Линейна регресия

Тема: Aerobic fitness

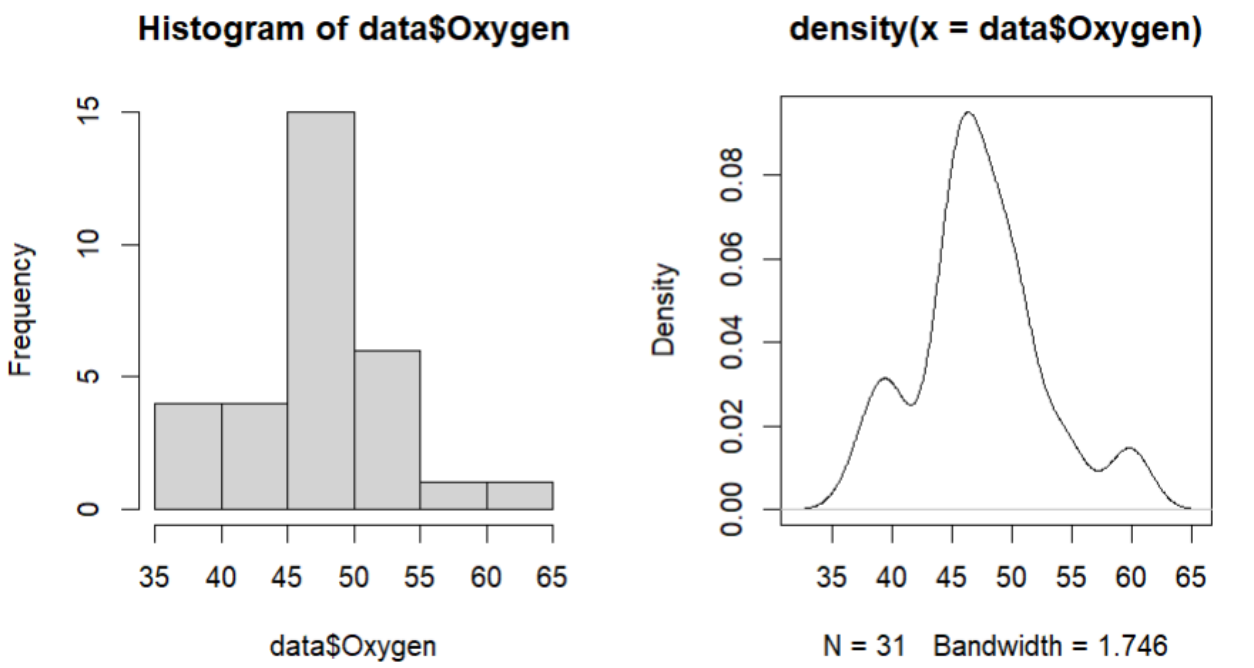
Изготвил: Кирил Романски

Цел: Да се разработи уравнение за предсказване на финеса (измерван чрез способността за приемане на кислород) на база упражнения, вместо ползването на скъпи и трудно изпълними измервания на кислородната консумация. Тези измервания са извършени върху мъже, участваши в курс по физическа подготовка.

Обработка и предварителен анализ на данни: Файлът с данни съдържа 7 променливи

Възраст-в години; Тегло-в килограми; Кислород-прием на кислород милилитри на килограм телесно тегло за минута; Време_на_бягане-време за завършване на 1.5 мили (в минути); Пулс_в_покой; Пулс_бягане; Максимален_Пулс.

Графики на разпределения и плъстности: Първо ще разгледаме разпределението на целевата променлива-Кислород, тъй като нейното разпределение е най-важно и е свързано с това какъв модел ще ползваме и хипотезите, който ще тестваме.



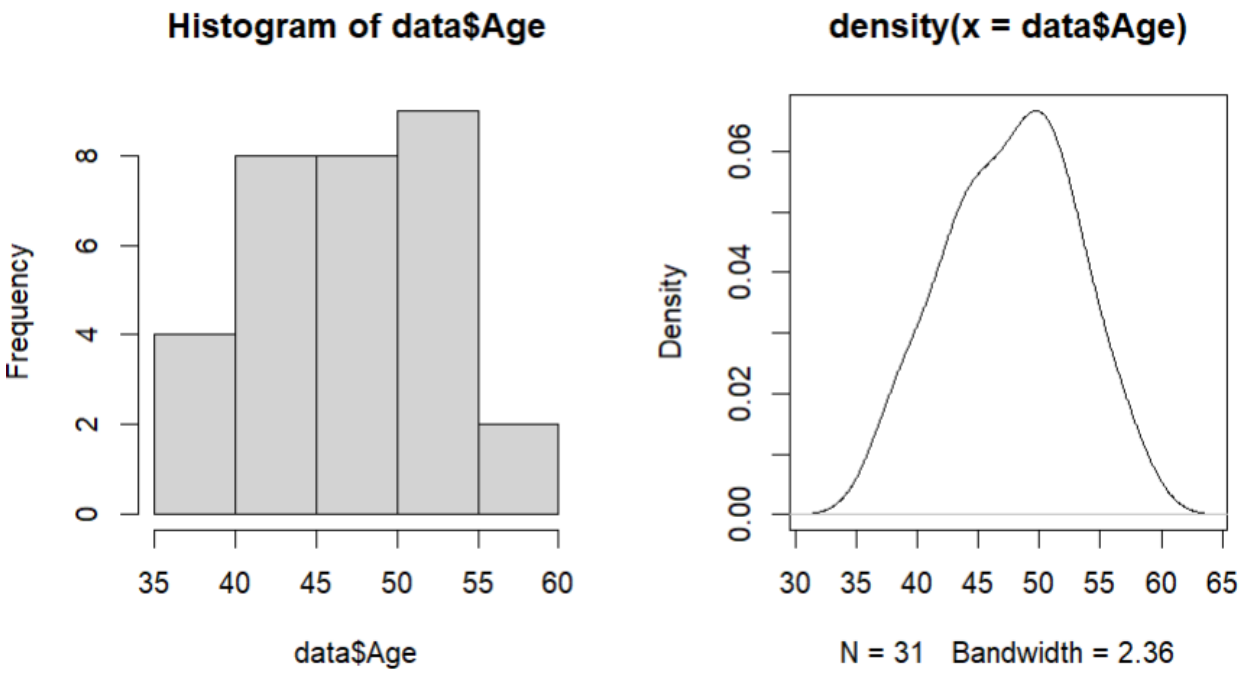
Разпределението изглежда нормално ще направим и Шапиро тест за всеки случай.

```
> shapiro.test(data$Oxygen)

        Shapiro-Wilk normality test

data:  data$Oxygen
W = 0.95366, p-value = 0.1968
```

п-стойността е голяма дори не близко до 0.05, тоест можем да приемем, че нашия отклик Килород е нормално разпределен.

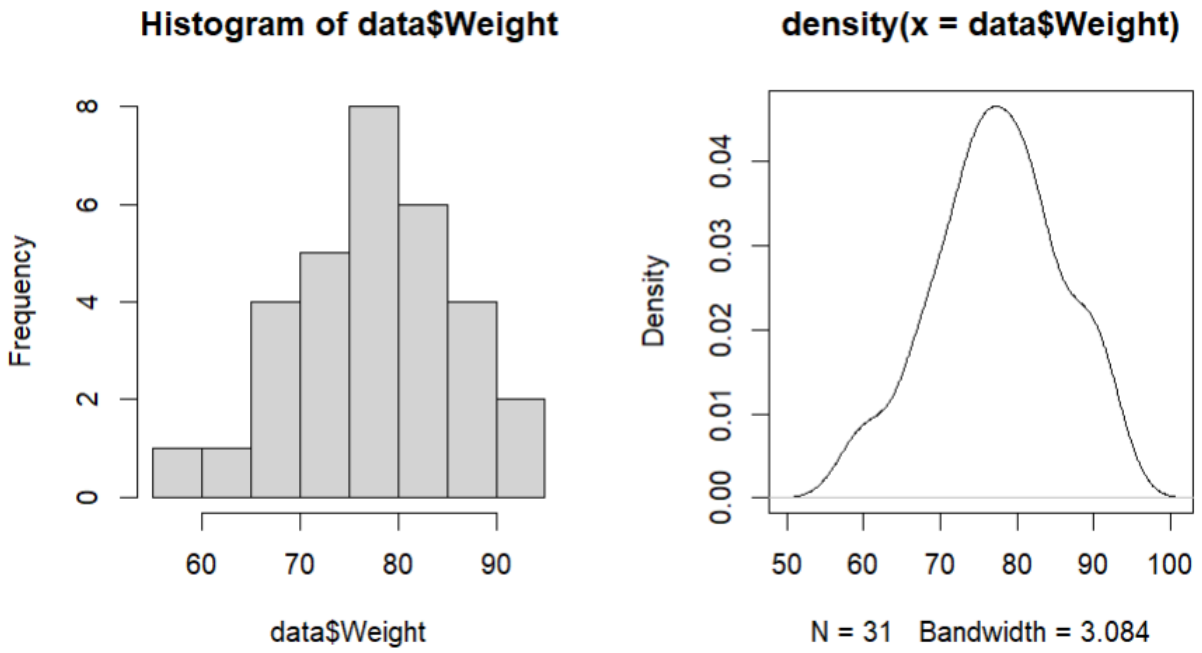
Останалите промеливи не е задължително да са нормално разпределени, но все пак е важно да видим тяхното разпределение и техните интервали, тъй като голяма разлика в интервала на данните може да изкриви нашия модел, и дали има голямо изместване.

