Лабораторная работа №5.0

**Цель работы:** Научиться следующим навыкам:

#### 

1. Использование датафреймов для статистических вычислений;
2. Преобразование данных для удобной работы с ними;
3. Линейные модели.

### Загрузка исходных данных

Проанализируйте данные о возрасте и физ. характеристиках молюсков

datafrm <- read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data", header = TRUE, sep = ",")  
summary(datafrm)

## M X0.455 X0.365 X0.095   
## F:1307 Min. :0.075 Min. :0.0550 Min. :0.0000   
## I:1342 1st Qu.:0.450 1st Qu.:0.3500 1st Qu.:0.1150   
## M:1527 Median :0.545 Median :0.4250 Median :0.1400   
## Mean :0.524 Mean :0.4079 Mean :0.1395   
## 3rd Qu.:0.615 3rd Qu.:0.4800 3rd Qu.:0.1650   
## Max. :0.815 Max. :0.6500 Max. :1.1300   
## X0.514 X0.2245 X0.101 X0.15   
## Min. :0.0020 Min. :0.0010 Min. :0.00050 Min. :0.0015   
## 1st Qu.:0.4415 1st Qu.:0.1860 1st Qu.:0.09337 1st Qu.:0.1300   
## Median :0.7997 Median :0.3360 Median :0.17100 Median :0.2340   
## Mean :0.8288 Mean :0.3594 Mean :0.18061 Mean :0.2389   
## 3rd Qu.:1.1533 3rd Qu.:0.5020 3rd Qu.:0.25300 3rd Qu.:0.3290   
## Max. :2.8255 Max. :1.4880 Max. :0.76000 Max. :1.0050   
## X15   
## Min. : 1.000   
## 1st Qu.: 8.000   
## Median : 9.000   
## Mean : 9.932   
## 3rd Qu.:11.000   
## Max. :29.000

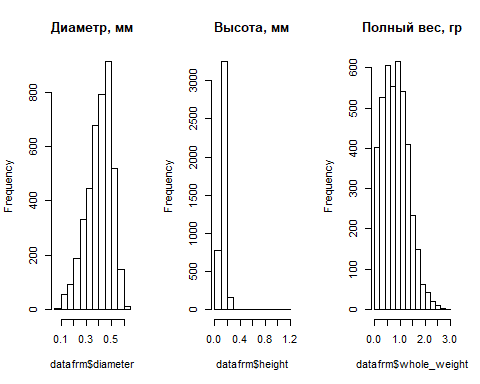
colnames(datafrm)

## [1] "M" "X0.455" "X0.365" "X0.095" "X0.514" "X0.2245" "X0.101"   
## [8] "X0.15" "X15"

colnames(datafrm) <- c("sex", "length", "diameter", "height",  
 "whole\_weight", "shucked\_weight",  
 "viscera\_weight", "shell\_weight", "rings")  
  
colnames(datafrm)

## [1] "sex" "length" "diameter" "height"   
## [5] "whole\_weight" "shucked\_weight" "viscera\_weight" "shell\_weight"   
## [9] "rings"

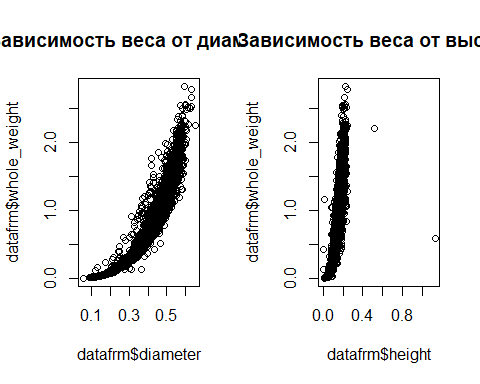
datafrm$sex <- factor(c("Female", "Infant", "Male")[datafrm$sex])  
par(mfrow = c(1, 3))  
hist(datafrm$diameter, main = "Диаметр, мм")  
hist(datafrm$height, main = "Высота, мм")  
hist(datafrm$whole\_weight, main = "Полный вес, гр")



Видим ассиметрию и выбросы (от них нужно избавиться)

### Визулизируем возможные зависимости

par(mfrow = c(1, 2))  
plot(datafrm$diameter, datafrm$whole\_weight, 'p', main = "Зависимость веса от диаметра")  
plot(datafrm$height, datafrm$whole\_weight, 'p', main = "Зависимость веса от высоты")



Хорошо видна зависимость, нужно её исследовать.

