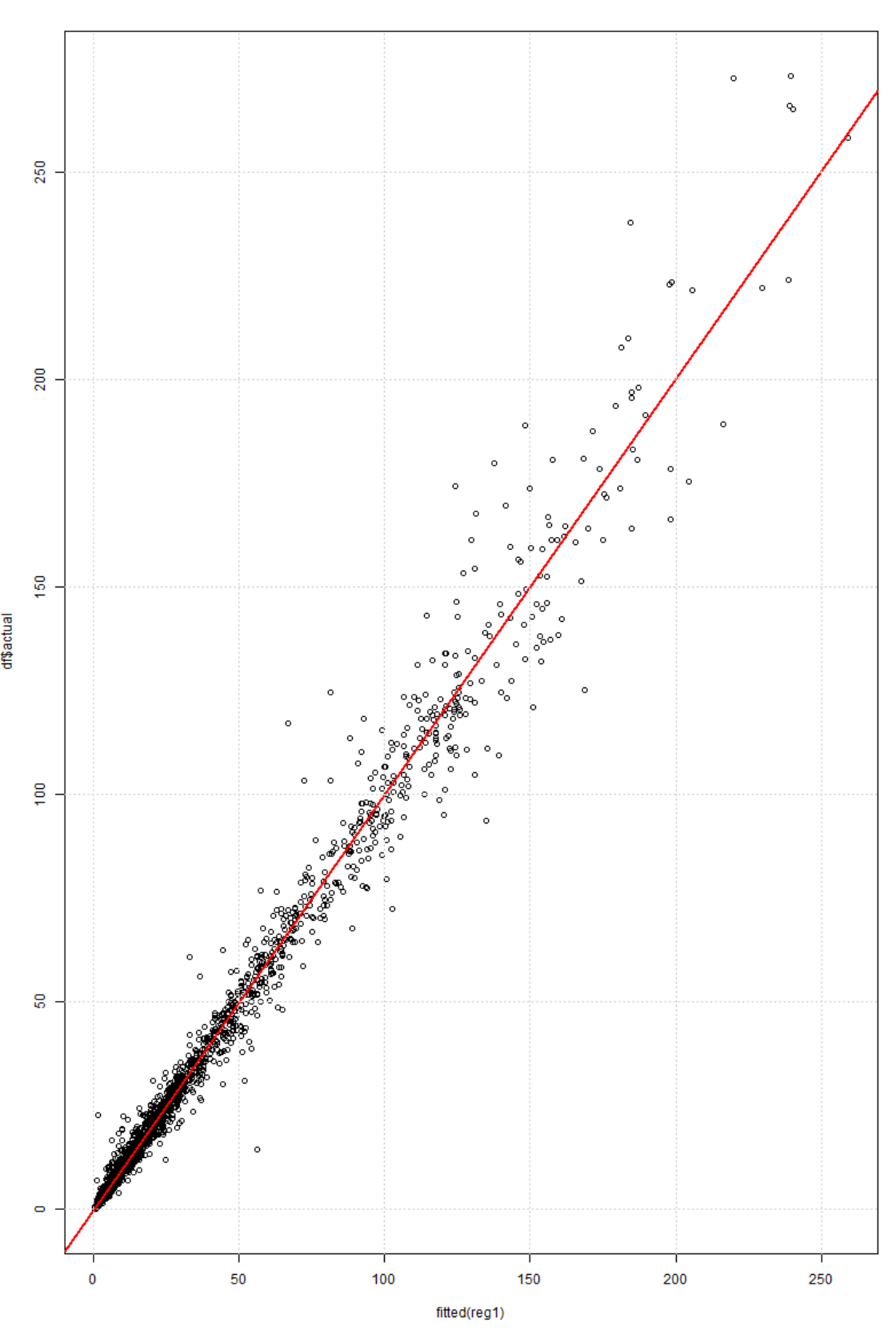


Далее строим данные фактических наблюдений от подогнанных значений регрессии

plot(df$actual ~ fitted(reg1))

abline(c(0, 1), lwd = 2, col = "red")

grid()



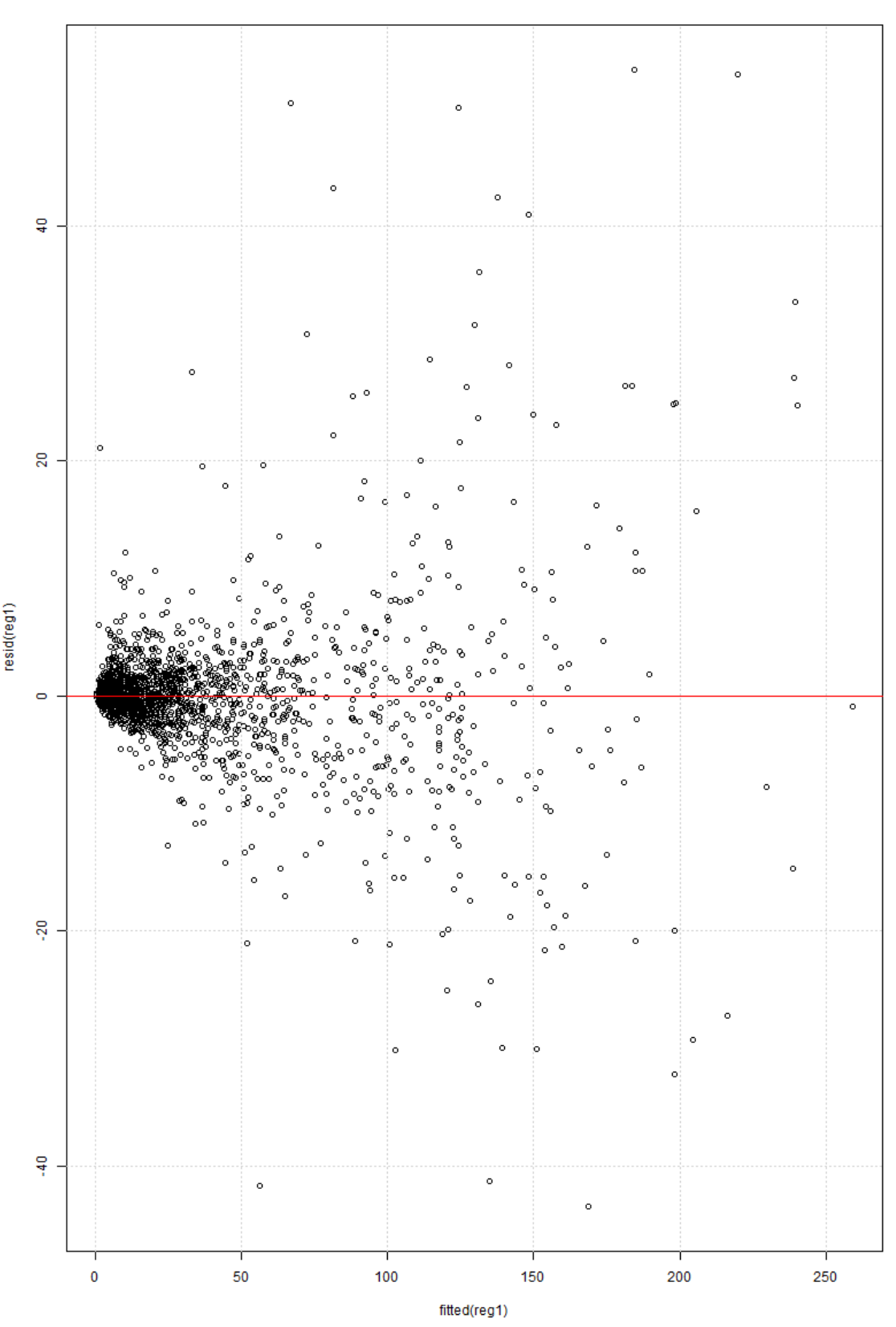
Построим график остатков модели от подогнанных значений регрессии и построим линию нуля

plot(resid(reg1) ~ fitted(reg1), data = df)

points(sqrt(fitted(reg.bp)) ~ rank(df$estimated))

grid()

abline(h = 0, col = "red")