Възраст:



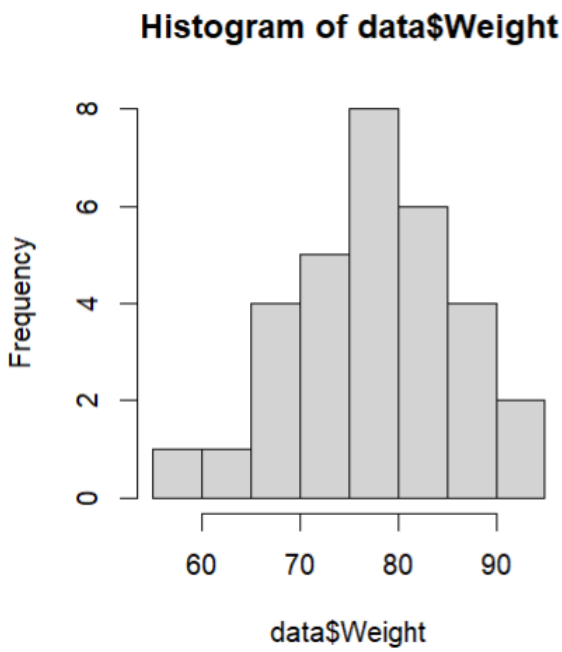
Разпределението изглежда нормално и няма аутлаяри или различна скала на измерванията

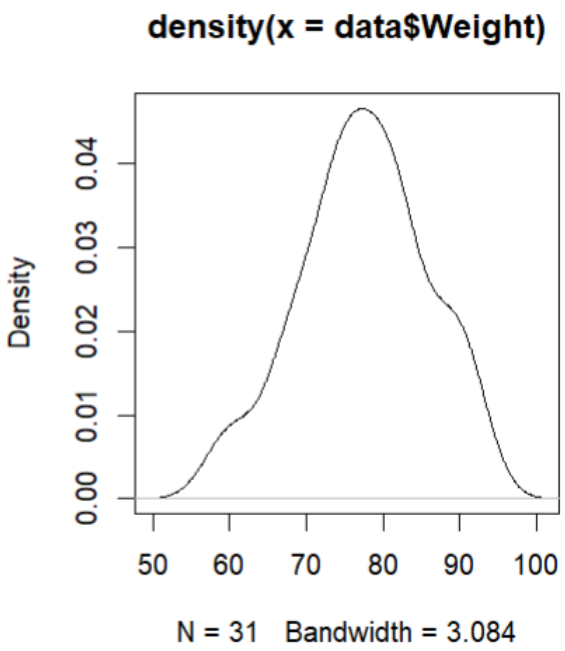
Тегло:



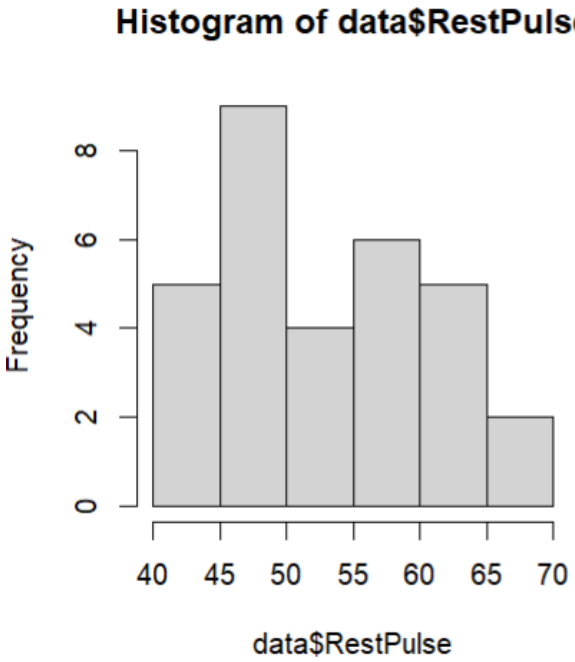
Приблизително нормално разпределение.

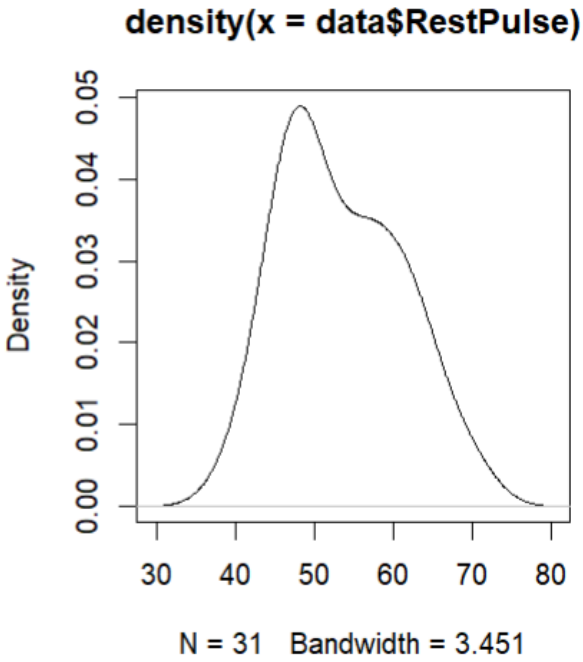
Време_на_бягане:


Histogram of data$Weight
Frequency
0
2
4
6
8
60
70
80
90
data$Weight


density(x = data$Weight)
Density
0.00
0.01
0.02
0.03
0.04
50
60
70
80
90
100
N = 31 Bandwidth = 3.084

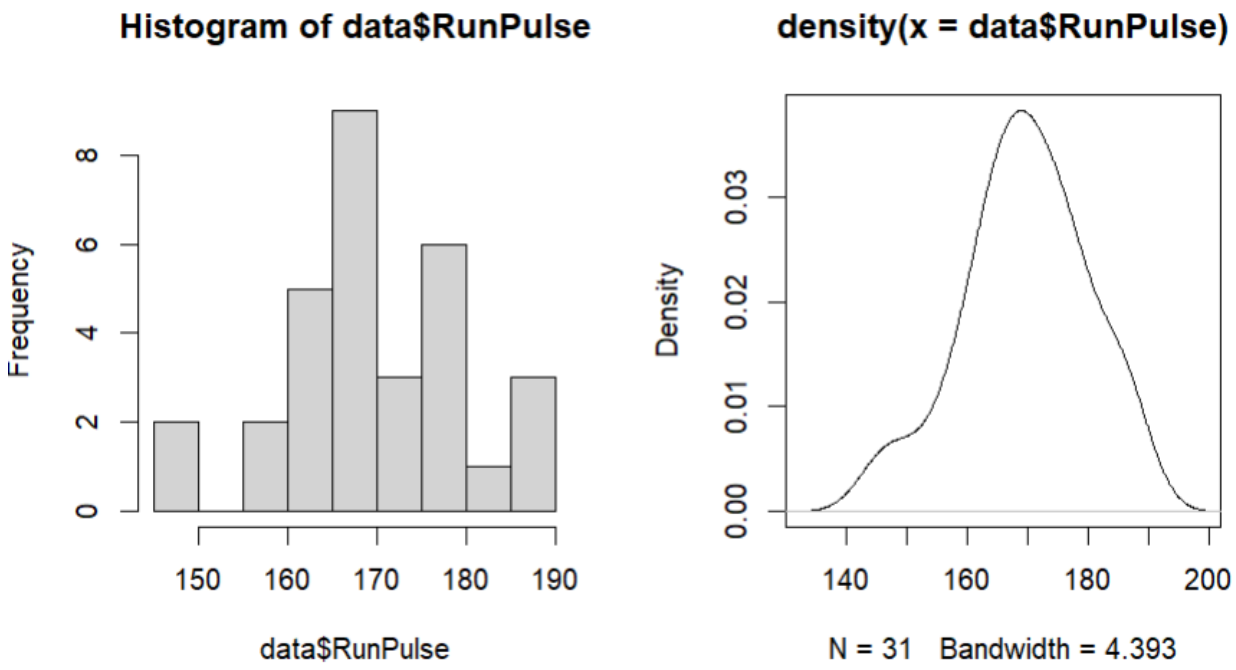
Пулс_в_покой:


Histogram of data$RestPulse
Frequency
0
2
4
6
8
40
45
50
55
60
65
70
data$RestPulse

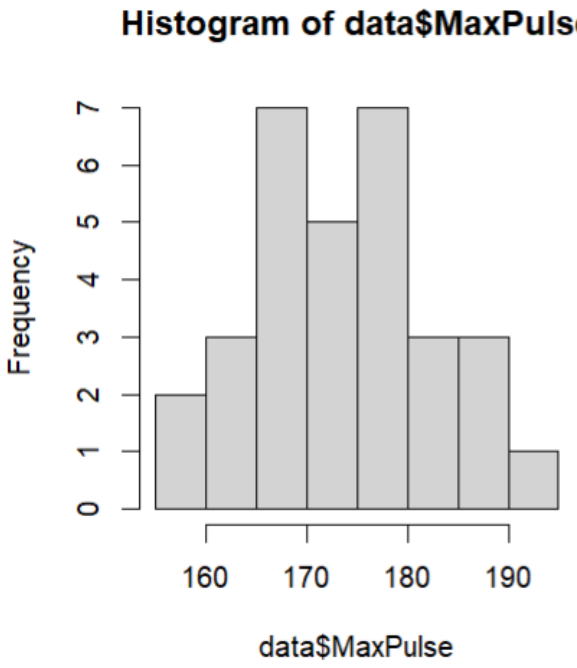

density(x = data$RestPulse)
Density
0.00
0.01
0.02
0.03
0.04
0.05
30
40
50
60
70
80
N = 31 Bandwidth = 3.451

