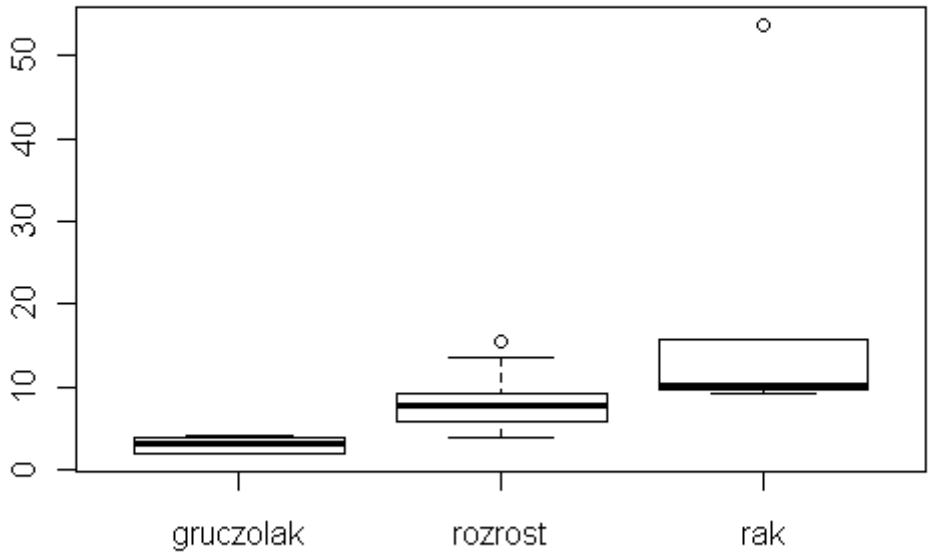
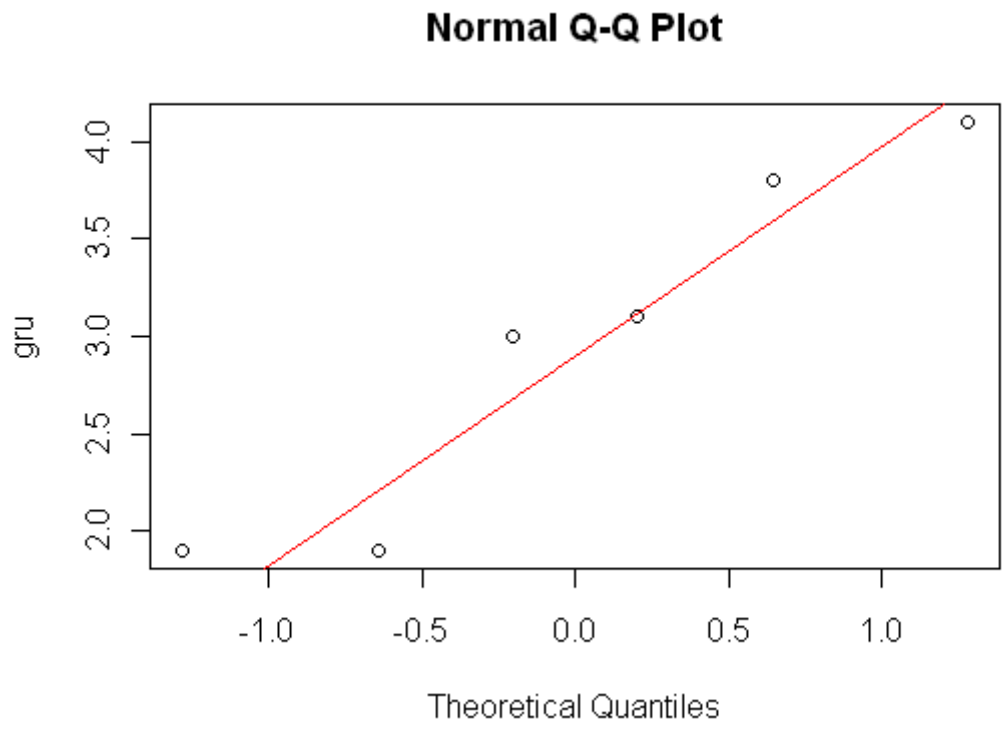