### Построить линейные модели при помощи функции lm, посмотреть их характеристики

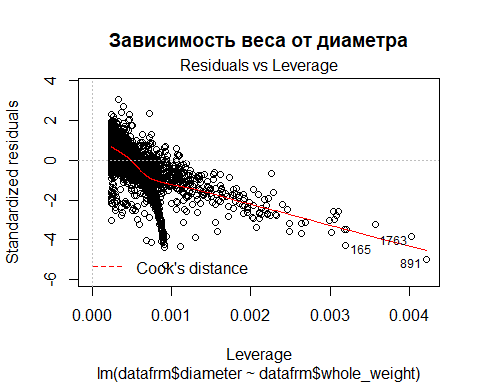
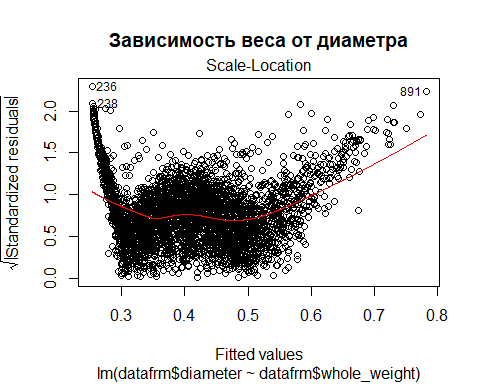
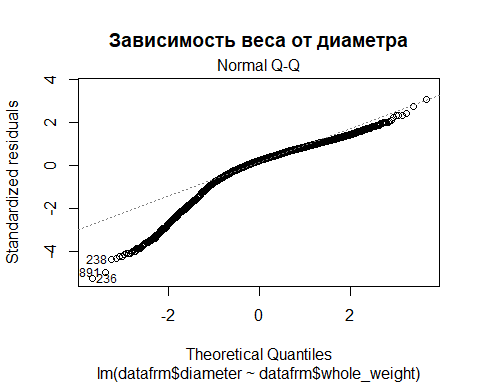
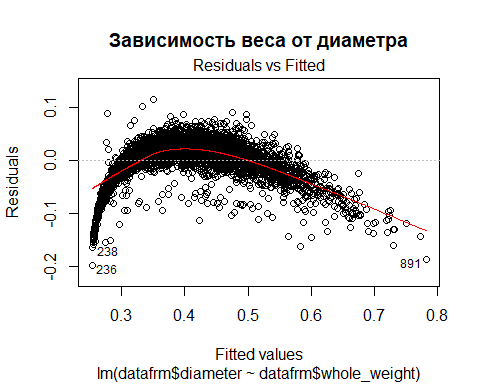
linear.model.d\_w <- lm(datafrm$diameter ~ datafrm$whole\_weight, data = datafrm)  
summary(linear.model.d\_w)

##   
## Call:  
## lm(formula = datafrm$diameter ~ datafrm$whole\_weight, data = datafrm)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.198038 -0.015281 0.008041 0.024858 0.114478   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.252664 0.001143 221.1 <2e-16 \*\*\*  
## datafrm$whole\_weight 0.187288 0.001187 157.8 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.03761 on 4174 degrees of freedom  
## Multiple R-squared: 0.8565, Adjusted R-squared: 0.8564   
## F-statistic: 2.491e+04 on 1 and 4174 DF, p-value: < 2.2e-16