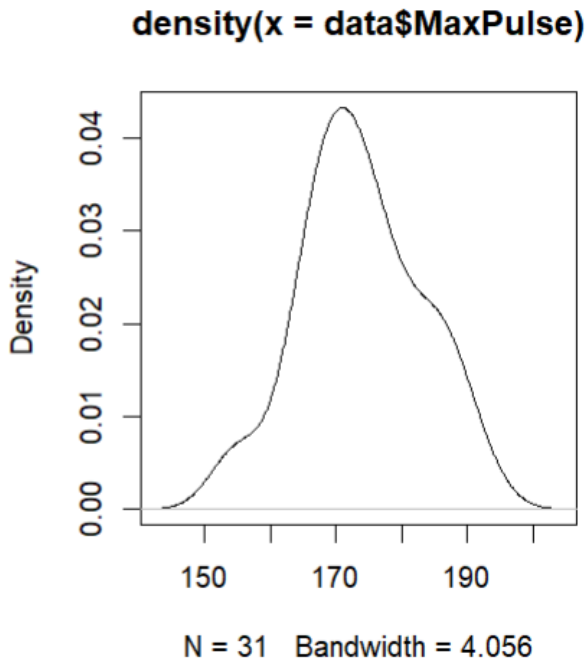
Тук можем да видим че разпределението не прилича на нормално, но това не е проблем, това е предикторна променлива.

Пулс_бягане:



Максимален пулс:





Тук хистограмата изглежда така защото сме дали повече стълбове. От плътността се вижда, че разпределението е приблизително нормално.

```
> summary(data)
      Age            Weight          Oxygen         RunTime        RestPulse       RunPulse        MaxPulse
 Min.   :38.00   Min.   :59.08   Min.   :37.39   Min.   : 8.17   Min.   :40.00   Min.   :146.0   Min.   :155.0
 1st Qu.:44.00   1st Qu.:73.20   1st Qu.:44.96   1st Qu.: 9.78   1st Qu.:48.00   1st Qu.:163.0   1st Qu.:168.0
 Median :48.00   Median :77.45   Median :46.77   Median :10.47   Median :52.00   Median :170.0   Median :172.0
 Mean   :47.68   Mean   :77.44   Mean   :47.38   Mean   :10.59   Mean   :53.45   Mean   :169.6   Mean   :173.8
 3rd Qu.:51.00   3rd Qu.:82.33   3rd Qu.:50.13   3rd Qu.:11.27   3rd Qu.:58.50   3rd Qu.:176.0   3rd Qu.:180.0
 Max.   :57.00   Max.   :91.63   Max.   :60.05   Max.   :14.03   Max.   :70.00   Max.   :186.0   Max.   :192.0
>
```

Стандартни отклонения на променливите:

```
      Age    Weight    Oxygen   RunTime RestPulse  RunPulse  MaxPulse
5.211443  8.328568  5.327231  1.387414  7.619443 10.251986  9.164095
```

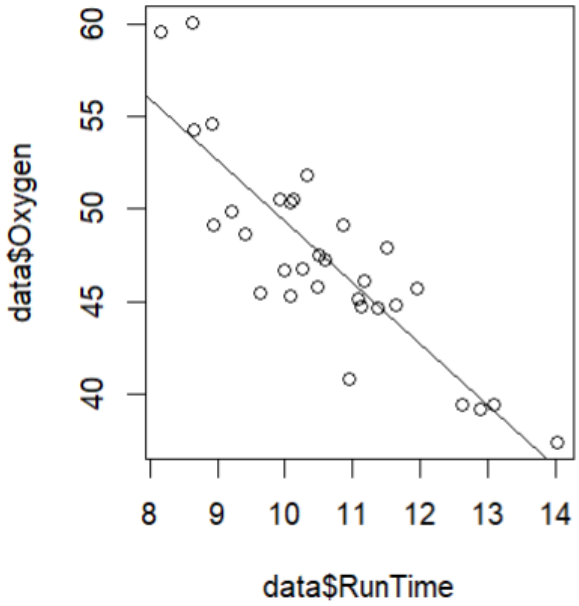
Корелационна матрица:

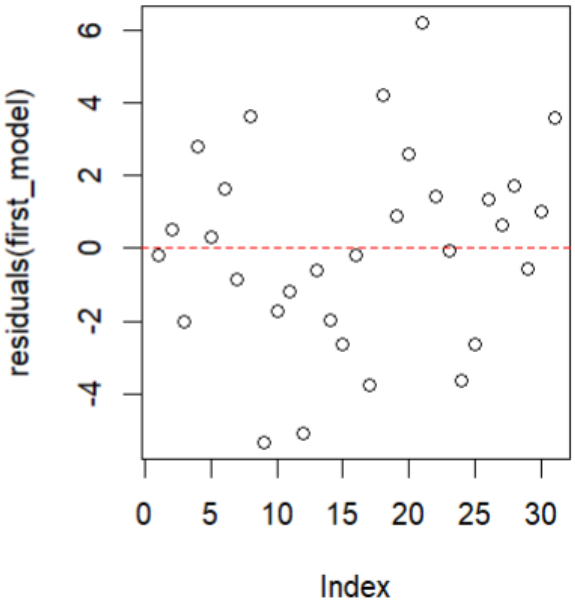
```
> cor(data)
                Age      Weight     Oxygen    RunTime   RestPulse   RunPulse   MaxPulse
Age        1.0000000 -0.23353903 -0.3045924  0.1887453 -0.16409995 -0.3378703 -0.4329159
Weight    -0.2335390  1.00000000 -0.1627528  0.1435076  0.04397417  0.1815163  0.2493812
Oxygen    -0.3045924 -0.16275285  1.0000000 -0.8621949 -0.39935611 -0.3979742 -0.2367402
RunTime    0.1887453  0.14350758 -0.8621949  1.0000000  0.45038260  0.3136478  0.2261030
RestPulse -0.1640999  0.04397417 -0.3993561  0.4503826  1.00000000  0.3524606  0.3051240
RunPulse  -0.3378703  0.18151633 -0.3979742  0.3136478  0.35246060  1.0000000  0.9297538
MaxPulse  -0.4329159  0.24938123 -0.2367402  0.2261030  0.30512400  0.9297538  1.0000000
```

Силна корелация между Максималния пулс и Пулса по време на бягане, което е съвсем логично, когато правим модела ще проверим кое ни носи повече информация и ще премахнем другото, за да нямаме мултиколинеарност. Друга силна корлация можем да видим между кислорода и Времето_за_бягане тоест можем да очакваме, че това ще бъде важна променлива за нашия модел. Направих и модел само с тази промелнива като начална отправна точка на нашия експеримент.

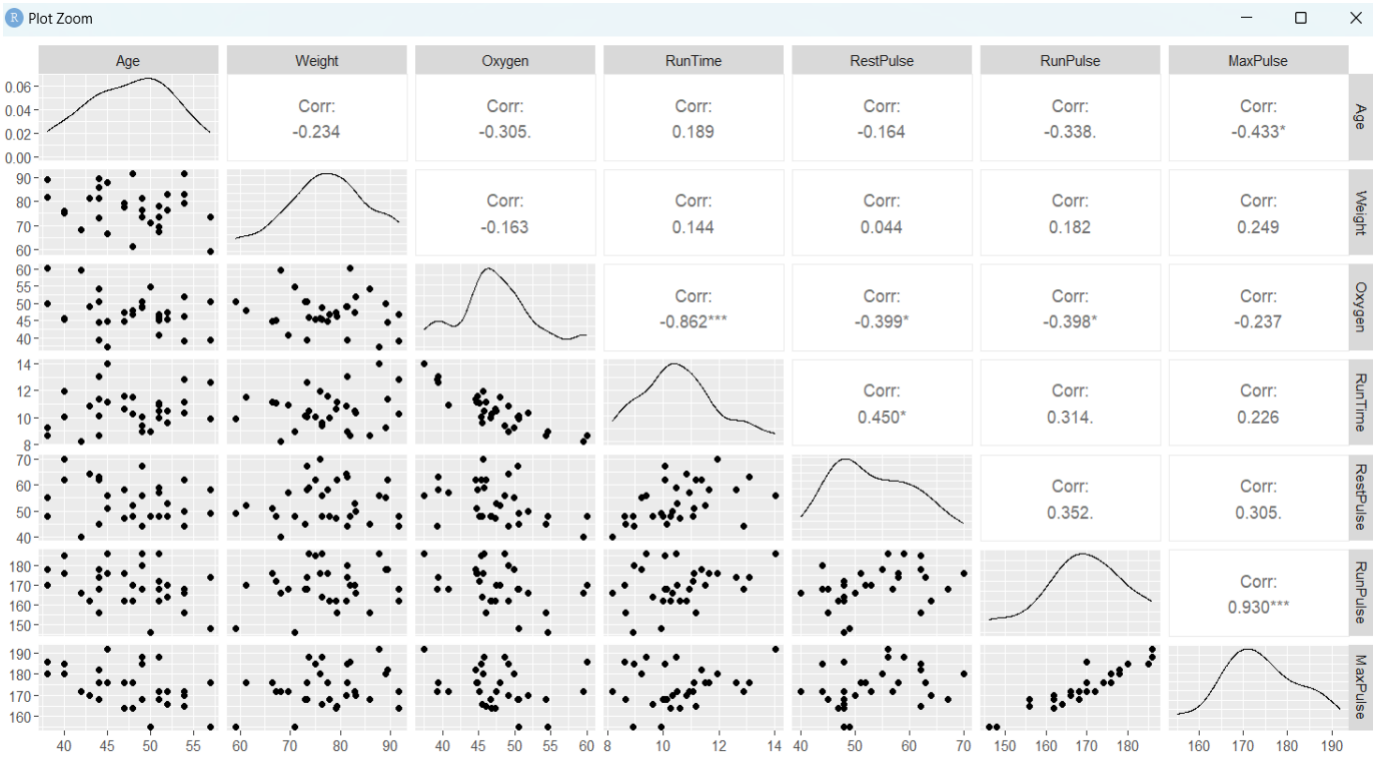
```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  82.4218     3.8553  21.379  < 2e-16 ***
RunTime      -3.3106     0.3612  -9.166 4.59e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.745 on 29 degrees of freedom
Multiple R-squared:  0.7434,    Adjusted R-squared:  0.7345
F-statistic: 84.01 on 1 and 29 DF,  p-value: 4.585e-10

>
```