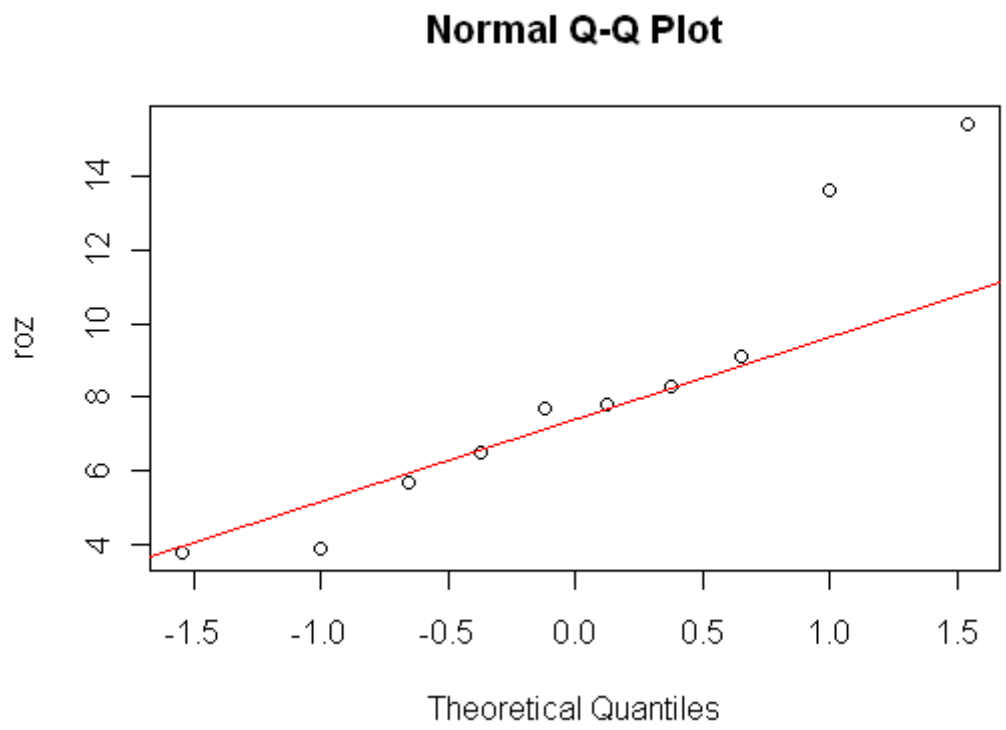
*

```
> apply(kortyzol,2,FUN=mean,na.rm=T)
gruczolak   rozrost       rak
 2.966667  8.180000 19.720000
> apply(kortyzol,2,FUN=sd,na.rm=T)
 gruczolak    rozrost        rak
 0.9244818  3.7891072 19.2388149

> t.test(gruczolak,rozrost)

        Welch Two Sample t-test

data:  gruczolak and rozrost
t = -4.1499, df = 10.685, p-value = 0.001719
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -7.988307 -2.438360
sample estimates:
mean of x mean of y
 2.966667  8.180000

> t.test(gruczolak,rak)

        Welch Two Sample t-test

data:  gruczolak and rak
t = -1.9453, df = 4.015, p-value = 0.1233
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -40.628334   7.121667
sample estimates:
mean of x mean of y
 2.966667 19.720000

> t.test(rozrost,rak)

        Welch Two Sample t-test
```