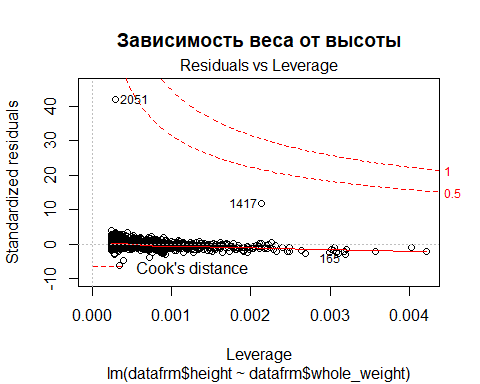
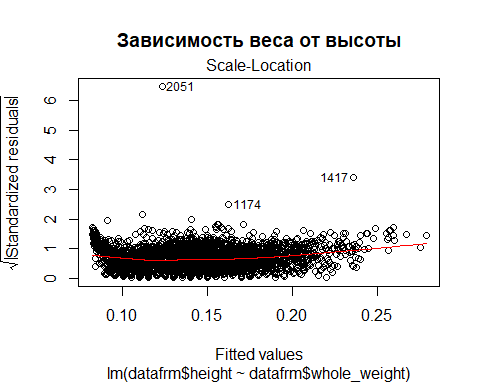
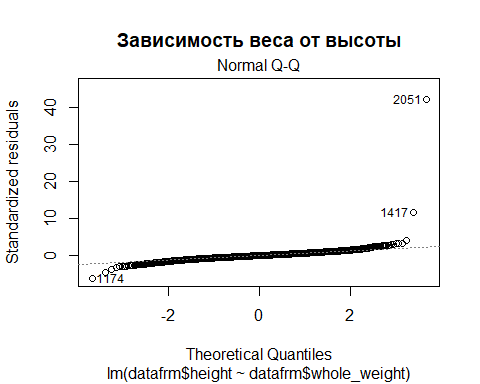
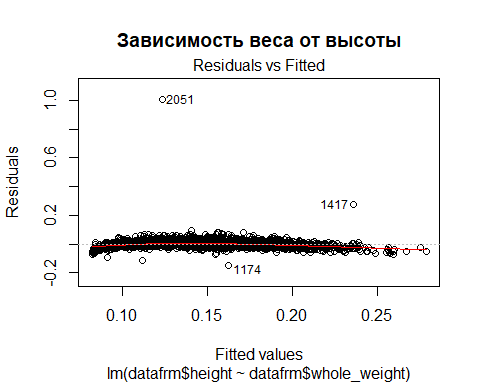
linear.model.h\_w <- lm(datafrm$height ~ datafrm$whole\_weight, data = datafrm)  
summary(linear.model.h\_w)

##   
## Call:  
## lm(formula = datafrm$height ~ datafrm$whole\_weight, data = datafrm)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.14742 -0.01031 -0.00035 0.00993 1.00688   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.0816199 0.0007291 111.95 <2e-16 \*\*\*  
## datafrm$whole\_weight 0.0698672 0.0007571 92.29 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.02399 on 4174 degrees of freedom  
## Multiple R-squared: 0.6711, Adjusted R-squared: 0.671   
## F-statistic: 8517 on 1 and 4174 DF, p-value: < 2.2e-16

plot(linear.model.d\_w, main = "Зависимость веса от диаметра")

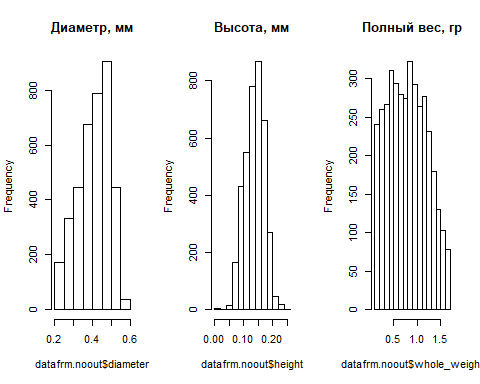


plot(linear.model.h\_w, main = "Зависимость веса от высоты")



### Избавиться от выборосов, построить ещё модели и проверить их

datafrm.noout <- datafrm[datafrm$height < 0.4 & datafrm$diameter > 0.2 & datafrm$diameter < 0.6 & datafrm$whole\_weight < 1.7 & datafrm$whole\_weight > 0.1,]  
par(mfrow = c(1, 3))  
hist(datafrm.noout$diameter, main = "Диаметр, мм")  
hist(datafrm.noout$height, main = "Высота, мм")  
hist(datafrm.noout$whole\_weight, main = "Полный вес, гр")