И както виждаме този модел се представя доста добре дори само с 1 предиктор.





Скатер плот на променливите с Кислород. Идеята на тази графика е да видим дали някъде имаме нелинейна зависимост между отклика и предикторите и да видим дали са нужди някакви трансформации.

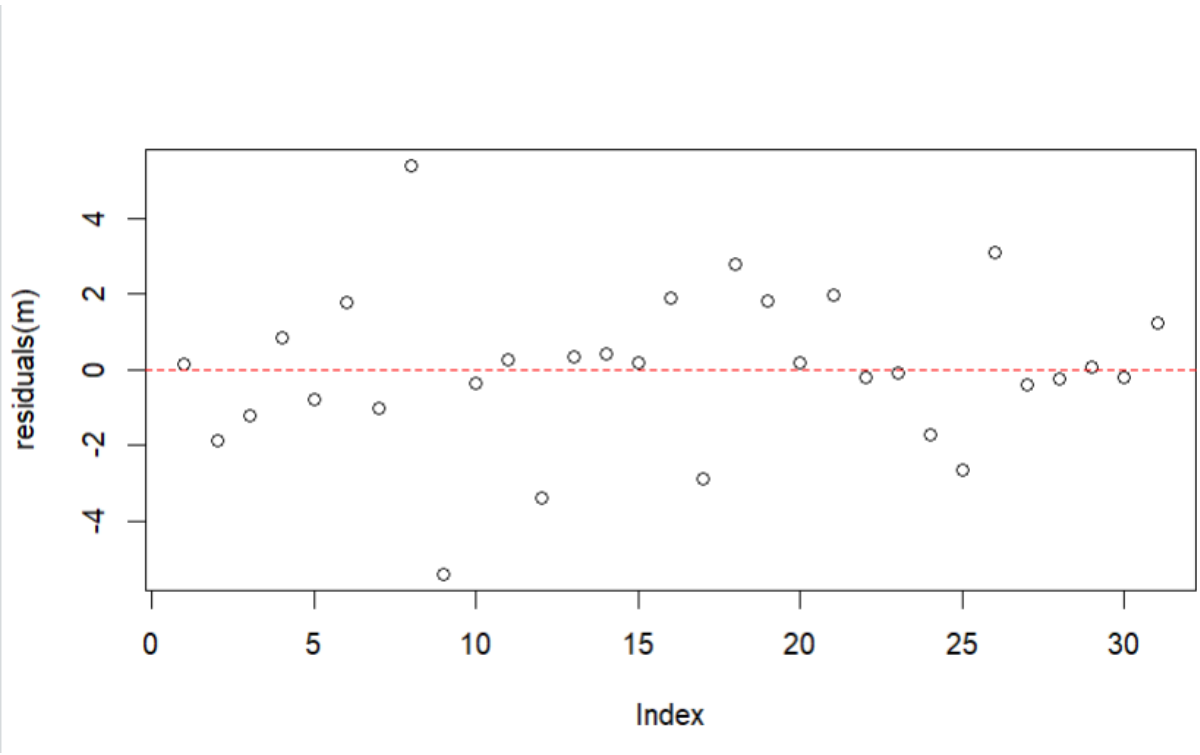


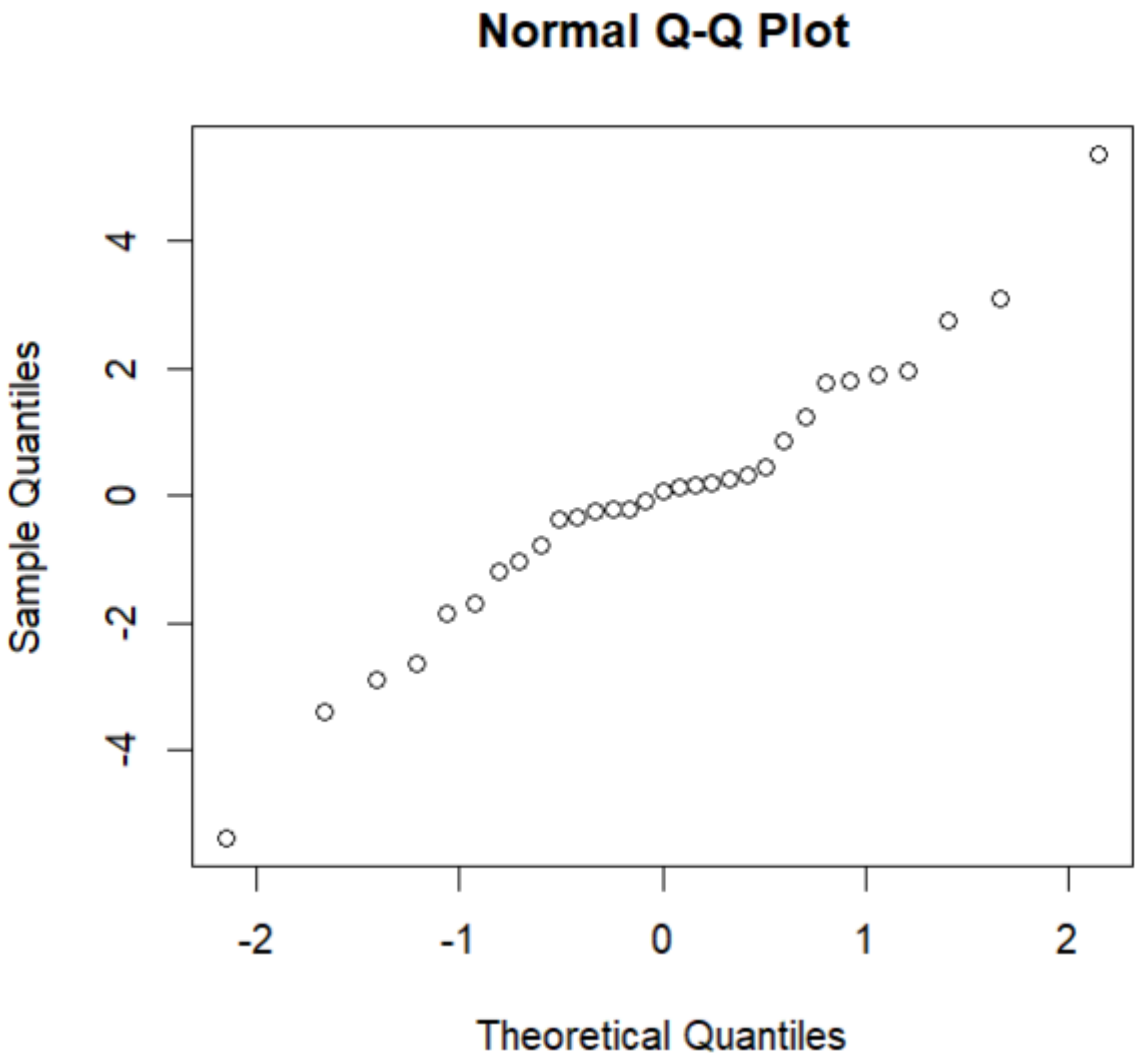
Не мога да видя някаква зависимост нелинейна между Кислорода и останалите променливи, така че можем да започнем със изграждане на модела.