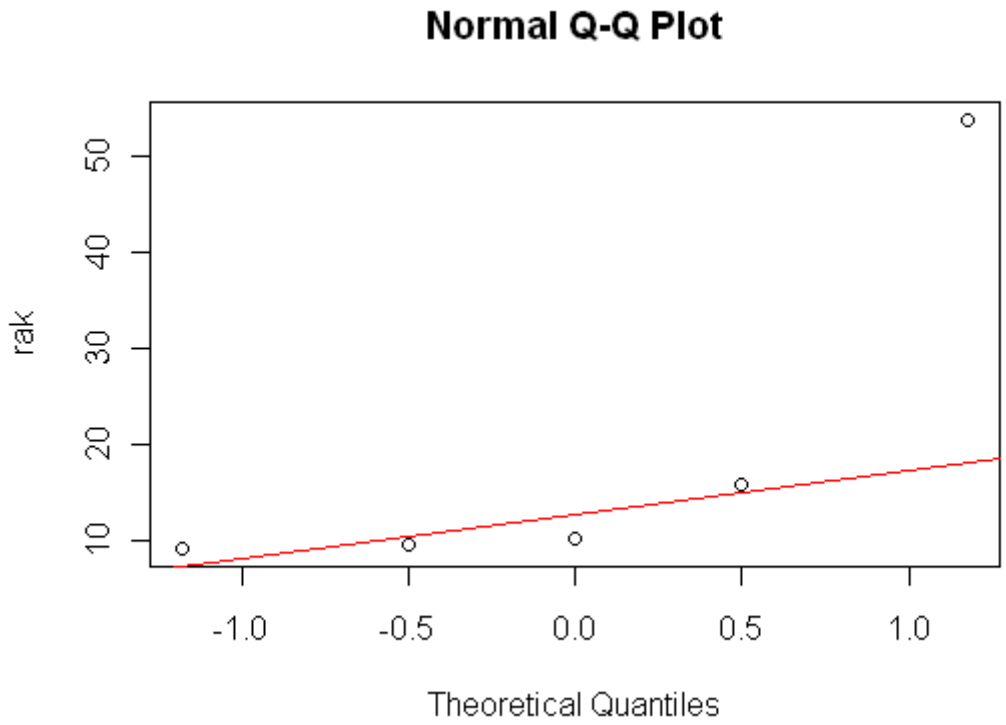
```
data:  rozrost and rak
t = -1.3284, df = 4.156, p-value = 0.2523
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -35.30587  12.22587
sample estimates:
mean of x mean of y
     8.18     19.72
> kort <-
data.frame(0.1*c(31,30,19,38,41,19,83,38,39,78,91,154,77,65,57,136,
+
102,92,96,538,158),c(rep(gru,6),rep(roz,10),rep(rak,5)))

> colnames(kort) <- c("kortyzol","choroba")

> qqnorm(kort$kortyzol[choroba=="gru"],ylab="gru")
> qqline(kort$kortyzol[choroba=="gru"],col="red")
```




```
> qqnorm(kort$kortyzol[choroba=="roz"],ylab="roz")
> qqline(kort$kortyzol[choroba=="roz"],col="red")
```