Строим новую регрессию

reg.bp <- lm(resid(reg1)^2 ~ df$estimated)

Строим график вспомогательных значений от выборочных рангов значений в расчитанной фирмой оценках и строим точки корней от подогнанных значений регрессии от выборочных рангов

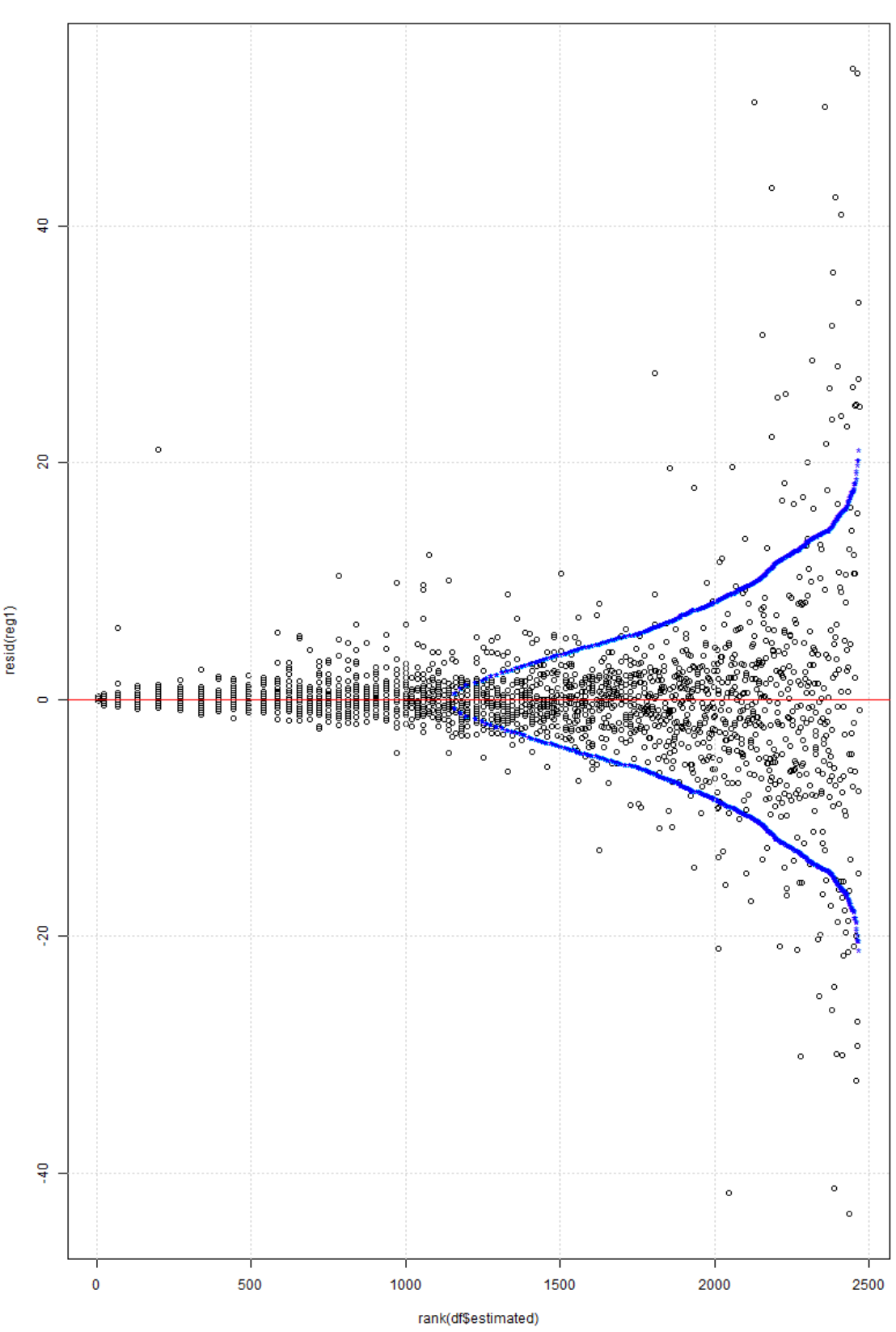
plot(resid(reg1) ~ rank(df$estimated))

points(sqrt(fitted(reg.bp)) ~ rank(df$estimated), col = "blue", pch = "\*")

points(-sqrt(fitted(reg.bp)) ~ rank(df$estimated), col = "blue", pch = "\*")

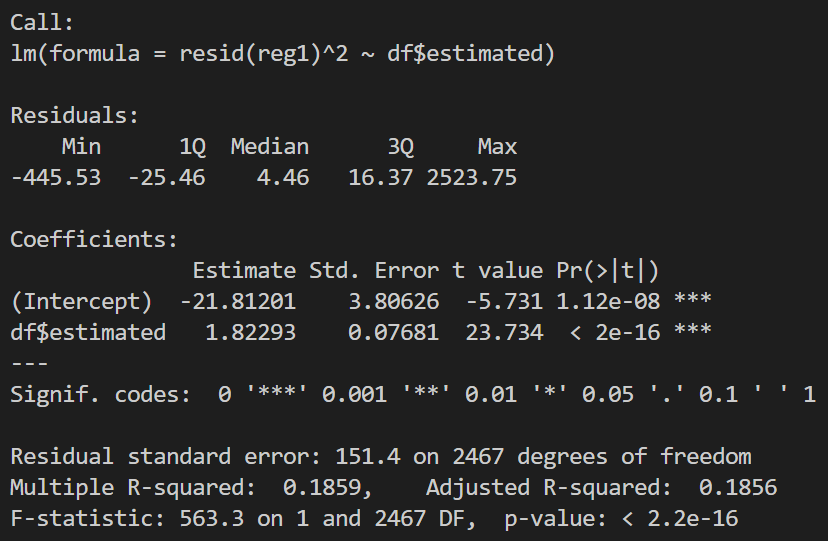
grid()

abline(h = 0, col = "red")



Посмотрим на основные статистические характеристики построенной регрессии

summary(reg.bp)



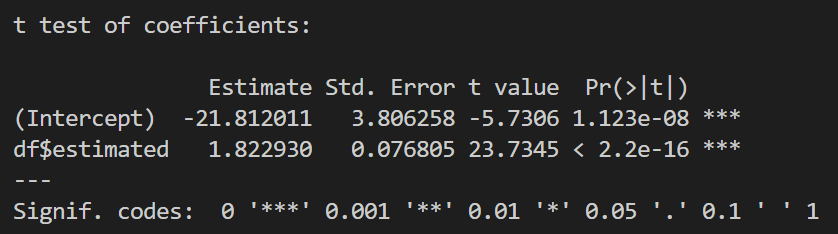
Загружаем и инициализируем библиотеку ***lmtest***

install.packages("lmtest")

library(lmtest)

Проводим t тест для нашей регрессии

lmtest::coeftest(reg.bp)