Първи модел: За начало ще използваме всички предиктори и ще направим пълен модел

```
> m=lm(Oxygen~.,data=data)
> summary(m)

Call:
lm(formula = Oxygen ~ ., data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-5.4026 -0.8991  0.0706  1.0496  5.3847

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 102.93448   12.40326   8.299 1.64e-08 ***
Age          -0.22697    0.09984  -2.273  0.03224 *
Weight       -0.07418    0.05459  -1.359  0.18687
RunTime      -2.62865    0.38456  -6.835 4.54e-07 ***
RestPulse    -0.02153    0.06605  -0.326  0.74725
RunPulse     -0.36963    0.11985  -3.084  0.00508 **
MaxPulse      0.30322    0.13650   2.221  0.03601 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.317 on 24 degrees of freedom
Multiple R-squared:  0.8487,    Adjusted R-squared:  0.8108
F-statistic: 22.43 on 6 and 24 DF,  p-value: 9.715e-09

>
```

Модела се представя добре има висок коефициент на детерминация-0.81, но не твърде висок. Има предиктори, които могат да бъдат премахнати, но за първи се представя добре. Имайки предвид, че нашия експеримент е само с 31 наблюдения целта ни ще бъде около 3 предиктора. Нека сега проверим остатъците и дали модела е правилен.


residuals(m)
4
2
0
-2
-4
0
5
10
15
20
25
30
Index



Остатъците изглеждат нормално разпределени. Не се вижда някаква зависимост в от графиката.

VIF на предикторите: Ще пресметнем мултиколинеарност използвайки variance inflation factor, който се смята по формулата $Vif_{xj}=1/1-R_j$ където R_j е коеф. на детерминация на линеен модел фитнат върху j-тия предиктор. Идеята е да тестваме каква информация ни носи съответния предитор спрямо линейна комбинация на останалите, което простата корелация не може да оцени. Имайки предвид, че корелацията между MaxPulse и RunPulse

е голяма очакваме големи стойности на vif за тези две променливи всичко над 5 ще считаме за мултиколинеарност

```
> vif(m)
      Age    Weight   RunTime RestPulse  RunPulse  MaxPulse
 1.512836  1.155329  1.590868  1.415589  8.437274  8.743848
>
```

И както предполагахме RunPulse и MaxPulse имат >5 стойност на vif. Ще поправим този проблем като махнем една от променливите, но първо ще направим стъпкова регресия, за да видим оптималния модел.

Стъпкова регресия: С функцията step(model, direction="backward") ще направим премахване на слаби предиктори. Посоката е backward понеже вече имаме целия модел и ще махаме.

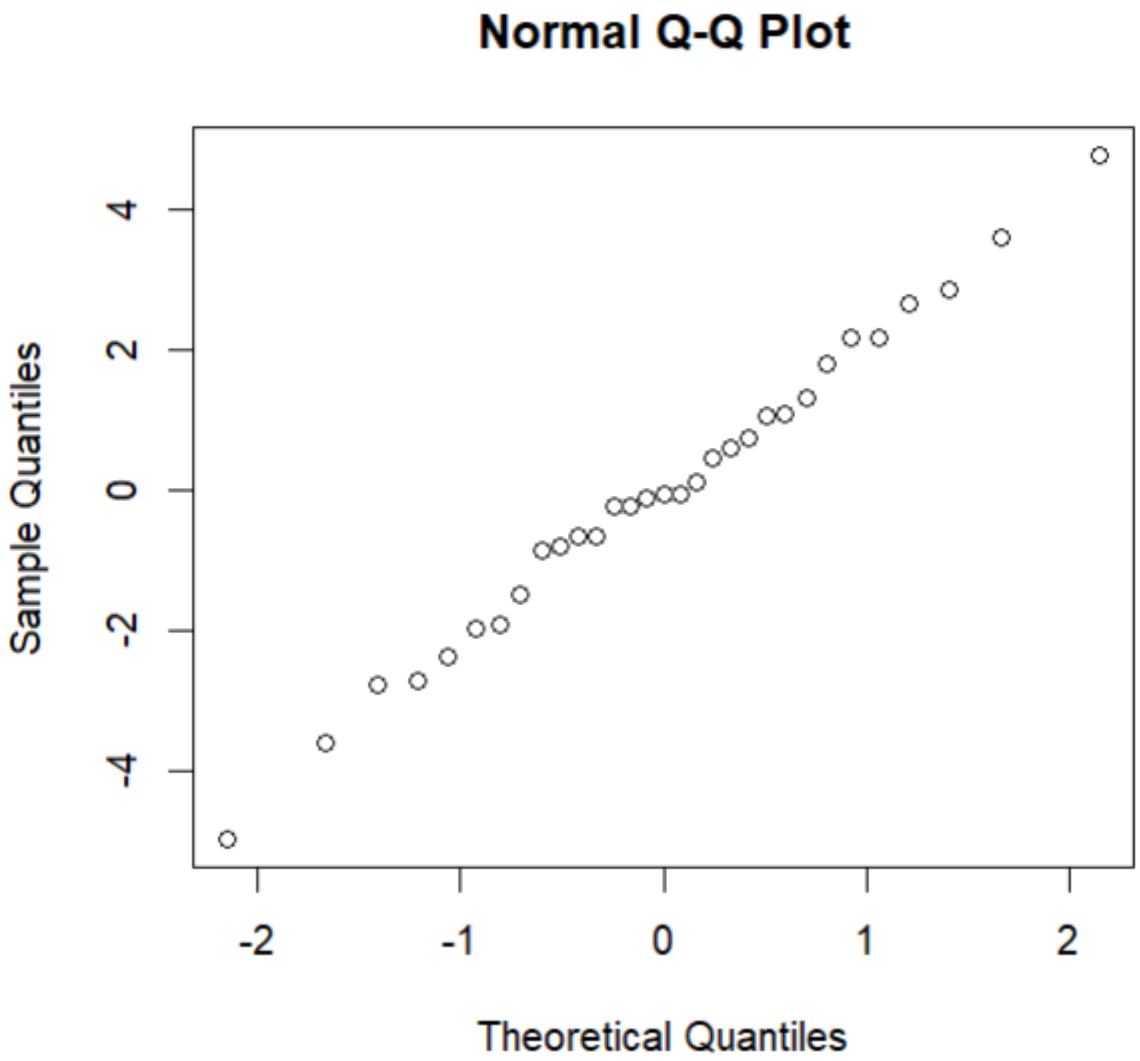
```
> stepwise_reg=step(m, direction = "backward")
Start:  AIC=58.16
Oxygen ~ Age + Weight + RunTime + RestPulse + RunPulse + MaxPulse

            Df Sum of Sq    RSS    AIC
- RestPulse  1     0.571 129.41 56.299
<none>                   128.84 58.162
- Weight     1     9.911 138.75 58.459
- MaxPulse   1    26.491 155.33 61.958
- Age        1    27.746 156.58 62.208
- RunPulse   1    51.058 179.90 66.510
- RunTime    1   250.822 379.66 89.664

Step:  AIC=56.3
Oxygen ~ Age + Weight + RunTime + RunPulse + MaxPulse

           Df Sum of Sq    RSS    AIC
<none>                  129.41 56.299
- Weight    1      9.52 138.93 56.499
- MaxPulse  1     26.83 156.23 60.139
- Age       1     27.37 156.78 60.247
- RunPulse  1     52.60 182.00 64.871
- RunTime   1    320.36 449.77 92.917
>
```

Виждаме, че step функцията махна само 1 предиктор, но това не означава, че модела е добър. AIC функцията представлява: AIC=2k−2ln(L) където к са параметрите, а L максимизиран likelihood. Тоест теоритично тази функция оценява, колко добре данните фитват модела

като има наказание за повече предиктори. Това обаче не контролира дали предикторите са мултиколинеарни или не. За това сами ще махаме предиктори и, ако не свалим много R квадрата ще даваме приоритет на модел с по-малко предиктори. Махаме предитора Тегло(Weight) и предиктора RestPulse, тъй като имат големи п-стойности.

```
> m_reduced=lm(Oxygen~Age+RunTime+RunPulse+MaxPulse,data=data)
> summary(m_reduced)

Call:
lm(formula = Oxygen ~ Age + RunTime + RunPulse + MaxPulse, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-4.9685 -1.1654 -0.0636  1.2004  4.7726

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 98.14789   11.78569   8.328 8.26e-09 ***
Age         -0.19773    0.09564  -2.068  0.04877 *
RunTime     -2.76758    0.34054  -8.127 1.31e-08 ***
RunPulse    -0.34811    0.11750  -2.963  0.00644 **
MaxPulse     0.27051    0.13362   2.024  0.05330 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.312 on 26 degrees of freedom
Multiple R-squared:  0.8368,    Adjusted R-squared:  0.8117
```

Добре от модела се вижда, че r-квадрат почти не падна, но за сметка на това модела е по-прост и остатъците изглеждат по-добре като ги начертаем срещу теоритичните квантили.