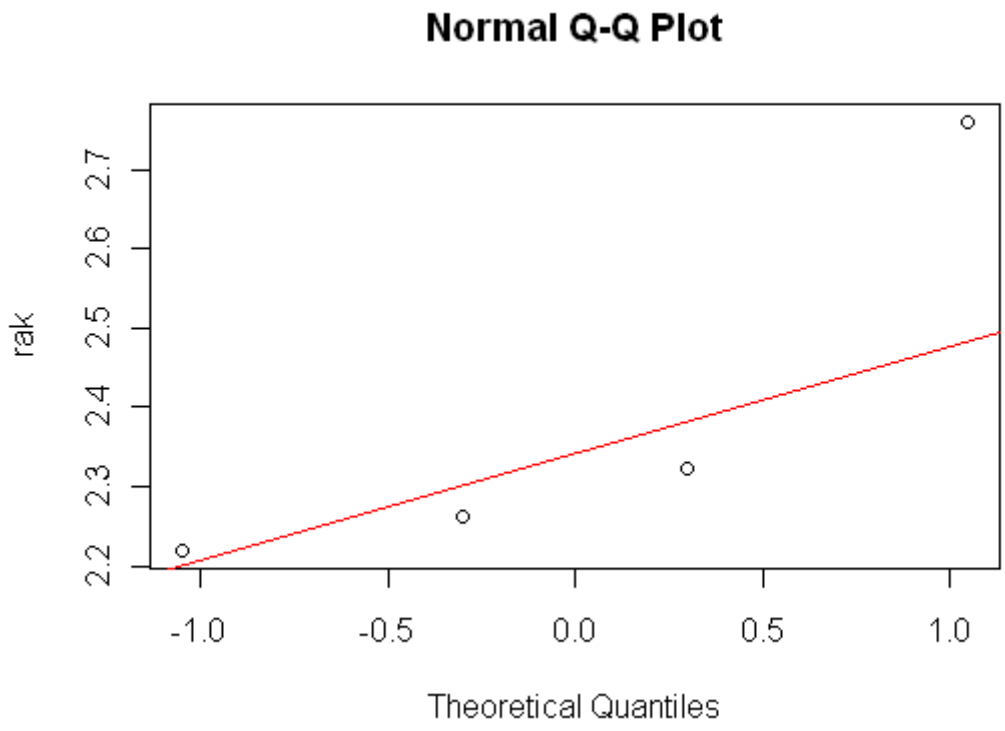



```
> qqnorm(kort$kortyzol[choroba=="rak"],ylab="rak")
> qqline(kort$kortyzol[choroba=="rak"],col="red")
```




```
> (mkor <- log(apply(kortyzol,2,FUN=median,na.rm=T)))
gruczolak   rozrost       rak
 1.115142  2.047693  2.322388
> (iqrkor <- log(apply(kortyzol,2,FUN=IQR,na.rm=T)))
gruczolak   rozrost       rak
0.3715636 1.0986123 1.8245493
```

```
> summary(lm(iqrkor~mkor))

Call:
lm(formula = iqrkor ~ mkor)

Residuals:
gruczolak   rozrost       rak
  0.05457  -0.23981   0.18525

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.9044     0.6537  -1.383    0.398
mkor          1.0953     0.3441   3.183    0.194

Residual standard error: 0.3079 on 1 degrees of freedom
Multiple R-squared: 0.9102,    Adjusted R-squared: 0.8204
F-statistic: 10.13 on 1 and 1 DF,  p-value: 0.1938

> apply(log(kortyzol),2,FUN=mean,na.rm=T)
gruczolak   rozrost       rak
 1.043285  2.007256  2.709728
> apply(log(kortyzol),2,FUN=sd,na.rm=T)
gruczolak   rozrost       rak
0.3326928 0.4602950 0.7451282
```



