Regresja_ppodsumowanie

Ewa Bojke

17 grudnia 2019

Dane

Zbiór danych w tym zadaniu pochodzi od Służb Leśnych Departamentu Rolncitwa USA w Kaliforni. Są to dane służące do kalibracji czujnika gęstości. Materiały o dziewięciu różnych, znanych, gęstościach zbadano czujnikiem (każdy przypadek został pomierzony 10-krotnie) i zanotowano wartość sygnału wyjściowego.

Zmienne

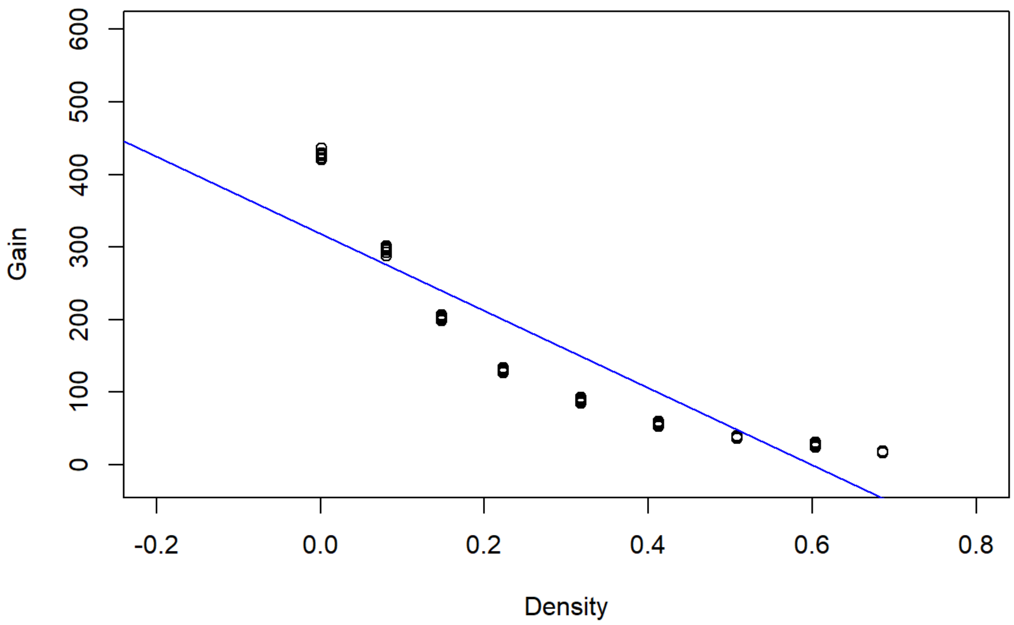
zmienna	opis
`density`	gęstość materiału $(\mathrm{g}/\mathrm{cm}^3)$
`gain`	wartość sygnału wyjściowego

```
gauge <- read.csv("C:/Users/Ewci/Desktop/gauge.csv")
```