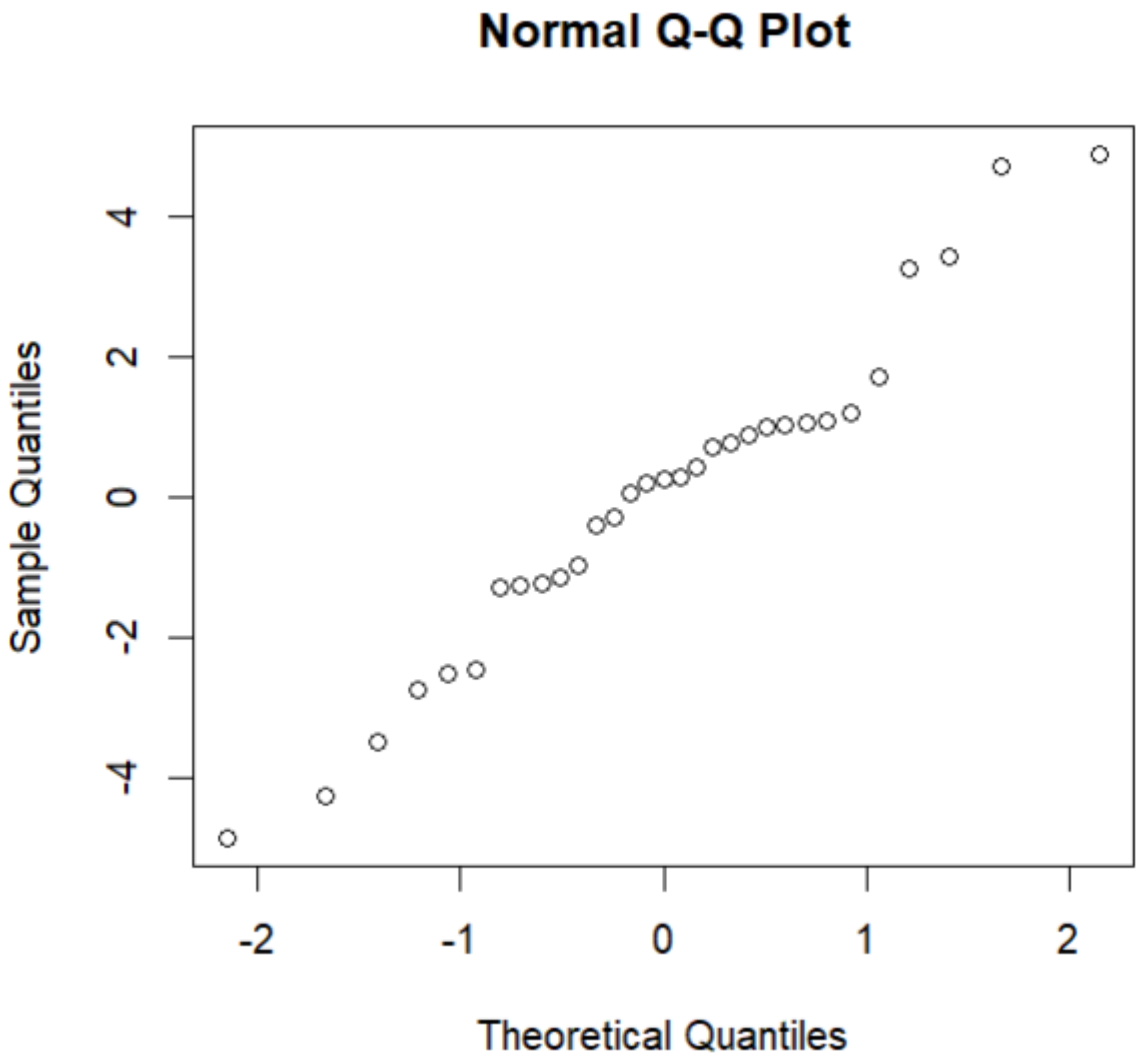
Остана само да пробваме да махнем мултиколинеарността от модела, което ще направим като премахнем предиктора MaxPulse. Защо него? Ами той има по висока п-стойност и реално е незначим за ниво на доверие 0.95.

Модел без MaxPulse:

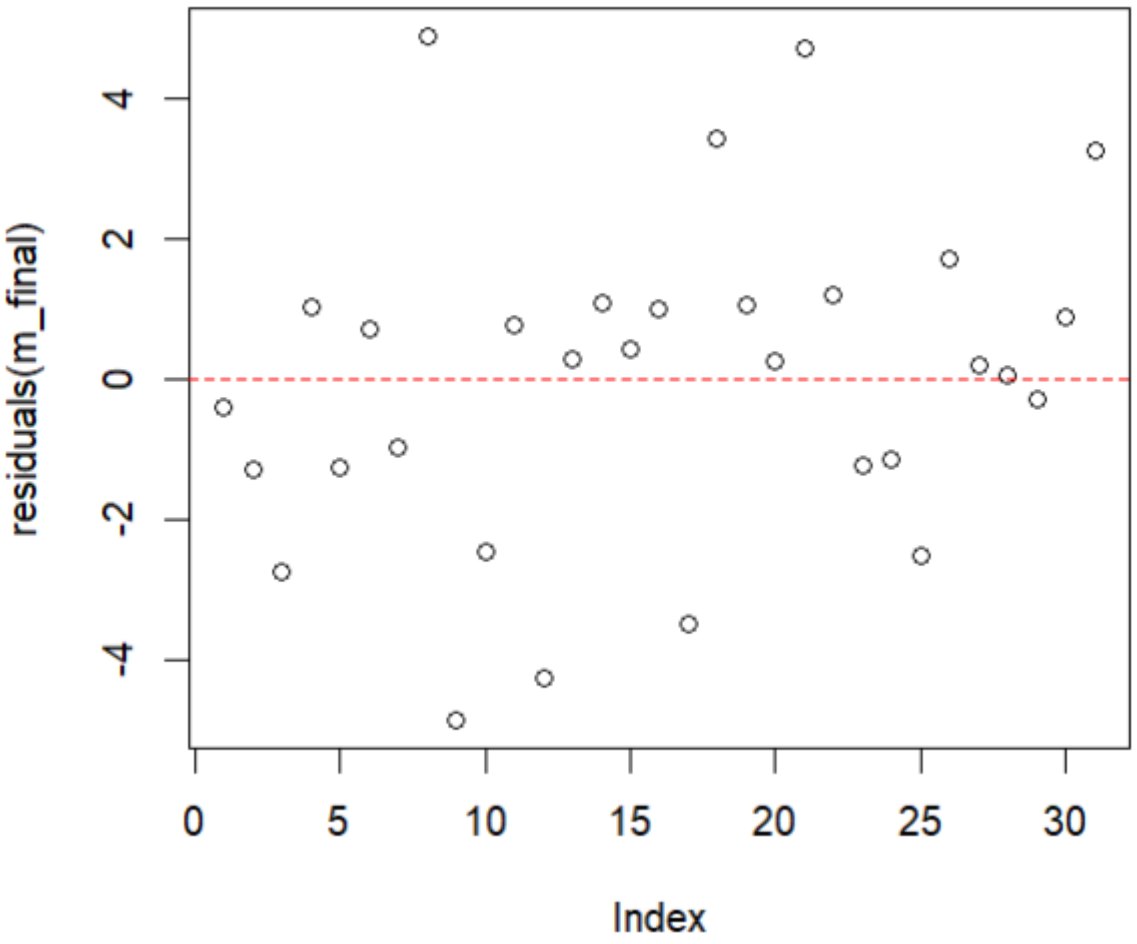
Oxygen~Age+RunTime+RunPulse

```
> summary(m_final)

Call:
lm(formula = Oxygen ~ Age + RunTime + RunPulse, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-4.8752 -1.2493  0.2606  1.0324  4.8994

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 111.71806   10.23509  10.915 2.10e-11 ***
Age          -0.25640    0.09623  -2.664   0.0129 *
RunTime      -2.82538    0.35828  -7.886 1.77e-08 ***
RunPulse     -0.13091    0.05059  -2.588   0.0154 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.441 on 27 degrees of freedom
Multiple R-squared:  0.8111,    Adjusted R-squared:  0.7901
F-statistic: 38.64 on 3 and 27 DF,  p-value: 6.557e-10

>
```

Малко се намалява коеф. на детерминация, но това не стига, за да използваме по-сложен модел.