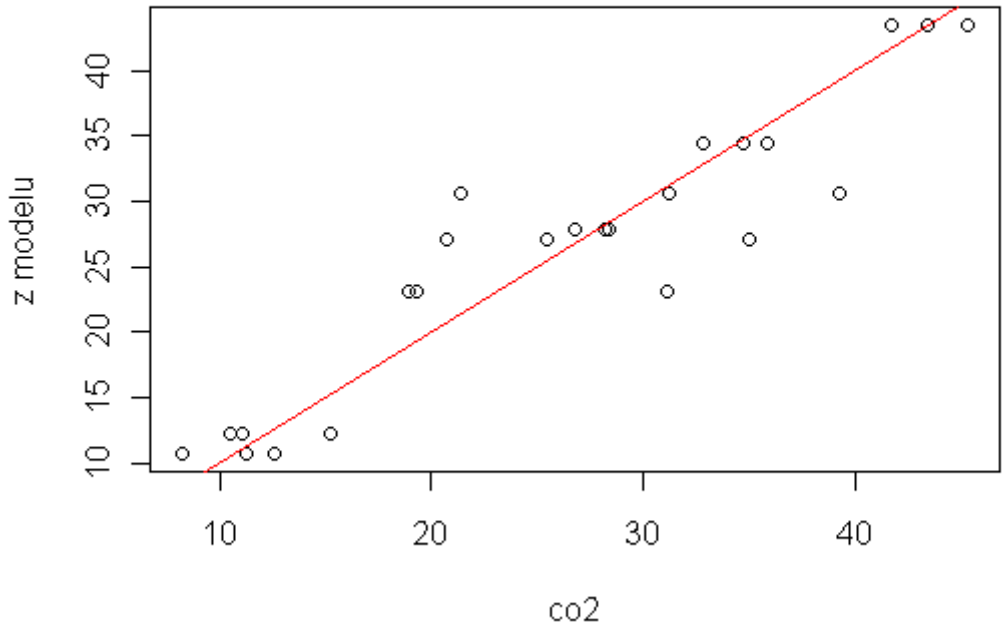

```
> kort.log <- transform(kort,logkrtzl=log(krtzl))
> kort.log.m <-lm(logkrtzl~choroba,data=kort.log)
> summary(kort.log.m)

Call:
lm(formula = logkrtzl ~ choroba, data = kort.log)

Residuals:
     Min       1Q   Median       3Q      Max
-0.67225 -0.40143  0.04687  0.20102  1.27555

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)   1.0433     0.2082   5.011 9.06e-05 ***
chorobarak    1.6664     0.3088   5.397 3.97e-05 ***
chorobaroz    0.9640     0.2633   3.660  0.00179 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.51 on 18 degrees of freedom
Multiple R-squared: 0.6249,    Adjusted R-squared: 0.5832
F-statistic: 14.99 on 2 and 18 DF,  p-value: 0.0001472