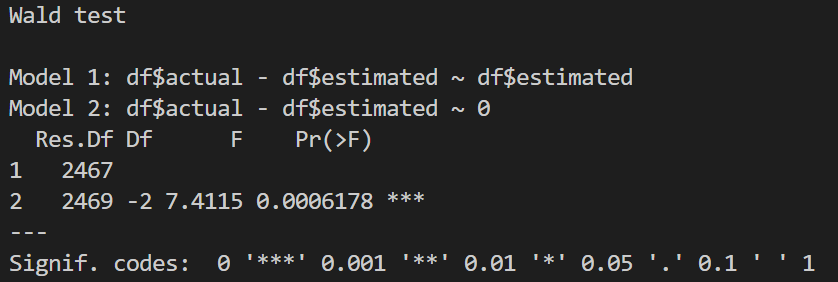
Постротим регрессию разности фактических наблюдений от расчитанных фирмой оценок от расчитанных фирмой оценок и регрессию разности фактических наблюдений от расчитанных фирмой оценок от нуля

reg1 <- lm(df$actual - df$estimated ~ df$estimated)

reg0 <- lm(df$actual - df$estimated ~ 0)

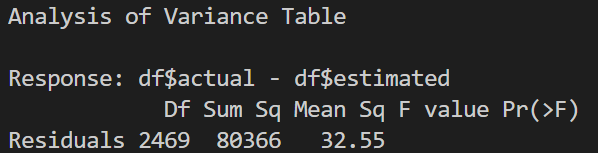
Проведём для данных регрессий тест Вальда

lmtest::waldtest(reg1, reg0)



Построим для нулевой регрессии таблицу анализа дисперсии

anova(reg0)



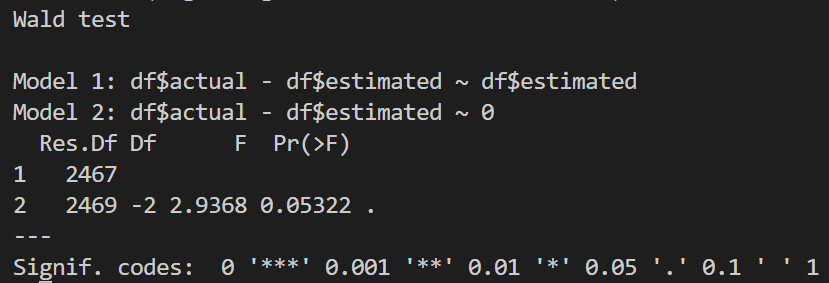
Теперь установим и инициализируем библиотеку ***sandwich***

install.packages("sandwich")

library(sandwich)

Теперь снова проведём тест Вальда, но уже с согласованной с гетероскедастичностью ковариационной матричной оценкой

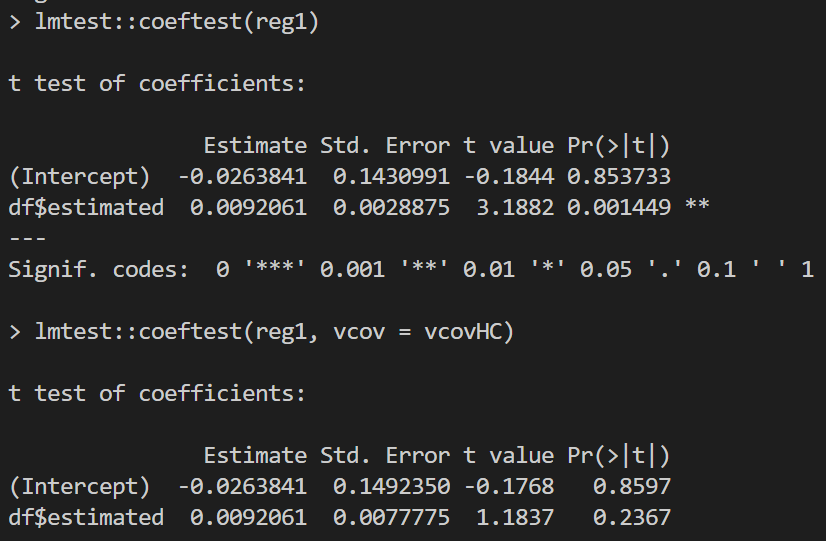
waldtest(reg1, reg0, vcov = sandwich::vcovHC)



Теперь проведём ещё ряд тестов для сравнения их с и без согласованной с гетерастичностью матричной оценкой

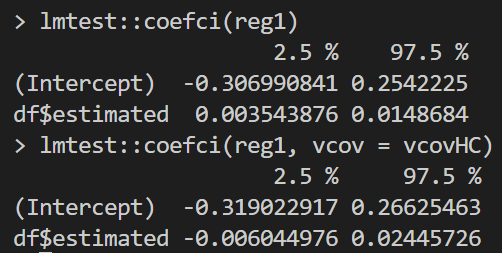
lmtest::coeftest(reg1)

lmtest::coeftest(reg1, vcov = vcovHC)



lmtest::coefci(reg1)

lmtest::coefci(reg1, vcov = vcovHC)



**Задание 2. Расходы на питание: Гетероскедастичность**

Установим и инициализируем библиотеку ***quantreg***

install.packages("quantreg")

library(quantreg)

Загружаем дату “engel” из библиотеки ***quantreg*** и присваиваем эти данные нашей переменной, а сами данные удаляем

data("engel")

df <- engel

rm(engel)

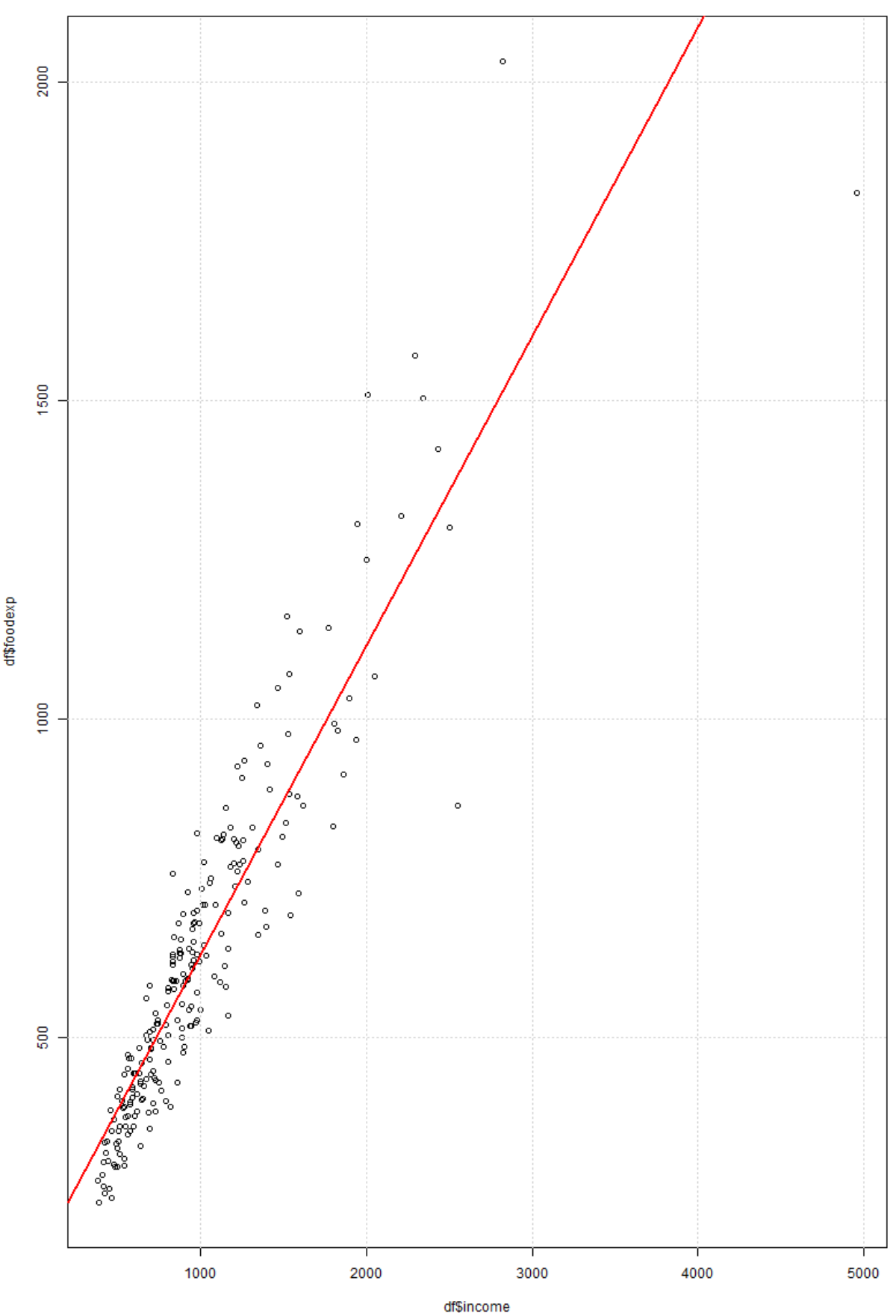
Построим график по нашим данным и добавим линию регрессии от этих данных на график

plot(df$foodexp ~ df$income)

grid()

reg1 <- lm(df$foodexp ~ df$income)

abline(reg1, col = "red", lwd = 2)

