

1. Перейти от нелинейной модели к линейной и выполнить ее анализ.

```
x <- c(9.35, 16.72, 21.87, 27.74, 32.95, 38.97,
      45.87, 52.32, 59.32, 65.57, 75.75)
y <- c(7.67, 11.83, 13.76, 15.48, 16.99, 17.74,
      18.41, 19.18, 19.88, 20.5, 21.34)
data <- data.frame(x,y)
z <- sqrt(x)
mod <- lm(y ~ z)
summary(mod)
print(data)
plot(data,type="p", col="blue", xlab="переменная x", ylab="переменная y")
```
Call:
lm(formula = y ~ z)

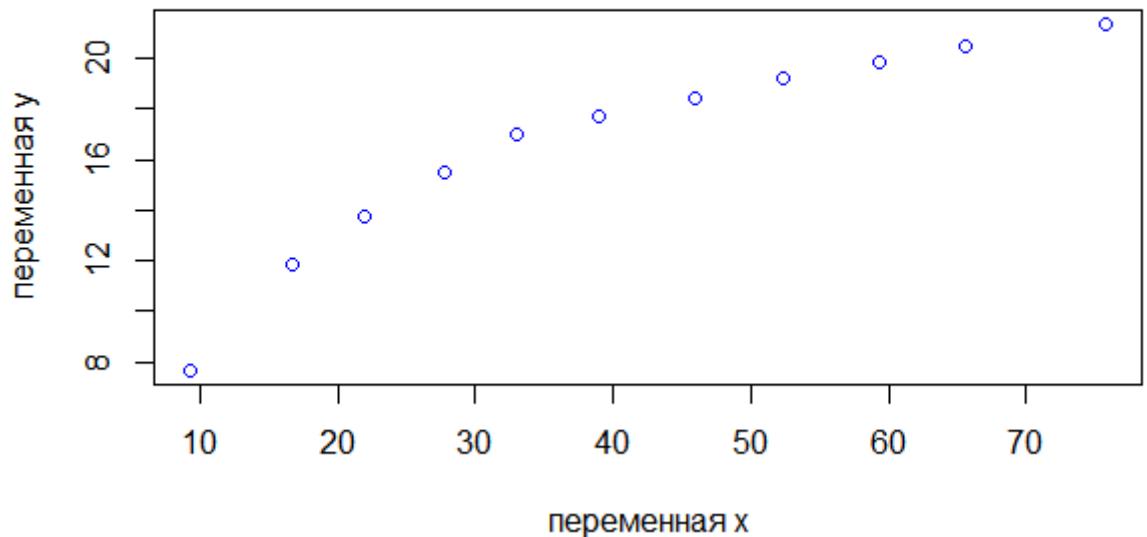
Residuals:
 Min 1Q Median 3Q Max
-1.90072 -0.43528 0.07635 0.67964 1.29509

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.590 1.134 2.283 0.0483 *
z 2.283 0.178 12.823 4.37e-07 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9948 on 9 degrees of freedom
Multiple R-squared: 0.9481, Adjusted R-squared: 0.9423
F-statistic: 164.4 on 1 and 9 DF, p-value: 4.367e-07

> print(data)
 x y
1 9.35 7.67
2 16.72 11.83
3 21.87 13.76
4 27.74 15.48
5 32.95 16.99
6 38.97 17.74
7 45.87 18.41
8 52.32 19.18
9 59.32 19.88
10 65.57 20.50
11 75.75 21.34
```



Уравнение регрессии имеет вид:

$$y = 2.2832 \cdot z + 2.5896.$$

2. Перейдем к начальной замене и получим уравнение нелинейной регрессионной модели:

$$y = 2.2832 \cdot \sqrt{x} + 2.5896$$

II способ. Для нахождение уравнения регрессии используем функцию `nls()`:

```
#Принять предполагаемые значения и вписать в модель.
model <- nls(y ~ b1 * sqrt(x) + b0, start = list(b0 = 3 ,b1 = 2))
summary(model)

Formula: y ~ b1 * sqrt(x) + b0

Parameters:
 Estimate Std. Error t value Pr(>|t|)
b0 2.590 1.134 2.283 0.0483 *
b1 2.283 0.178 12.823 4.37e-07 ***

signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9948 on 9 degrees of freedom