> anova(kort.log.m)
Analysis of Variance Table

Response: logkrtzl
          Df Sum Sq Mean Sq F value    Pr(>F)
choroba    2 7.7971  3.8985  14.991 0.0001472 ***
Residuals 18 4.6811  0.2601
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

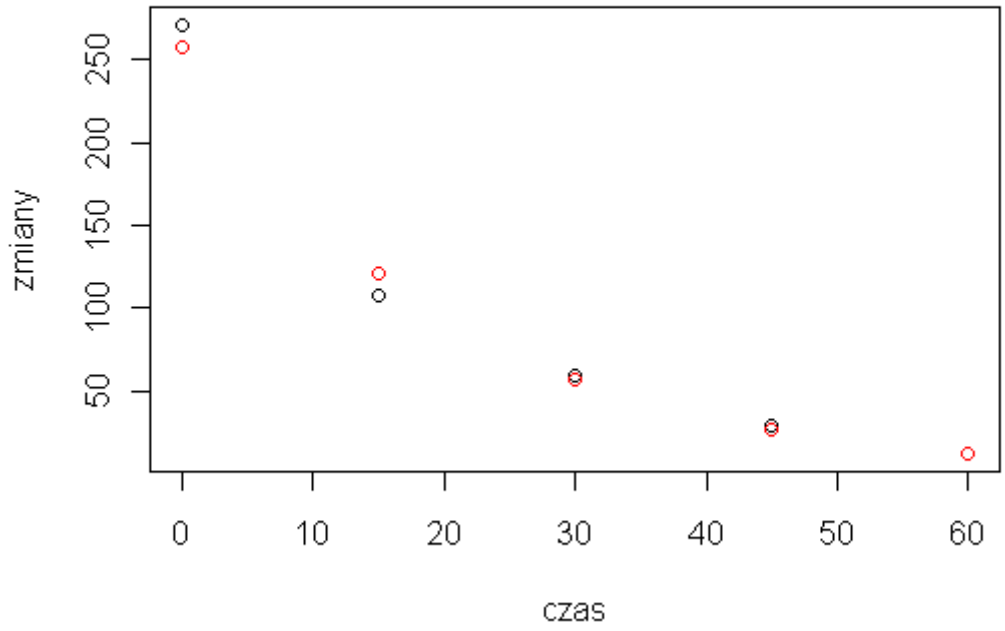
#zad 2

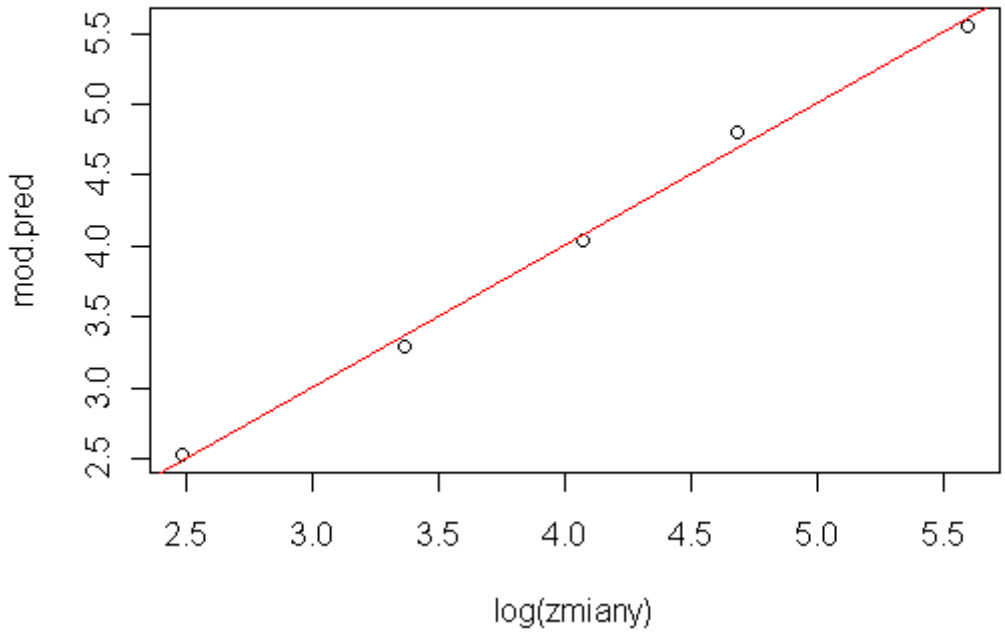
```
> co2 <- c(8.220, 19.296, 25.479, 31.186,
+ 12.594, 31.115, 34.951, 39.237,
+ 11.301, 18.891, 20.688, 21.403,
+ 15.255, 28.200, 32.862, 41.677,
+ 11.069, 26.765, 34.730, 43.448,
+ 10.481, 28.414, 35.830, 45.351)
> dni <- rep(c(24,30,35,38),6)
> poziom <- c(rep("w",12),rep("n",12))
> dni.f <- as.factor(dni)
> poziom.f <- as.factor(poziom)
> dw.mf <- lm(co2~dni.f+poziom.f)
> summary(dw.mf)