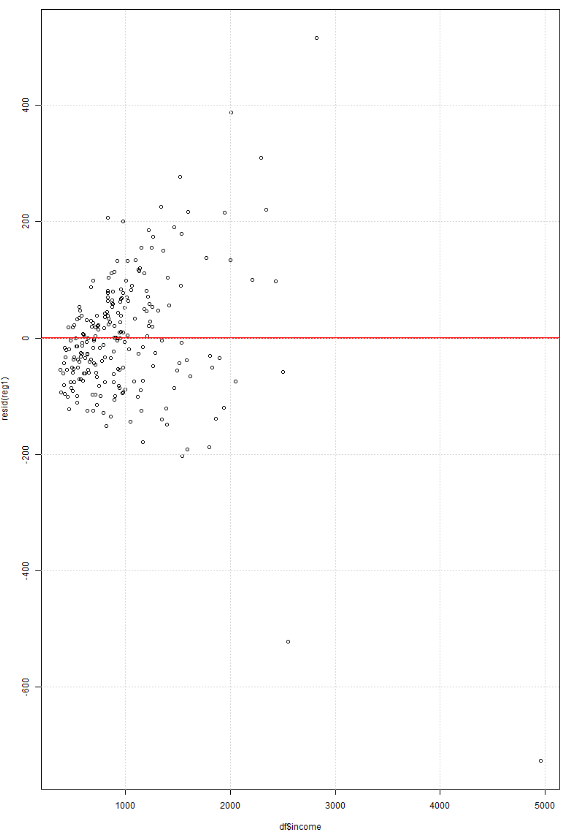


Теперь построим график вспомогательных значений регрессии от income

plot(resid(reg1) ~ df$income)

abline(h = 0, col = "red", lwd = 2)

grid()

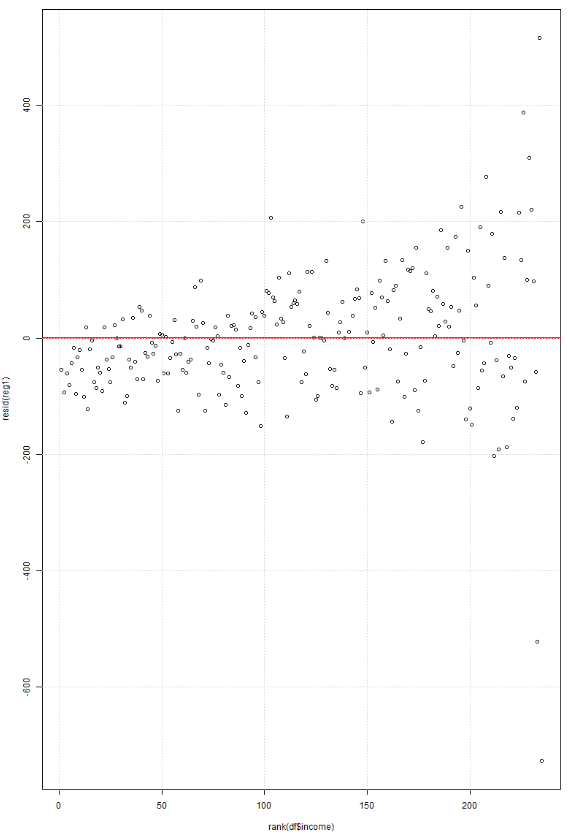


Теперь построим график вспомогательных значений регрессии от выборочных рангов

plot(resid(reg1) ~ rank(df$income))

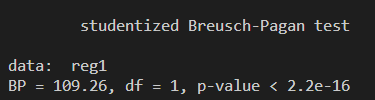
abline(h = 0, col = "red", lwd = 2)

grid()



Проверим нашу регрессию тестом Бреуша-Пагана

lmtest::bptest(reg1)

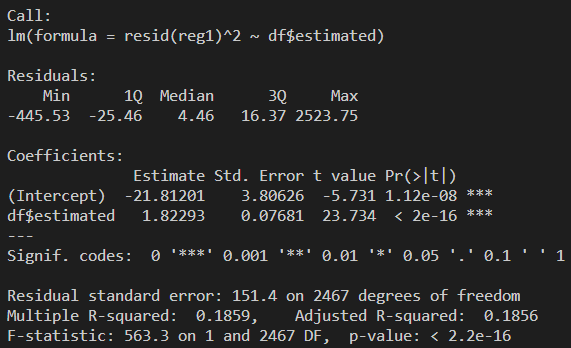


Построим новую регрессию от квадрата вспомогательных значений и параметра income

reg.bp <- lm(resid(reg1)^2 ~ df$income)

Посмотрим на стандартные статистические параметры данной регрессии

summary(reg.bp)



Далее построим схожий график с предыдущим заданием, а именно строим график вспомогательных значений от выборочных рангов значений в расчитанной фирмой оценках и строим точки корней от подогнанных значений регрессии от выборочных рангов

plot(resid(reg1) ~ rank(df$income))

points(sqrt(fitted(reg.bp)) ~ rank(income),

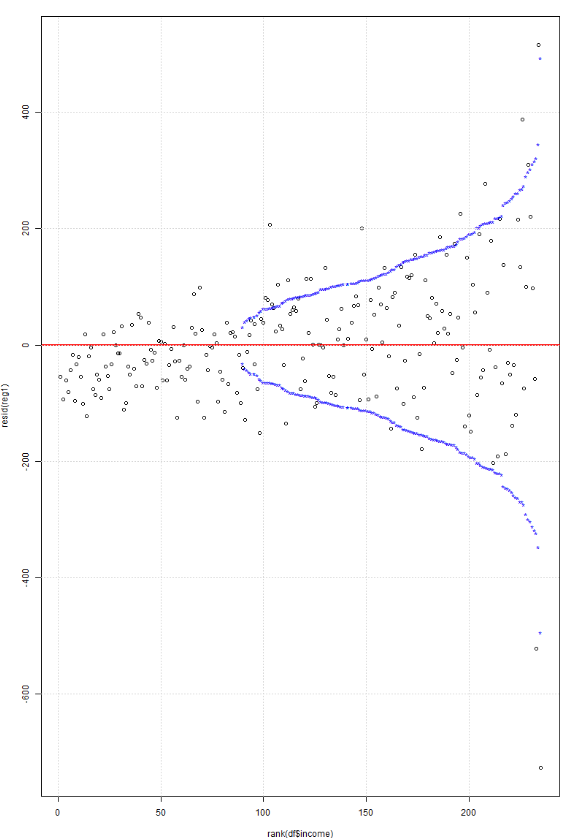
data = df, col = "blue", pch = "\*")

points(-sqrt(fitted(reg.bp)) ~ rank(income),

data = df, col = "blue", pch = "\*")

abline(h = 0, col = "red", lwd = 2)

grid()