Number of iterations to convergence: 1
Achieved convergence tolerance: 1.992e-09
```

Соответственно, уравнение нелинейной регрессии примет вид:

$$y = 2.2832 \cdot \sqrt{x} + 2.5896$$

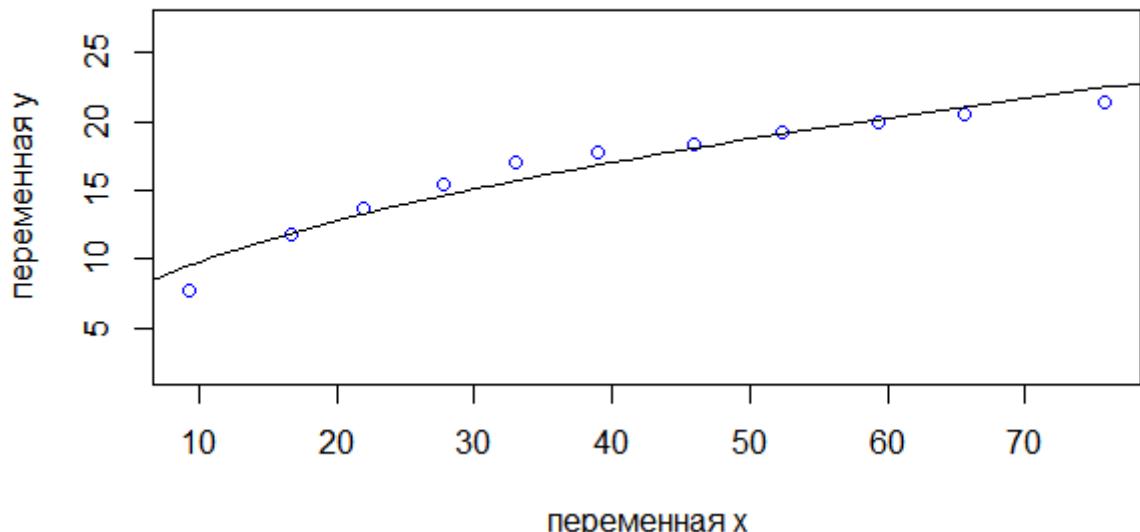
3. Создадим новый вектор со значениями от 1 до 100 для построения графика экспериментальных значений и кривой регрессии:

```

x <- seq(1,100)
plot(data,type="p", col="blue", xlab="переменная x", ylab="переменная у",
 asp=1, main="график нелинейной регрессии")
lines(2.283 * sqrt(x) + 2.590)

```

### график нелинейной регрессии



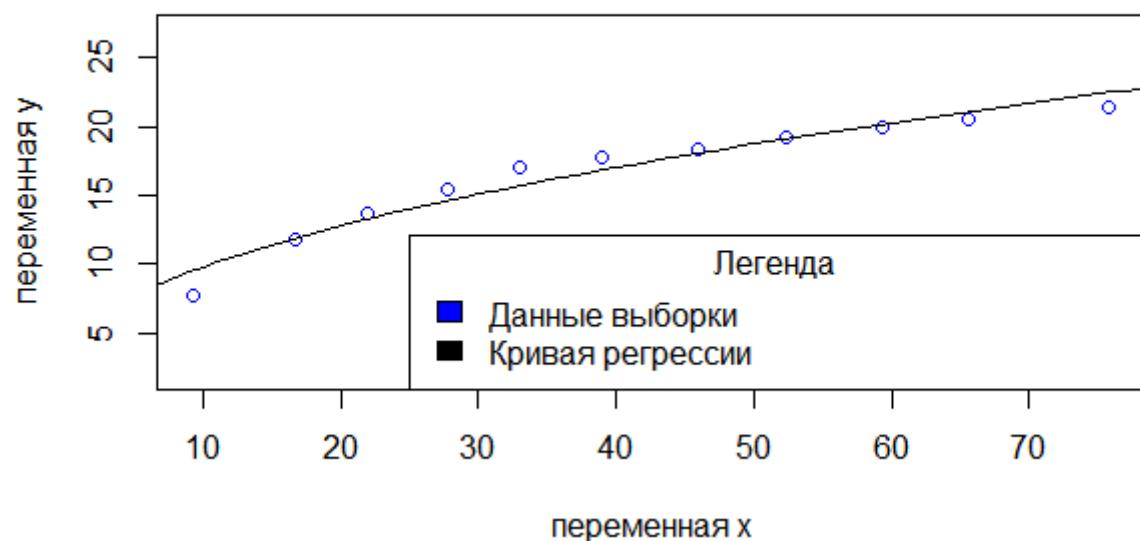
Добавим легенду к графику:

```

plot(data,type="p", col="blue", xlab="переменная x", ylab="переменная у",
 asp=1, main="график нелинейной регрессии")
lines(2.283 * sqrt(x) + 2.590)
main = "Легенда"
location = "bottomright"
labels = c("данные выборки", "Кривая регрессии")
colors = c("blue", "black")
legend(location, labels, title = main, fill=colors)

```

### график нелинейной регрессии



4. Загружаем пакет **car**:

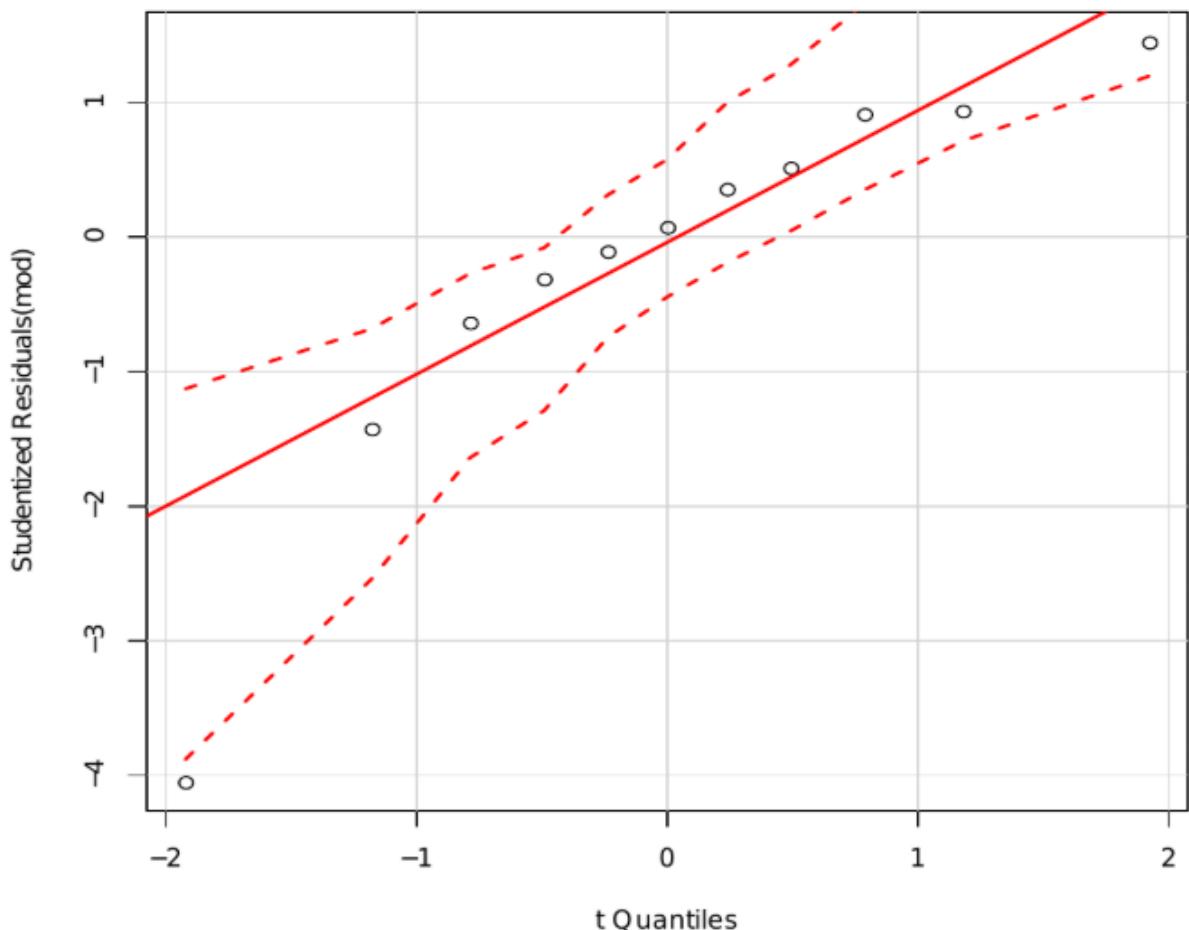
```
> install.packages("car")
Installing package into 'C:/Users/polos/OneDrive/документы/R/win-library/4.0'
(as 'lib' is unspecified)
 пробую URL 'https://cran.rstudio.com/bin/windows/contrib/4.0/car_3.0-8.zip'
Content type 'application/zip' length 1556946 bytes (1.5 MB)
downloaded 1.5 MB

package 'car' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
C:\Users\polos\AppData\Local\Temp\Rtmpi8BjkP\downloaded_packages
```

4.1 Так как параметр функции qqPlot() должен быть объект lm, то используем первый вариант нахождения регрессии:

```
library("car")
z <- sqrt(x)
mod <- lm(y ~ z)
qqPlot(mod)
```



Модель – нормальная.

4.2 Независимость.

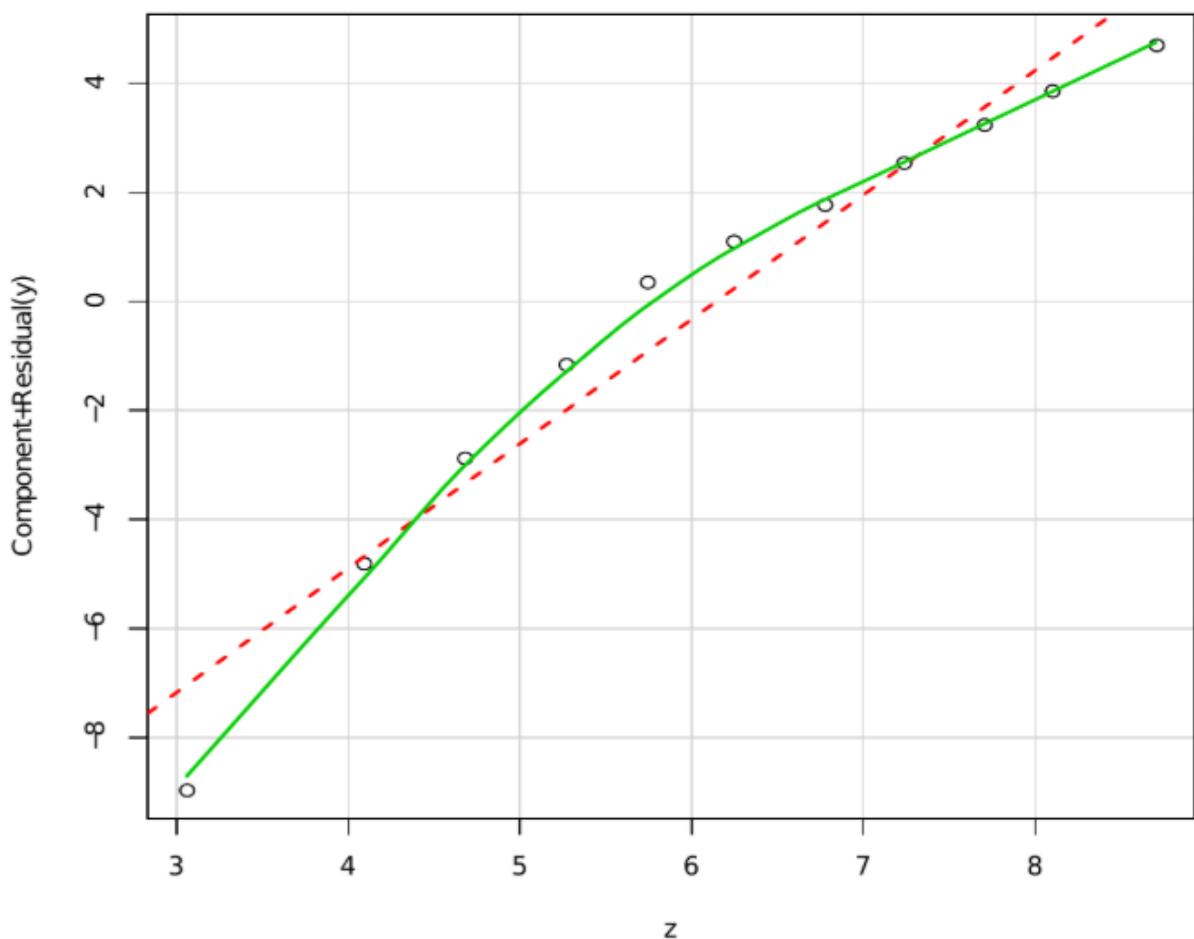
```
durbinWatsonTest(mod)
```

```
lag Autocorrelation D-W Statistic p-value
 1 0.4476818 0.5581095 0
Alternative hypothesis: rho != 0
```

$p-value = 0$  следовательно присутствует положительная автокорреляция.

#### 4.3 Линейность.

```
crPlots(mod)
```



Вывод – модель нелинейная.

#### 4.4 Гомоскедастичность.

```
ncvTest(mod)
spreadLevelPlot(mod)
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 1.847507 Df = 1 p = 0.1740737

Suggested power transformation: 2.462306
```

Результат теста незначим ( $p = 0.174$ ) , что свидетельствует о выполнении условия однородности дисперсии.