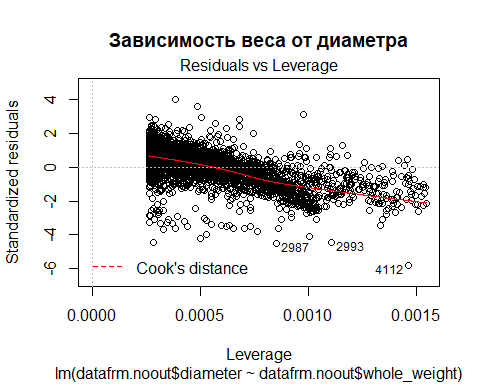
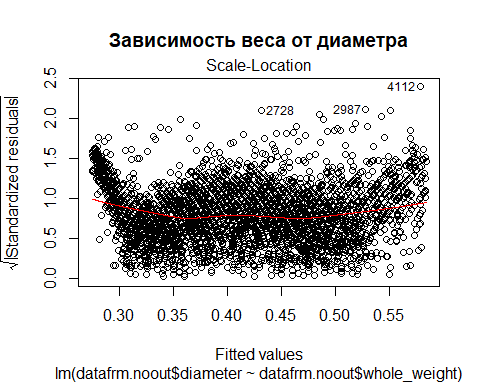
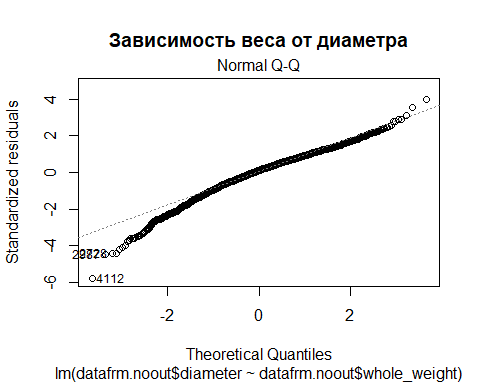
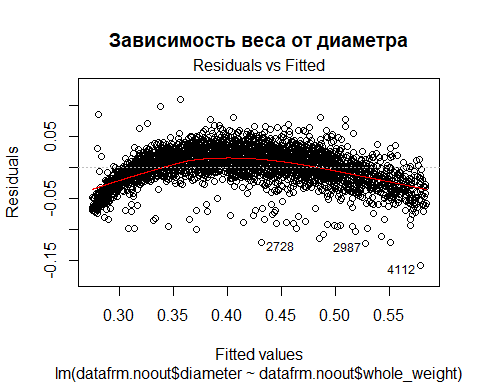
par(mfrow = c(1, 1))  
  
linear.model.d\_w <- lm(datafrm.noout$diameter ~ datafrm.noout$whole\_weight, data = datafrm.noout)  
summary(linear.model.d\_w)

##   
## Call:  
## lm(formula = datafrm.noout$diameter ~ datafrm.noout$whole\_weight,   
## data = datafrm.noout)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.157985 -0.014944 0.003189 0.018626 0.108706   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.2548634 0.0009811 259.8 <2e-16 \*\*\*  
## datafrm.noout$whole\_weight 0.1941255 0.0010945 177.4 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.02735 on 3800 degrees of freedom  
## Multiple R-squared: 0.8922, Adjusted R-squared: 0.8922   
## F-statistic: 3.146e+04 on 1 and 3800 DF, p-value: < 2.2e-16

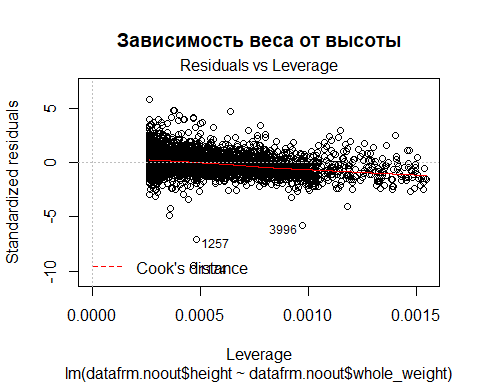
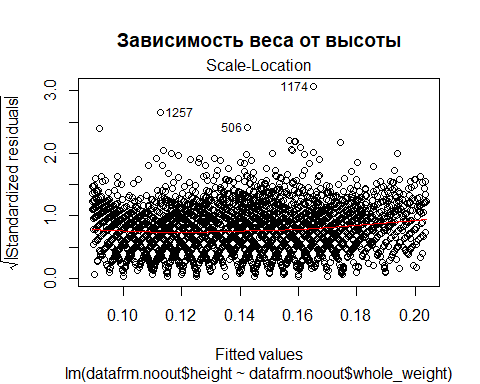
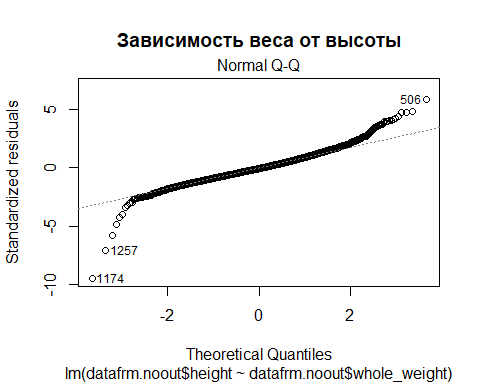
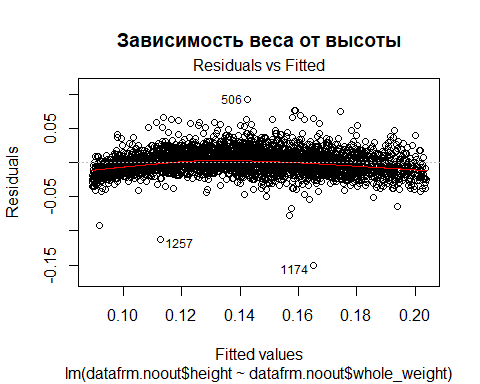
linear.model.h\_w <- lm(datafrm.noout$height ~ datafrm.noout$whole\_weight, data = datafrm.noout)  
summary(linear.model.h\_w)