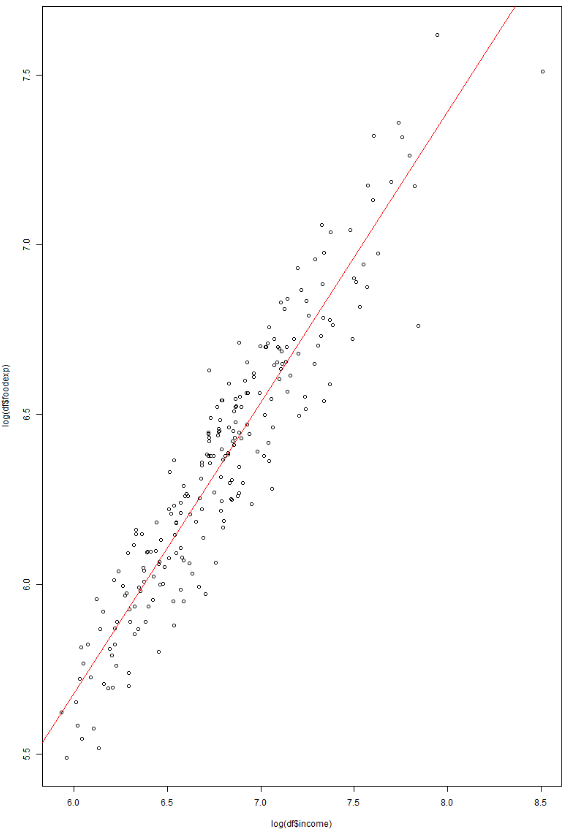


Теперь построим график логарифмлв наших данных и линию регрессии логарифмов данных

plot(log(df$foodexp) ~ log(df$income))

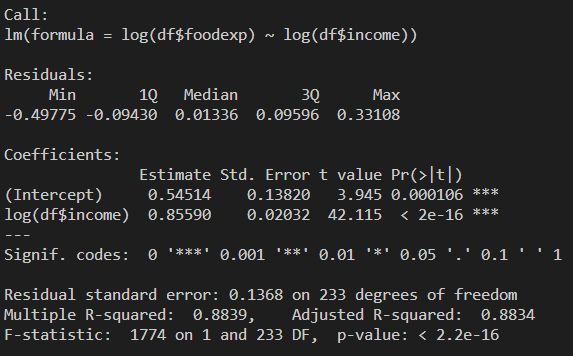
reg.log <- lm(log(df$foodexp) ~ log(df$income))

abline(reg.log, col = "red")

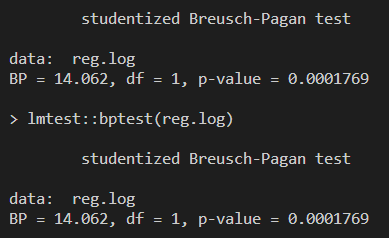


Посчитаем все основные статистические коэффициенты

summary(reg.log)



lmtest::bptest(reg.log)

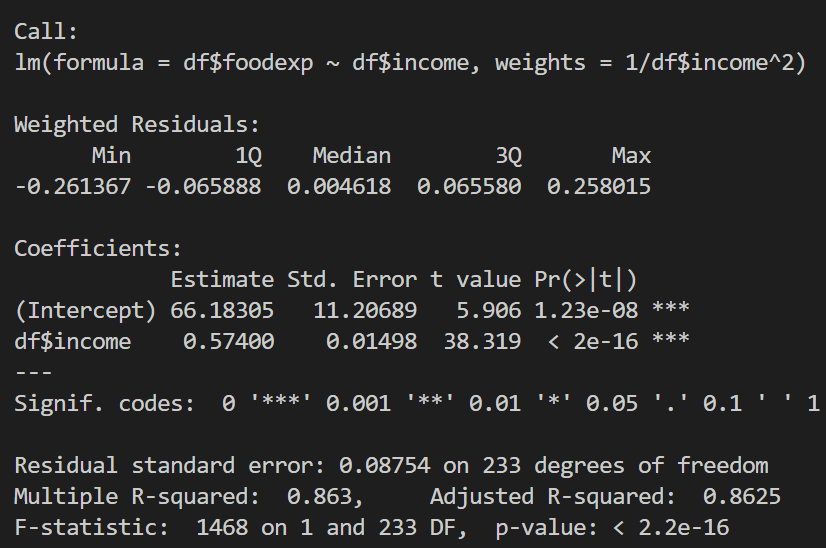


Теперь построим регрессию наших данных с весом

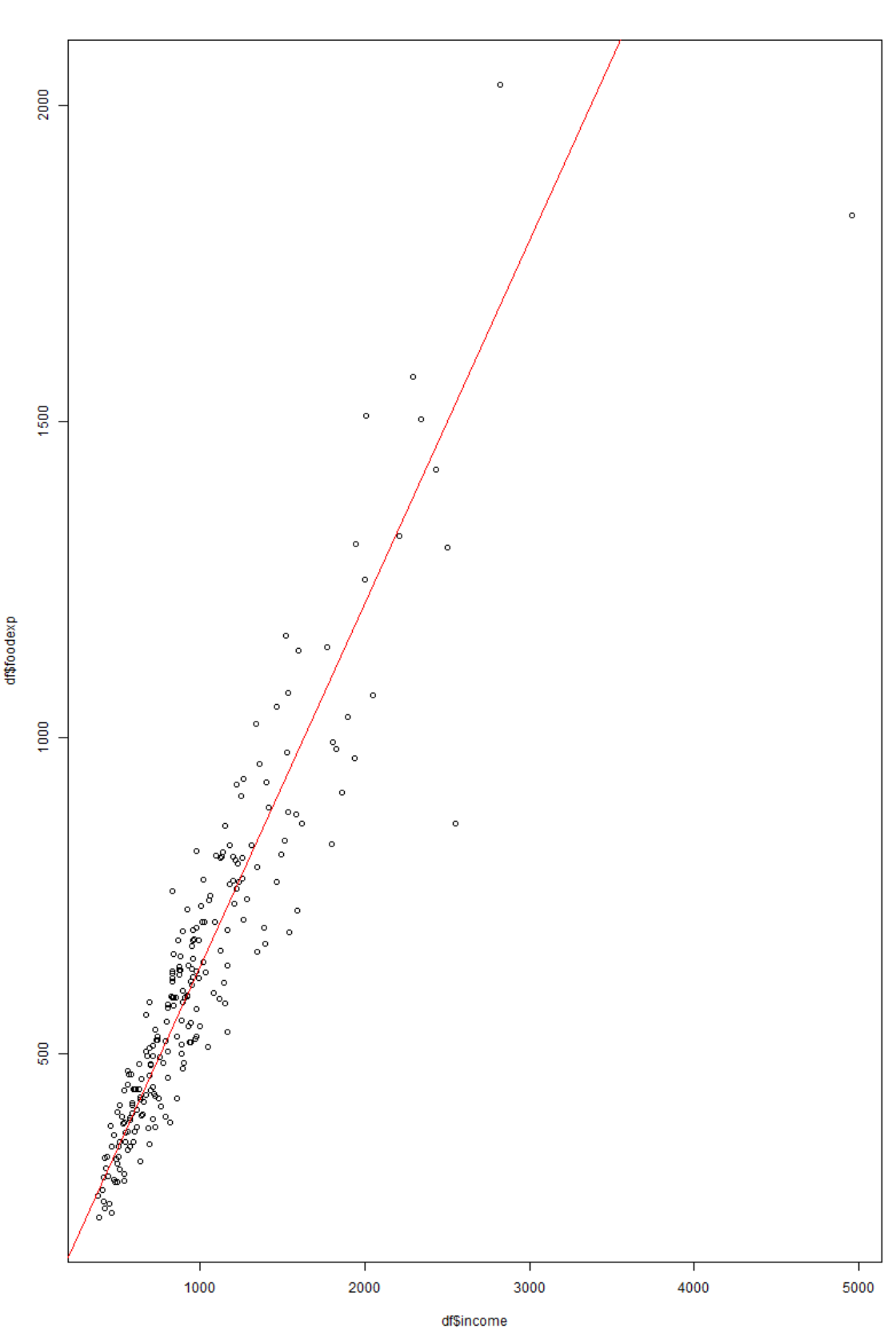
reg.w <- lm(df$foodexp ~ df$income, weights = 1 / df$income^2)