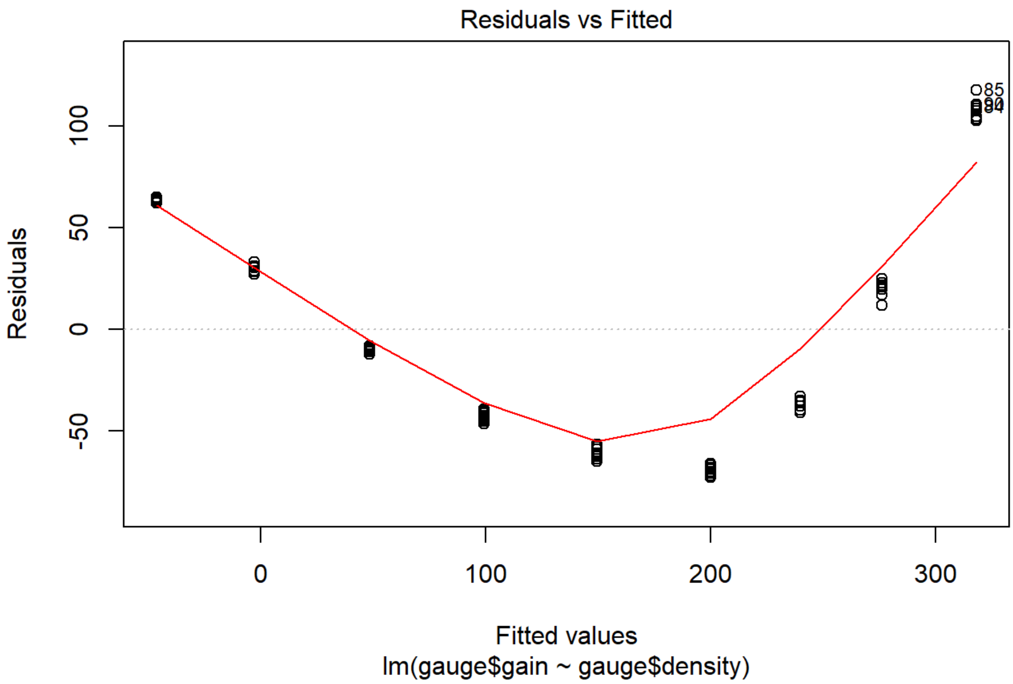
Problemy badawcze

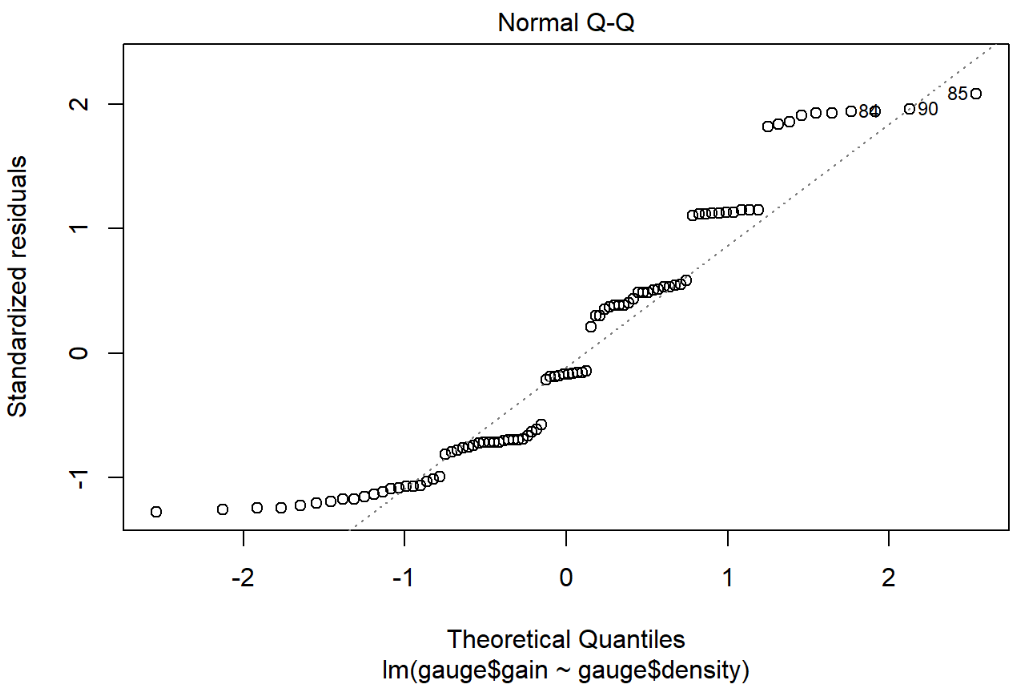
1. Dopasuj model liniowy do naszych danych. Jak wyglądają wykresy diagnostyczne? Odp:

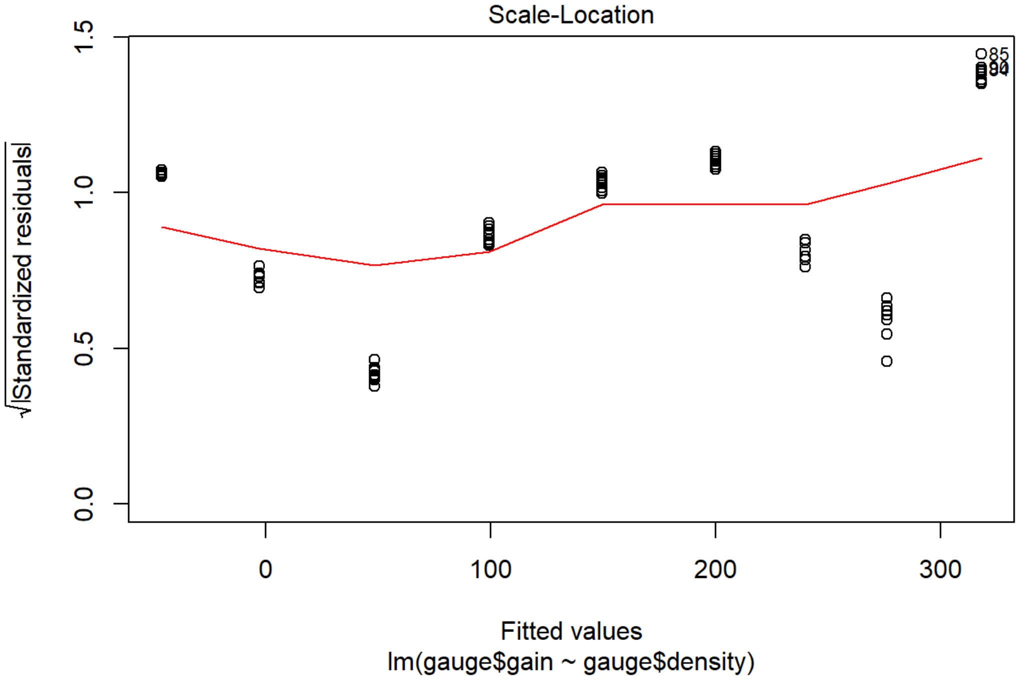
```
model<-lm(gauge$gain~gauge$density)
plot(gauge$density,gauge$gain, xlab="Density", ylab="Gain",xlim=c(-0.2,0.8),ylim=c(-20,600))
abline(model,col="blue")
```

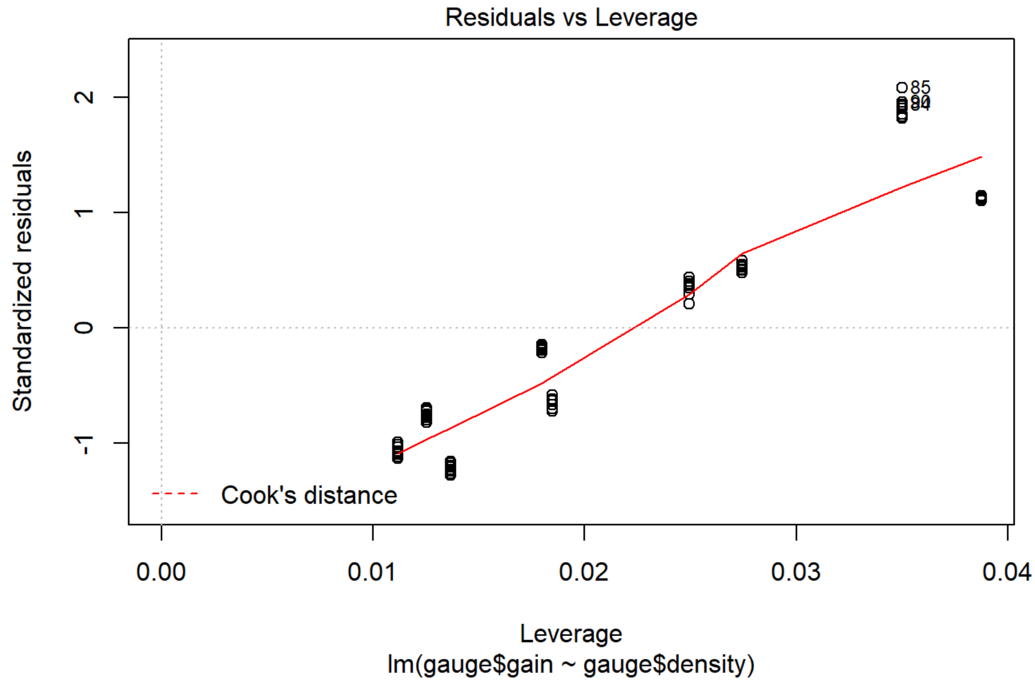



```
plot(model)
```


Residuals vs Fitted
Residuals
100
50
0
-50
0
100
200
300
85
Fitted values
lm(gauge$gain ~ gauge$density)


Normal Q-Q
Standardized residuals
2
1
0
-1
-2
-1
0
1
2
84
90
85
Theoretical Quantiles
lm(gauge$gain ~ gauge$density)