##   
## Call:  
## lm(formula = datafrm.noout$height ~ datafrm.noout$whole\_weight,   
## data = datafrm.noout)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.150195 -0.009841 -0.001024 0.008979 0.092608   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.0820025 0.0005708 143.7 <2e-16 \*\*\*  
## datafrm.noout$whole\_weight 0.0719349 0.0006368 113.0 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.01591 on 3800 degrees of freedom  
## Multiple R-squared: 0.7706, Adjusted R-squared: 0.7705   
## F-statistic: 1.276e+04 on 1 and 3800 DF, p-value: < 2.2e-16

plot(linear.model.d\_w, main = "Зависимость веса от диаметра")



plot(linear.model.h\_w, main = "Зависимость веса от высоты")

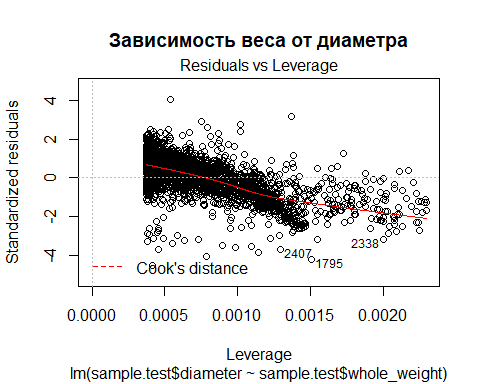
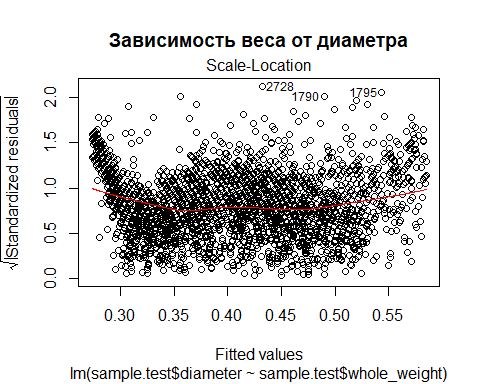
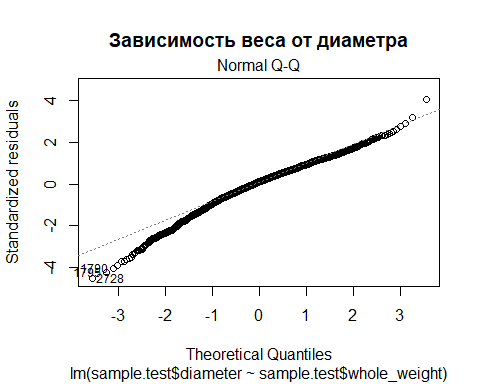
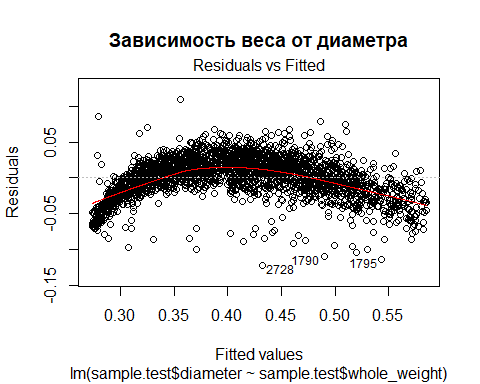