Посчитаем все основные статистические коэффициенты

summary(reg.w)



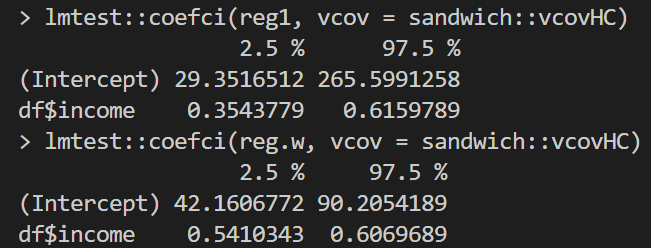
Посроим график наших данных с линией новой регрессии



Получаем расчётные коэффициенты для двух наших регрессий с использованием согласованной гетероскедастичностью ковариационной матричной оценкой

lmtest::coefci(reg1, vcov = sandwich::vcovHC)

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



Установим и инициализируем библиотеку ***gamlss***

install.packages("gamlss")

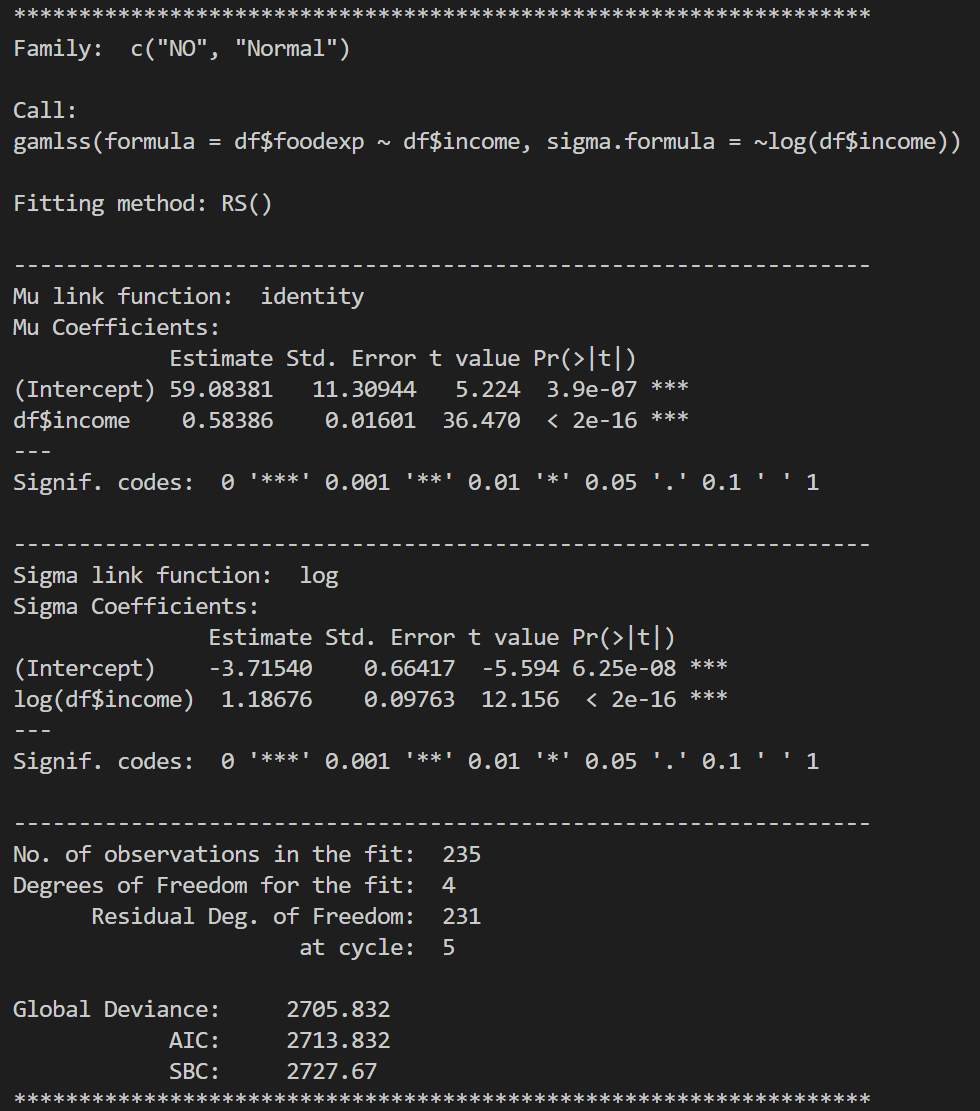
library(gamlss)

Построим модель

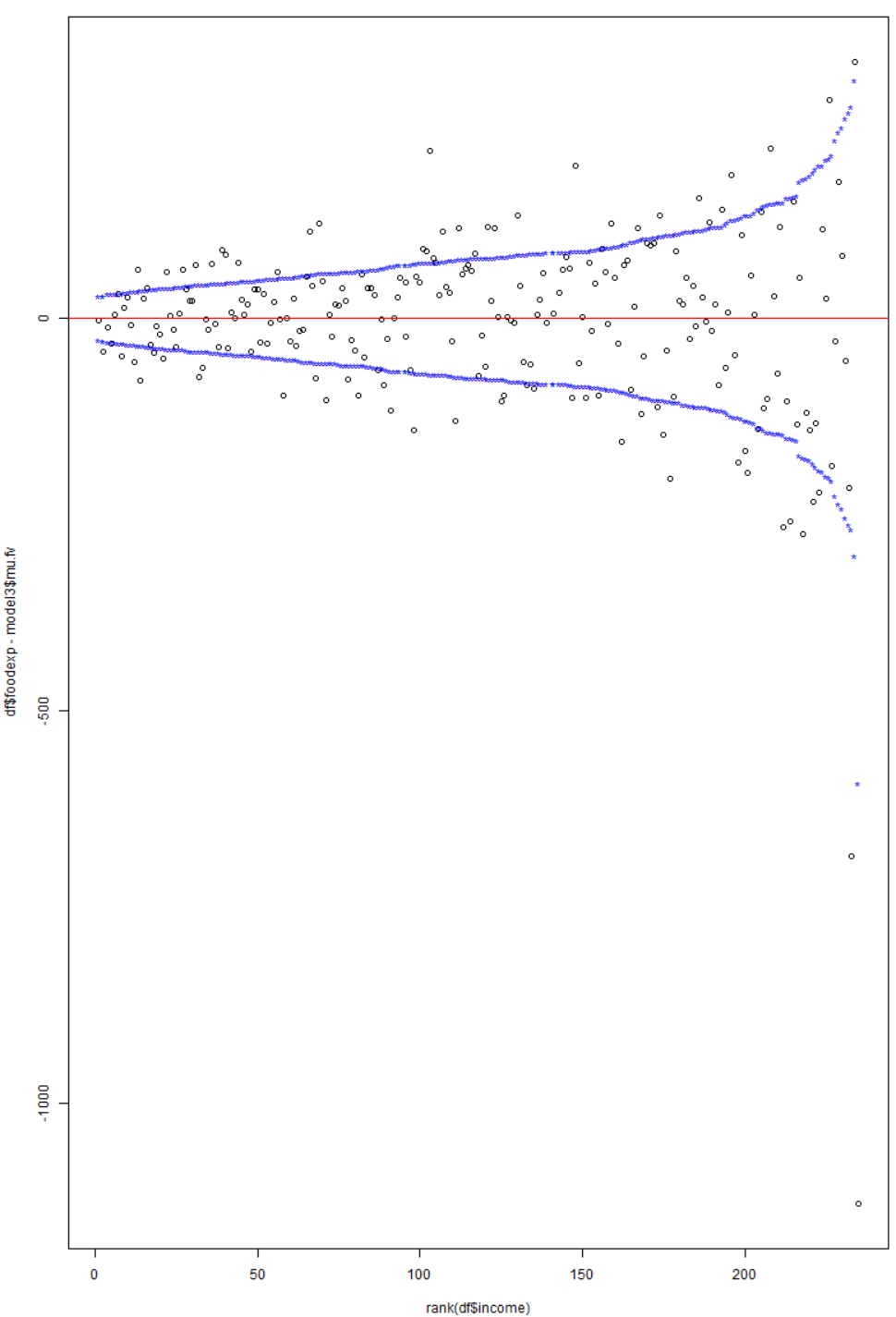
model3 <- gamlss(df$foodexp ~ df$income, ~ log(df$income))