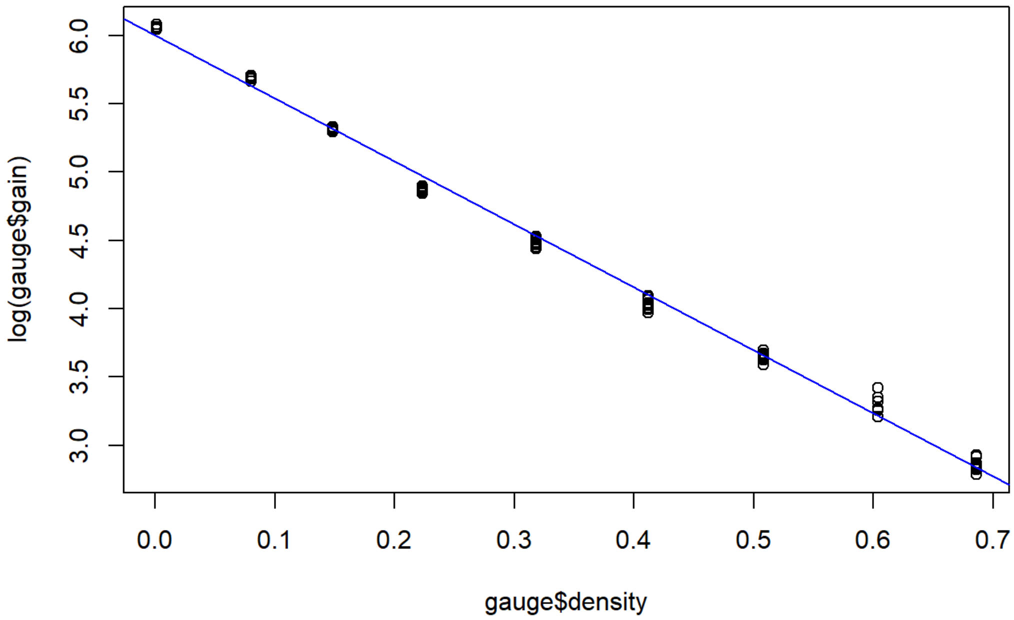
Wnioski i obserwacje: Narysowaliśmy wykres naszych danych z pliku "Gauge.csv" i dopasowaliśmy do nich model liniowy. Można powiedzieć, że model nie dopasował się do końca naszych danych gdyż wykres tych danych przypomina funkcje wykładniczą. Wykresy diagnostyczne potwierdzają wniosek.

---

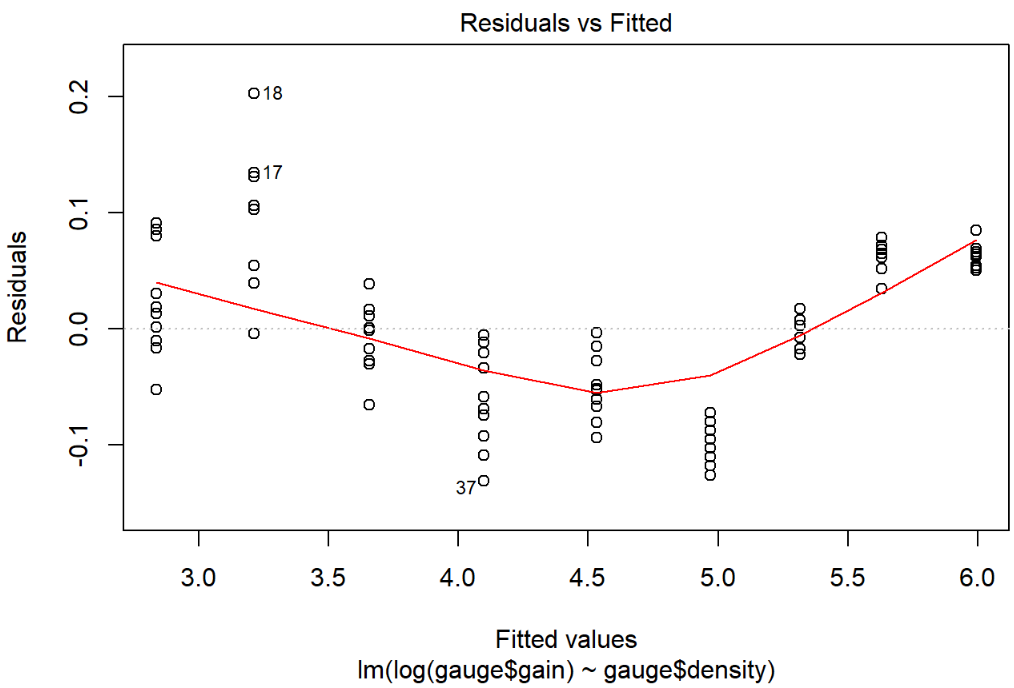
2. Jak wyglądają wykresy diagnostyczne, gdy zmienna zależna zostanie zlogarytmowana?