Call:
lm(formula = co2 ~ dni.f + poziom.f)

Residuals:
     Min       1Q   Median       3Q      Max
-12.3256  -2.6142  -0.1498   3.0910   8.9899

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   14.808      2.345   6.314 4.64e-06 ***
dni.f30       13.960      2.966   4.706 0.000154 ***
dni.f35       19.270      2.966   6.496 3.18e-06 ***
dni.f38       25.564      2.966   8.618 5.46e-08 ***
poziom.fw     -6.643      2.098  -3.167 0.005074 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.138 on 19 degrees of freedom
Multiple R-squared: 0.827,     Adjusted R-squared: 0.7906
F-statistic: 22.71 on 4 and 19 DF,  p-value: 5.109e-07

> dw.mf2 <- lm(co2~dni.f*poziom.f)
> summary(dw.mf2)

Call:
lm(formula = co2 ~ dni.f * poziom.f)

Residuals:
   Min     1Q Median     3Q    Max
-9.206 -1.794  0.106  1.482  8.628

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)           12.268      2.879   4.261 0.000597 ***
dni.f30               15.525      4.071   3.813 0.001530 **
```

```
dni.f35             22.206      4.071   5.454 5.30e-05 ***
dni.f38             31.224      4.071   7.669 9.55e-07 ***
poziom.fw           -1.563      4.071  -0.384 0.706050
dni.f30:poziom.fw   -3.129      5.758  -0.543 0.594324
dni.f35:poziom.fw   -5.871      5.758  -1.020 0.323039
dni.f38:poziom.fw  -11.320      5.758  -1.966 0.066896 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.986 on 16 degrees of freedom
Multiple R-squared: 0.8628,    Adjusted R-squared: 0.8028
F-statistic: 14.37 on 7 and 16 DF,  p-value: 8.059e-06

> dw.mf3 <- lm(co2~dni.f+poziom.f:dni.f)
> summary(dw.mf3)

Call:
lm(formula = co2 ~ dni.f + poziom.f:dni.f)

Residuals:
   Min     1Q Median     3Q    Max
-9.206 -1.794  0.106  1.482  8.628

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)         12.268      2.879   4.261 0.000597 ***
dni.f30             15.525      4.071   3.813 0.001530 **
dni.f35             22.206      4.071   5.454 5.30e-05 ***
dni.f38             31.224      4.071   7.669 9.55e-07 ***
dni.f24:poziom.fw   -1.563      4.071  -0.384 0.706050
dni.f30:poziom.fw   -4.692      4.071  -1.153 0.266043
dni.f35:poziom.fw   -7.435      4.071  -1.826 0.086561 .
dni.f38:poziom.fw  -12.883      4.071  -3.164 0.006012 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.986 on 16 degrees of freedom
Multiple R-squared: 0.8628,    Adjusted R-squared: 0.8028
F-statistic: 14.37 on 7 and 16 DF,  p-value: 8.059e-06

> nd<-data.frame(dni.f=as.factor(35),poziom.f)
> predict(dw.mf3,nd)
       1        2        3        4        5        6        7        8
9
27.03933 27.03933 27.03933 27.03933 27.03933 27.03933 27.03933 27.03933
27.03933
      10       11       12       13       14       15       16       17
18
27.03933 27.03933 27.03933 34.47400 34.47400 34.47400 34.47400 34.47400
34.47400
      19       20       21       22       23       24
34.47400 34.47400 34.47400 34.47400 34.47400 34.47400
> yy <- fitted(dw.mf3)
> plot(co2,yy,xlab="co2",ylab="z modelu")
> abline(0,1,col=2)
```



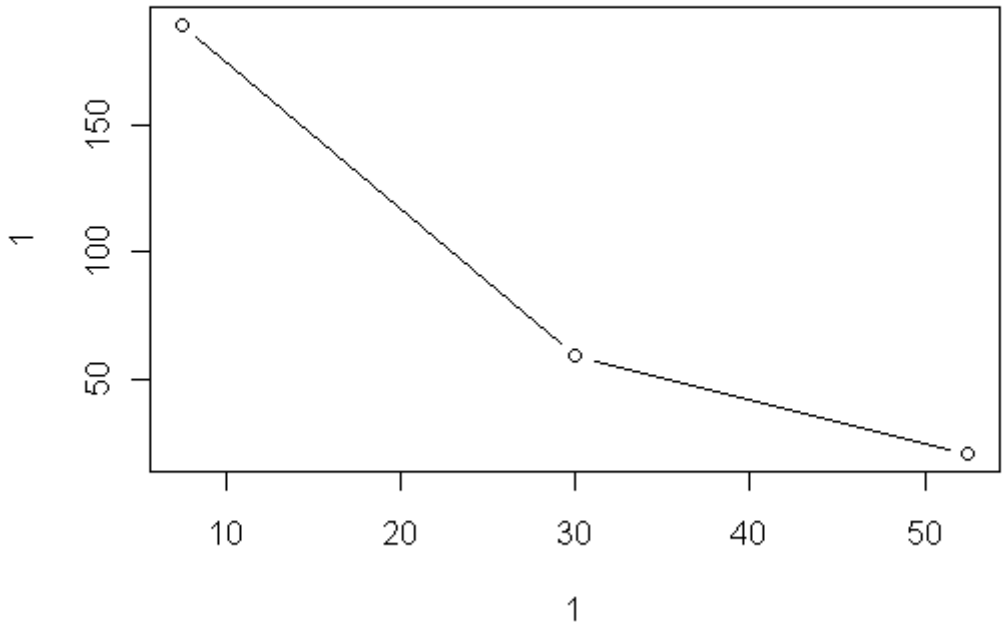