И посмотрим статистики

summary(model3)

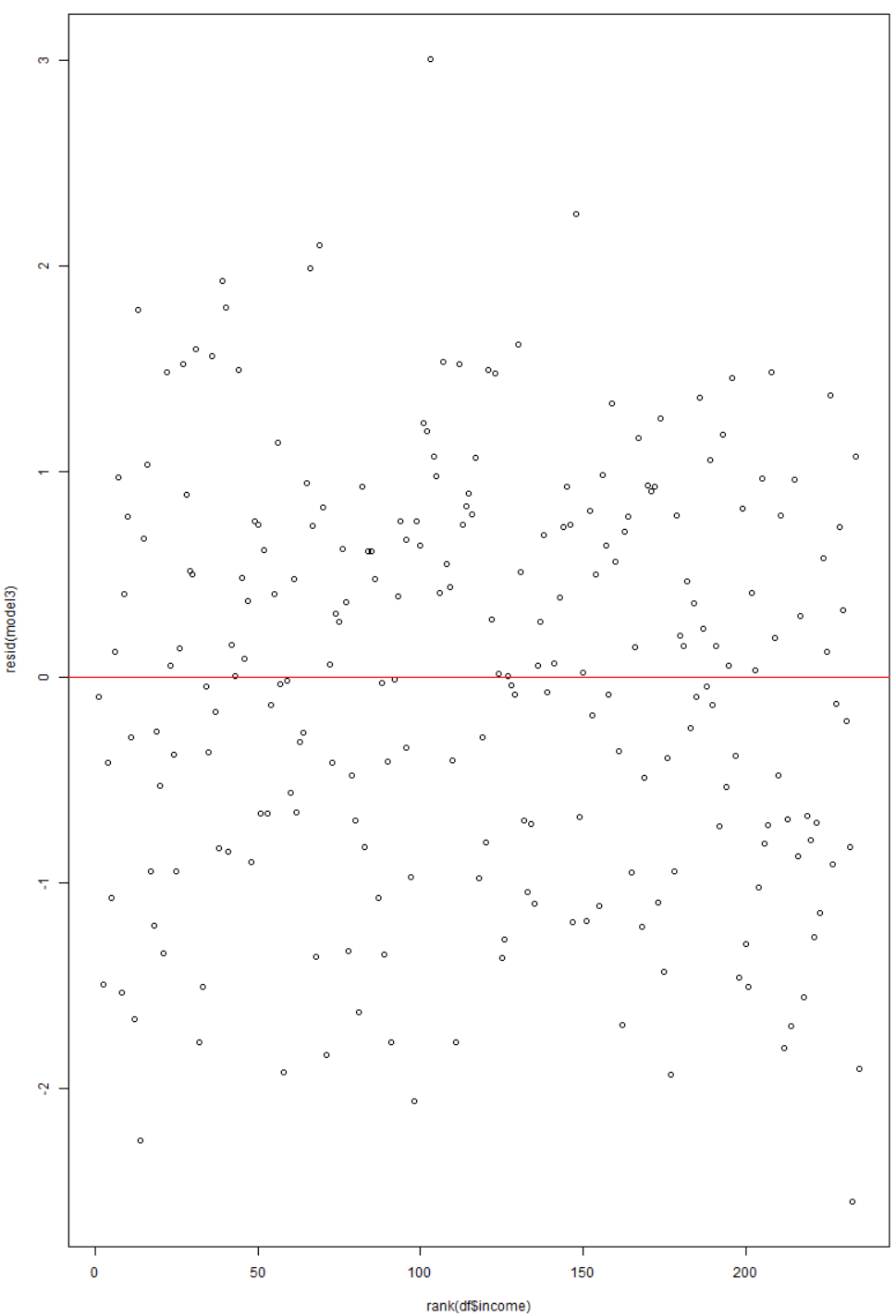


Строим график foodexp минус подобранные значения модели mu от выборочных рангов значений



plot(resid(model3) ~ rank(df$income))

abline(h = 0, col = "red")



Теперь перейдём к квантильной регрессии

Сначала построим медианню регрессию и посмотрим её на графики с обычной регрессией

qreg <- quantreg::rq(df$foodexp ~ df$income) # median regression

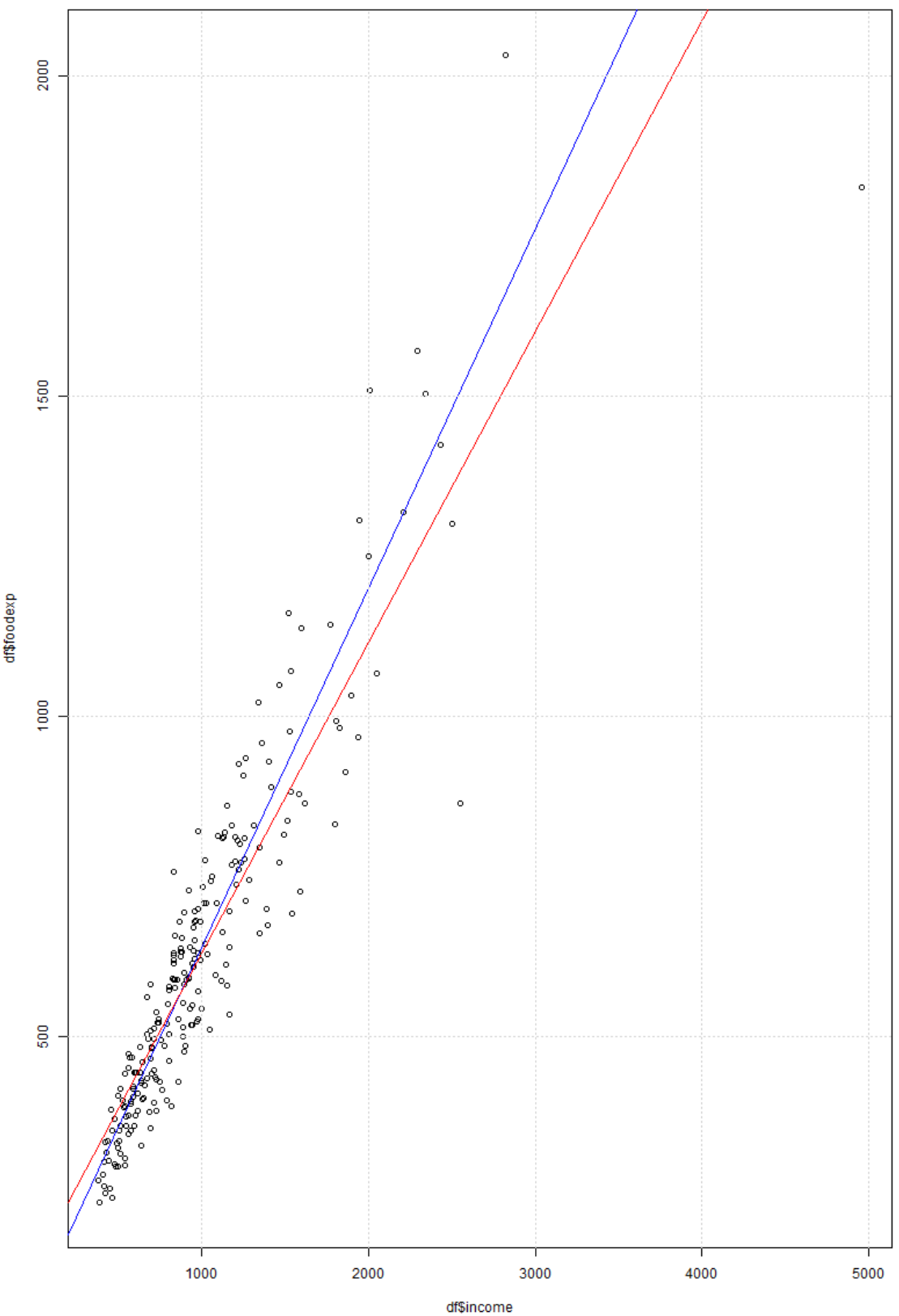
summary(qreg)