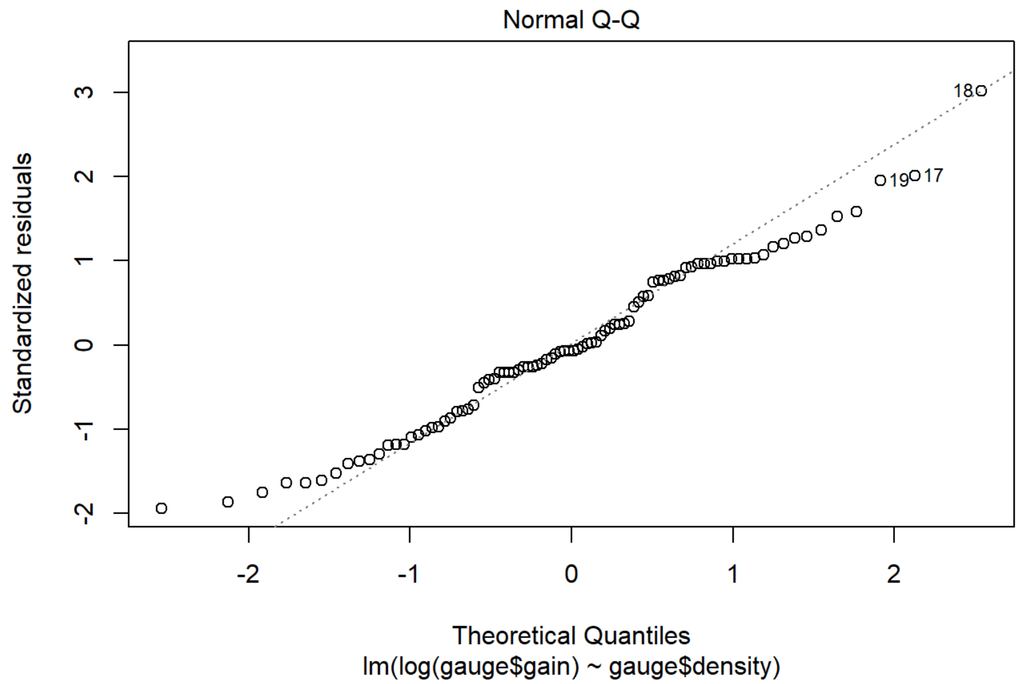
Zmienną zależną jest gain. ZLogarytmujmy ją w procesie modelu i porównajmy powstały model z poprzednim. ***