```
# zad 3
> czas <- c(0,15,30,45,60)
> mod.log <- lm(log(zmiany)~czas)
> zmiany <- c(271,108,59,29,12)
> mod.log <- lm(log(zmiany)~czas)
> summary(mod.log)

Call:
lm(formula = log(zmiany) ~ czas)

Residuals:
       1        2        3        4        5
 0.04947 -0.11559  0.03474  0.07942 -0.04804

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.552650   0.071598   77.55 4.73e-06 ***
czas        -0.050328   0.001949  -25.83 0.000127 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.09243 on 3 degrees of freedom
Multiple R-squared: 0.9955,    Adjusted R-squared: 0.994
F-statistic:   667 on 1 and 3 DF,  p-value: 0.0001273

> nczas <- data.frame (czas=czas)
> mod.pred <- predict(mod.log,nczas)
> mod.pred
       1        2        3        4        5
5.552650 4.797724 4.042798 3.287872 2.532946
> plot(log(zmiany),mod.pred)
> abline(0,1,col=2)

> plot(czas,zmiany)
> points(czas,exp(mod.pred),col=2)
```



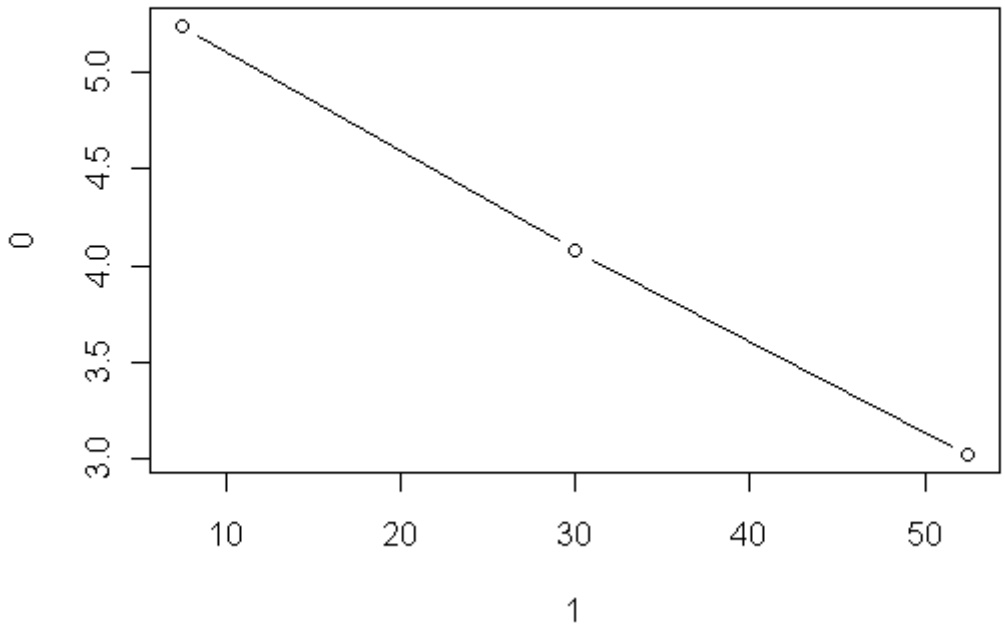


*

```
> strzalkaTukeya(czas,zmiany)

 potęga x = 1  potęga y = 1
 błąd = 0.5443787  b1 = -5.8  b2 = -1.711111

 potęga (x/y),koniec - ESC
```




```
Wybierz x,y,ESC: y
```




```
 potęga x = 1  potęga y = 0
błąd = 0.04934388  b1 = -0.05186007  b2 = -0.04698278

 PRZYBLIŻONE RÓWNANIE REGRESJI: a = 5.61505 b = -0.04942143
```