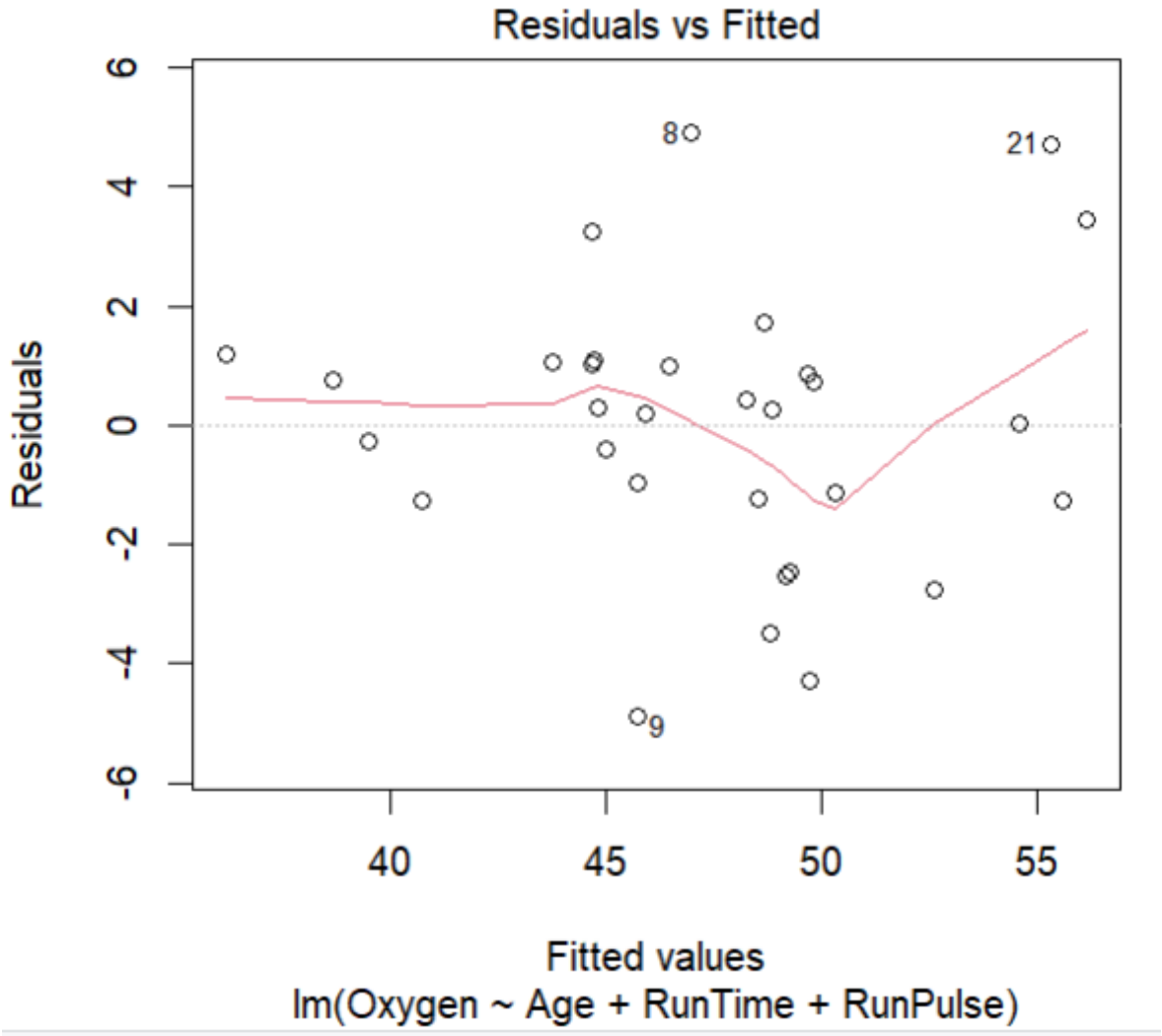
Тук обаче виждаме проблем, остатъците се влошиха, което значи, че може някое от преположенията, които имаме да са нарушени ще плотнем и остатъците срещу нулата да видим дали там ще намерим някаква зависимост.



Изглеждат добре, не се вижда зависимост, нека проверим дали някое наблюдение не дърпа целия модел надолу. Ще сметнем дали някой от остатъците е на повече от 2 стандартни отклонения. Първо ги стандартизираме с which функцията ще намерим индекса на даденото наблюдение.

```
> which(abs(rstandard(m_final)) > 2*sd(rstandard(m_final)))
 8  9 21
```

Нека видим тези аутлаяри срещу фитнати стойности



Нека пробваме модела без тези наблюдения, да видим как ще се представи.

```
> m_final_cleaned=lm(Oxygen~Age+RunTime+RunPulse,data=data_cleaned)
> summary(m_final_cleaned)

Call:
lm(formula = Oxygen ~ Age + RunTime + RunPulse, data = data_cleaned)

Residuals:
    Min      1Q  Median      3Q     Max
-4.0419 -0.8947  0.1947  1.0287  4.3669

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 107.41024    8.25276  13.015 2.29e-12 ***
Age          -0.21779    0.08100  -2.689  0.01283 *
RunTime      -2.60652    0.28366  -9.189 2.50e-09 ***
RunPulse     -0.13111    0.03933  -3.333  0.00278 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.894 on 24 degrees of freedom
Multiple R-squared:  0.862,     Adjusted R-squared:  0.8448
F-statistic: 49.98 on 3 and 24 DF,  p-value: 1.791e-10
```

Финалния модел без аутлаяри. Коеф. на детерминация скочи до 0.86, което е много добре. Редуцирания модел в който има Максималния пулс също се представя подобно, малко по-добре и остатъците изглеждат по-добре.

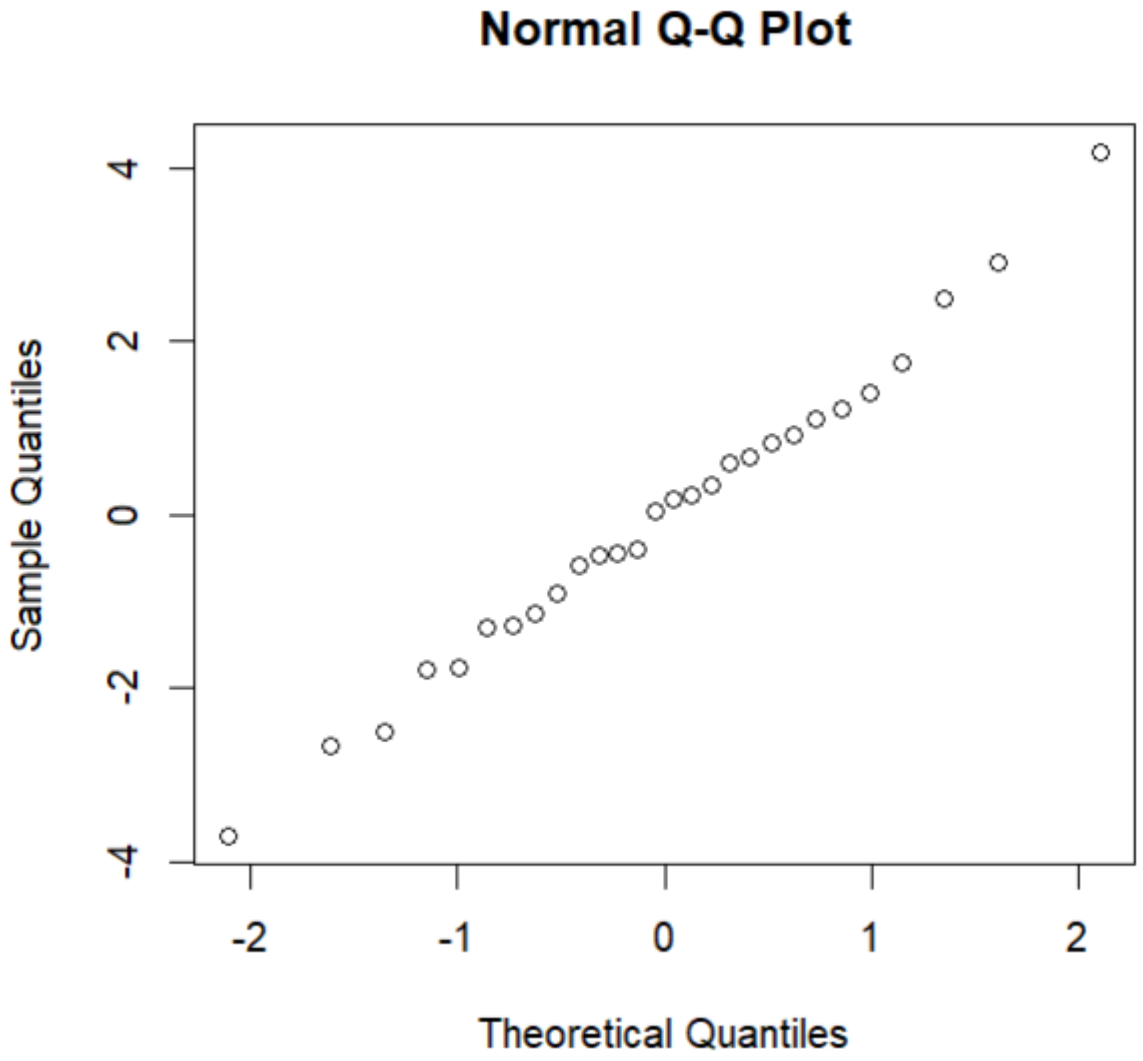
Финални проверки: Тъй като моделите се представят доста добре ни остана да проверим едно нещо: Кой е по-добър? Модела с MaxPulse има по ниска AIC стойност и по-добри показатели като коеф. На детерминация и остатъците се съпоставят по-добре към нормално разпределение, но има един проблем-мултиколинеарност между предикторите. Финалния модел без MaxPulse няма този проблем, но се преставя малко по-зле. Тъй като нямаме много данни (само 31) единствения вариант, който можем да използваме за някакъв вид валидация на моделите е крос валидация LOOCV. Ще тренираме модела на 30 точки и ще предвиждаме 1-та останала. Така ще имаме 31 остатъка и ще сравним моделите като този с по-малка сума на грешките ще бъде финалния ни модел.