### разделить массив данных на 2 случайные части

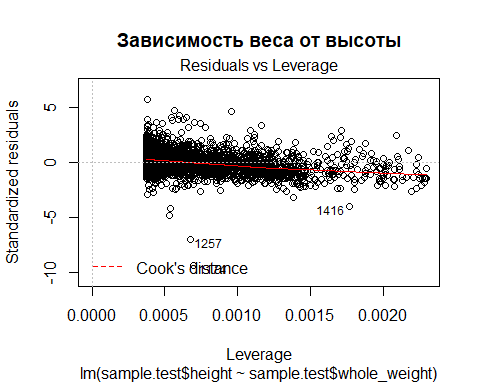
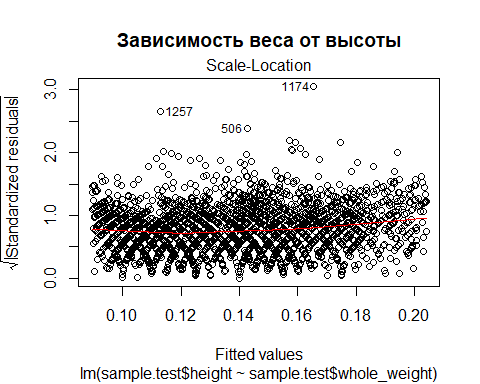
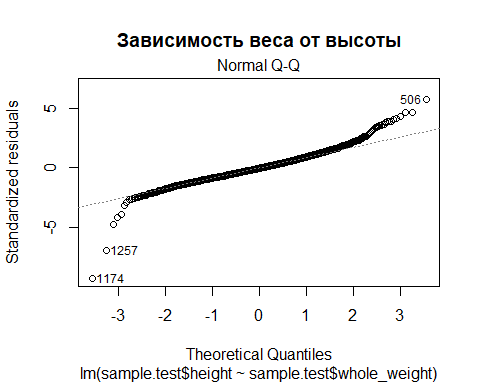
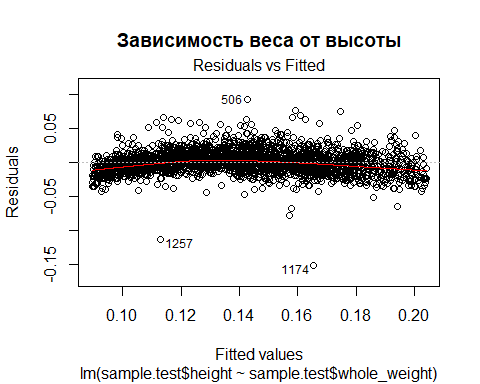
datalength <- nrow(datafrm.noout)  
testindex <- seq(1, trunc(datalength \* 0.7), by = 1)  
controlindex <- seq(round(datalength \* 0.3) + 1, datalength, by = 1)  
sample.test <- datafrm.noout[testindex,]  
sample.control <- datafrm.noout[controlindex,]

### подогнать модель по первой части

linear.model.d\_w <- lm(sample.test$diameter ~ sample.test$whole\_weight, data = sample.test)  
linear.model.h\_w <- lm(sample.test$height ~ sample.test$whole\_weight, data = sample.test)  
  
plot(linear.model.d\_w, main = "Зависимость веса от диаметра")



plot(linear.model.h\_w, main = "Зависимость веса от высоты")



### спрогнозировать (функция predict) значения во второй части

predicted.d\_w <- predict(linear.model.d\_w, sample.control)  
predicted.h\_w <- predict(linear.model.h\_w, sample.control)

### проверить качесвто прогноза

cor(sample.control$whole\_weight, predicted.d\_w)

## [1] -0.01330944

cor(sample.control$whole\_weight, predicted.h\_w)

## [1] -0.01330944

# Заключение

В ходе выполнения данной работы я получил незабываемый опыт и приобрел навыки, которые пригодятся в дальнейшей профессиональной деятельности. Были изучены и освоены на практике основы анализа данных. Прогноз исследуемых моделей получился достаточно точным.

С уважением,

студент гр. РИ-440005

Кабанов Евгений