plot(df$foodexp ~ df$income)

abline(qreg, col = "blue")

abline(reg1, col = "red")

grid()

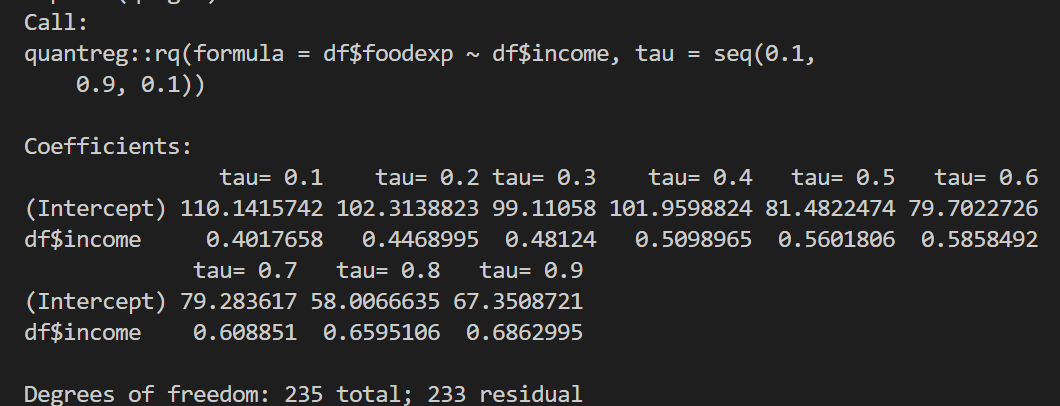


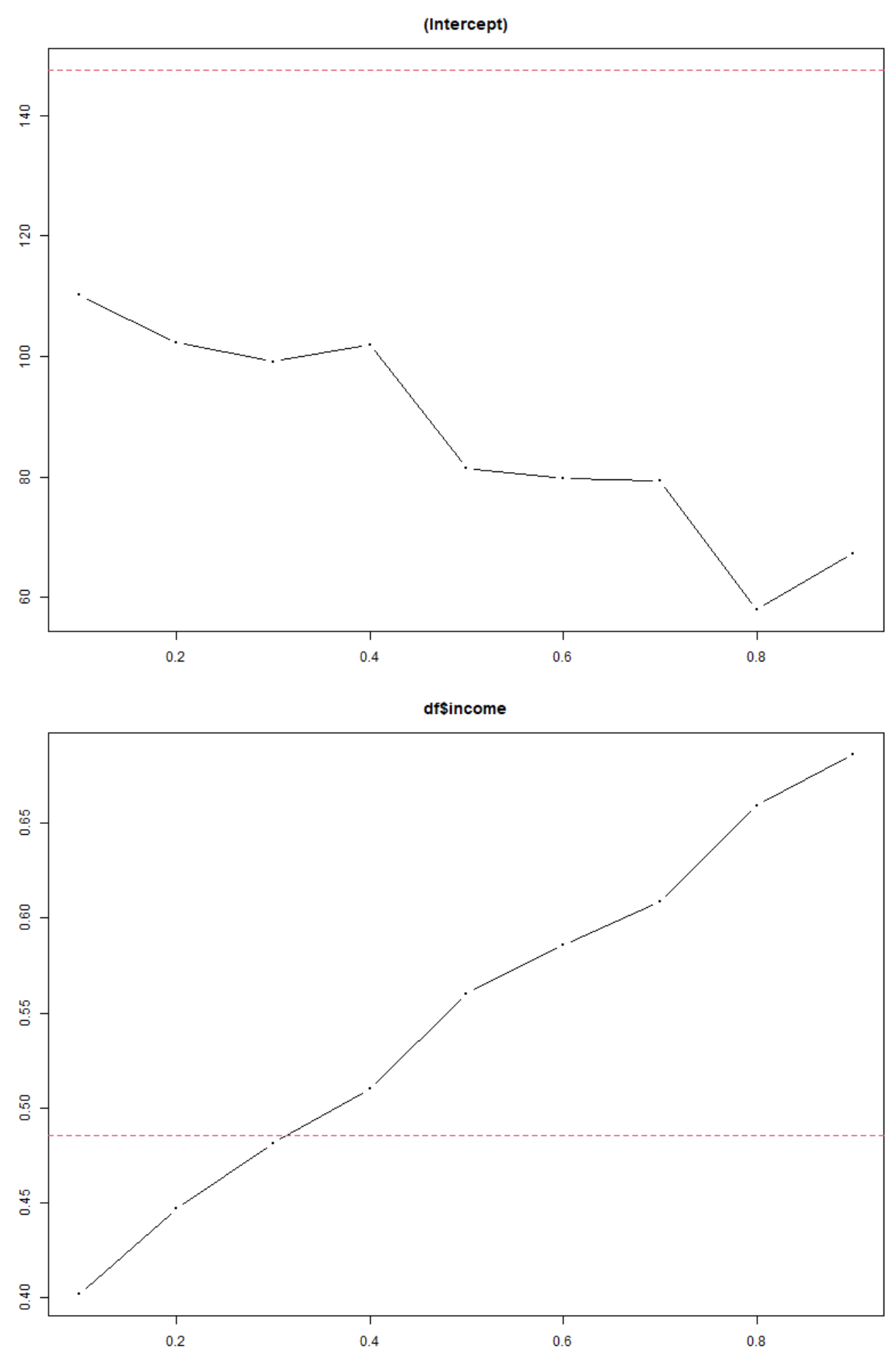
Теперь построим кватнтильную регрессию для различных тау и посмотрим на них

qreg10 <- quantreg::rq(df$foodexp ~ df$income, tau = seq(0.1, 0.9, 0.1))

print(qreg10)

plot(qreg10)





Построим коэффициенты квантильной регрессиии для всех тау на графике foodexp от income

plot(df$foodexp ~ df$income)

grid()

arr <- c(1:9)

for(x in arr){

  abline(coef(qreg10)[, x], col = x, lwd = 2)

}