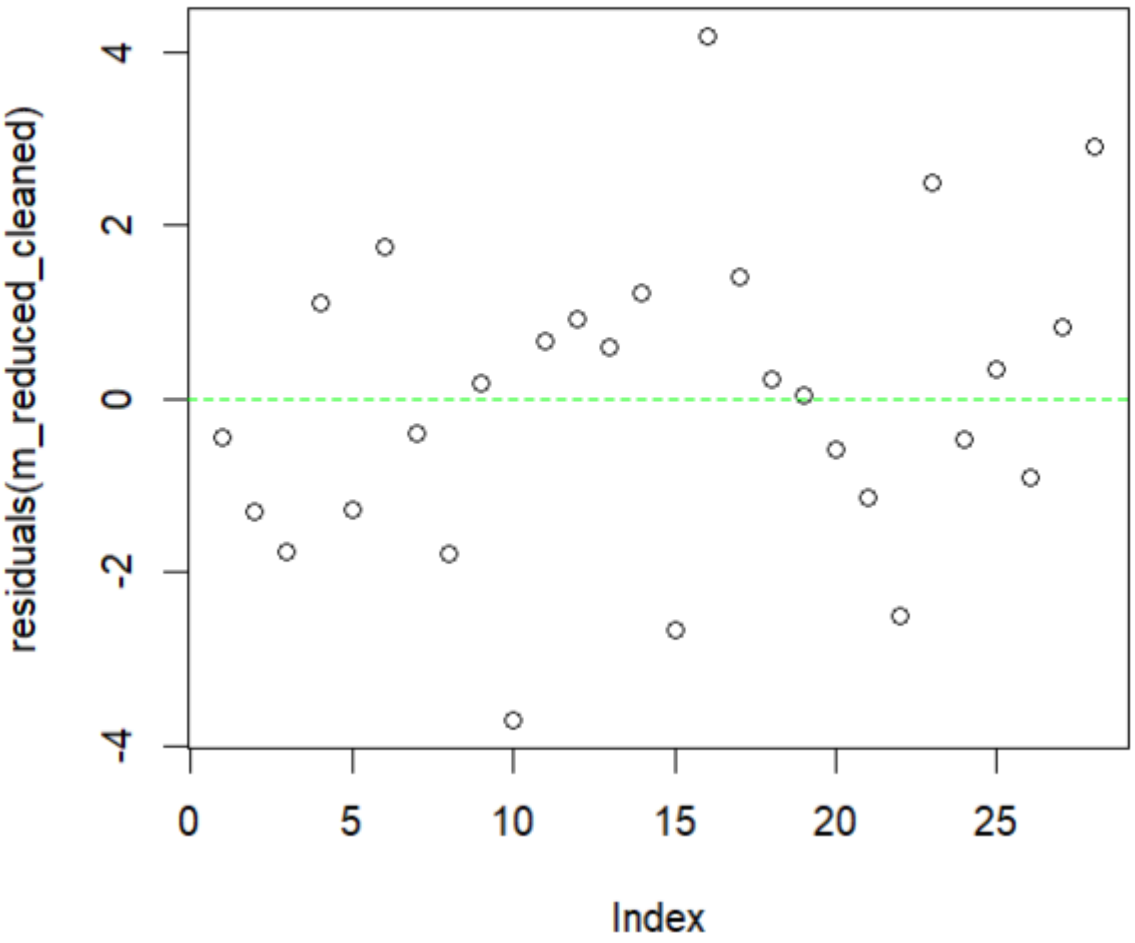
```
> cat( LOOCV предикшън грешка на пълния модел: ", error_full)
LOOCV предикшън грешка на пълния модел: 6.083837
> cat("LOOCV предикшън грепка на редуцирания модел:", error_reduced)
LOOCV предикшън грепка на редуцирания модел: 6.616924
>
```

Излиза, че все пак мултиколинеарността няма толкова голямо значение и модела включващ MaxPulse е по-добър.

Заключение: Финалния модел е Oxygen~Age+RunTime+RunPulse+MaxPulse. Този модел е удачен, защото както видяхме по-горе връзката между отклика и предикторите изглежда линейна, никъде не присъства квадратична или кубична зависимост. Използвах линейна регресия и поради причината, че е лесно изчислима и лесно може да се интерпретира резултата. Разпределението на остатъците е нормално, което е нашето най-важно допускане и то е изпълнено. R квадрата висок 0.87, което допълнително показва устойчивостта на модела и колко са сполучливи нашите предиктори. За жалост нямаме достатъчно данни, за да направим качествена диагностика. Най-доброто, за което се сетих е leave one out крос валидация, която има стоите недостатъци е добър алгоритъм за валидиране, с чиято помощ и решихме кой да е финалния модел.

```
Call:
lm(formula = Oxygen ~ Age + RunTime + RunPulse + MaxPulse, data = data_cleaned)

Residuals:
    Min      1Q  Median      3Q     Max
-3.7121 -1.1721  0.1139  0.9778  4.1953

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 101.91653    9.84759  10.349 3.98e-10 ***
Age          -0.20525    0.08186  -2.508   0.0197 *
RunTime      -2.61593    0.28357  -9.225 3.42e-09 ***
RunPulse     -0.24485    0.11819  -2.072   0.0497 *
MaxPulse      0.14010    0.13729   1.020   0.3181
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.892 on 23 degrees of freedom
Multiple R-squared:  0.868,     Adjusted R-squared:  0.8451
F-statistic: 37.81 on 4 and 23 DF,  p-value: 8.452e-10
```


Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
4
2
0
-2
-4
-2
-1
0
1
2



Код:

```r
#######
#######
###### SECOND PROJECT IN CASE DATASET 8 AIR POLUTION !!!

list.files()
install.packages("sas7bdat")
library("sas7bdat")
data=read.sas7bdat("C:\\Users\\kiril\\Downloads\\fitness.sas7bdat")
summary(data)
data
par(mfrow = c(1, 2))  # 1 row, 2 columns (side by side)

hist(data$Oxygen)
plot(density(data$Oxygen))
hist(data$Age)
plot(density(data$Age))
hist(data$Weight)
plot(density(data$Weight))
hist(data$RunTime)
plot(density(data$RunTime))
hist(data$RestPulse)
plot(density(data$RestPulse))
hist(data$RunPulse)
plot(density(data$RunPulse))
hist(data$MaxPulse)
plot(density(data$MaxPulse))
install.packages("GGally")
library(GGally)
ggpairs(data)
plot(data$RunTime, data$Oxygen)
abline(lm(Oxygen ~ RunTime, data = data))
plot(data$Oxygen,data$Weight)
first_model=lm(Oxygen~RunTime,data=data)
summary(first_model)
plot(residuals(first_model))
abline(h=0,col="red",lty=2)

sapply(data, sd)
summary(data)
```

```r
summary(data)
shapiro.test(data$MaxPulse)
cor(data)
m=lm(Oxygen~.,data=data)
summary(m)
plot(residuals(m))
abline(h=0,col="red",lty=2)
qqnorm(residuals(m))

library(car)
vif(m)
stepwise_reg=step(m, direction = "backward")
summary(stepwise_reg)
m_reduced=lm(Oxygen~Age+RunTime+RunPulse+MaxPulse,data=data)
summary(m_reduced)
qqnorm(residuals(m_reduced))


m_final=lm(Oxygen~Age+RunTime+RunPulse,data=data)
summary(m_final)
plot(m_final)
plot(rstandard(m_final))
qqnorm(residuals(m_final))
abline(h=0, col="red",lty=2)
?which
max(abs(residuals(m_final)))
sd(residuals(m_final))
which(abs(rstandard(m_final)) > 2*sd(rstandard(m_final)))
###Имаме 3 аутлаяри 8,9,21 нека проверим
shapiro.test(residuals(m_final))
###Остатъците са нормално разпределени
hist(residuals(m_final))
anova(stepwise_reg,m_final)### Резултатът е 0.06 като п-стойност което е на границата, но
##не можем да кажем, че по-комплекния модел е по-добър
length(data$Oxygen)
data_cleaned=data[-c(8,9,21),]
m_reduced_cleaned=lm(Oxygen~Age+RunTime+RunPulse+MaxPulse,data=data_cleaned)
summary(m_reduced_cleaned)
qqnorm(residuals(m_reduced_cleaned))
```

```r
length(data$Oxygen)
data_cleaned=data[-c(8,9,21),]
m_reduced_cleaned=lm(Oxygen~Age+RunTime+RunPulse+MaxPulse,data=data_cleaned)
summary(m_reduced_cleaned)
qqnorm(residuals(m_reduced_cleaned))
m_final_cleaned=lm(Oxygen~Age+RunTime+RunPulse,data=data_cleaned)
summary(m_final_cleaned)
plot(m_final_cleaned)
plot(rstandard(m_final_cleaned))
qqnorm(residuals(m_final_cleaned))
abline(h=0, col="red",lty=2)
length(data_cleaned$Weight)

vif(m_final)
if (!require(boot)) install.packages("boot")
data
library(boot)

loocv_error <- function(data, formula) {
  glm_fit <- glm(formula, data = data)
  cv_result <- cv.glm(data, glm_fit, K = nrow(data))
  return(cv_result$delta[1])  # чиста крос валидация за пресмятане на грешката
}

formula_full <- Oxygen ~ Age + RunTime + RunPulse + MaxPulse
formula_reduced <- Oxygen ~ Age + RunTime+ RunPulse

error_full <- loocv_error(data, formula_full)
error_reduced <- loocv_error(data, formula_reduced)


## тука ползвам cat вместо принт просто, за да излезе и надписа concatenate and print
cat("LOOCV предикшън грешка на пълния модел:", error_full)
cat("LOOCV предикшън грепка на редуцирания модел:", error_reduced)

plot(m_reduced_cleaned)
plot(residuals(m_reduced_cleaned))
abline(h=0, col="green",lty=2 )
```