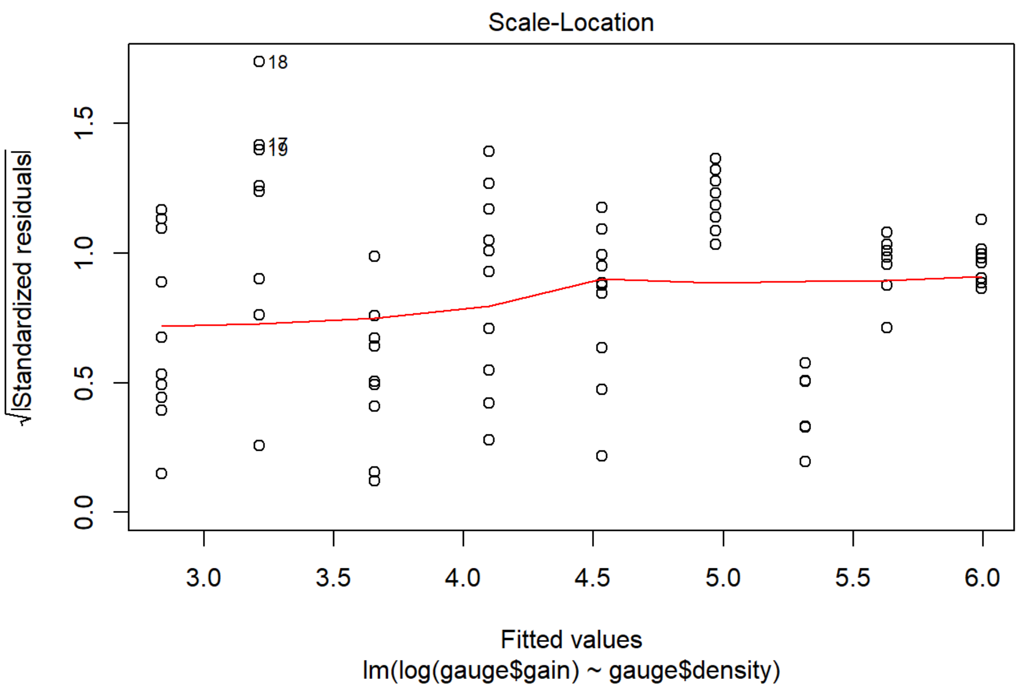
```
model.log <- lm(log(gauge$gain)~gauge$density)
plot(gauge$density,log(gauge$gain))
abline(model.log, col="blue")
```

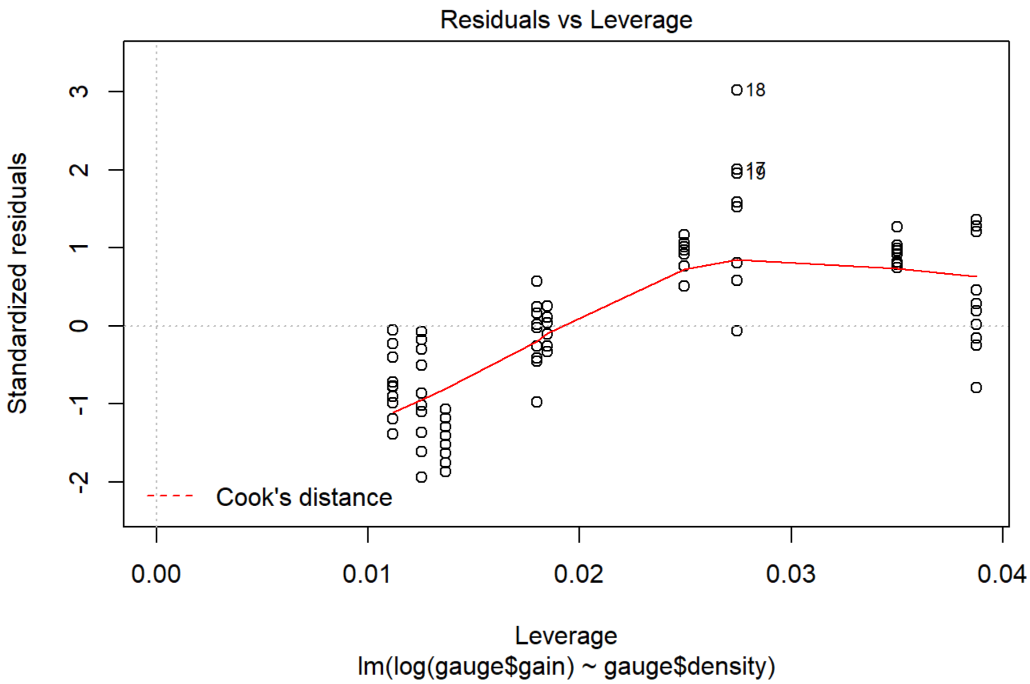



```
plot(model.log) #wykresy diagnostyczne
```




Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(log(gauge$gain) ~ gauge$density)
18
19
17
-2
-1
0
1
2
3


Scale-Location
√|Standardized residuals|
Fitted values
lm(log(gauge$gain) ~ gauge$density)
18
17
19
0.0
0.5
1.0
1.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0



Wnioski i obserwacje: Zmienna zależna jest lepiej dopasowana jak ją zlogarytmujemy. Wykresy diagnostyczne też wyglądaja lepiej.

---

3. Na podstawie modelu logarytmicznego opracuj procedurę kalibracji. Dla wartości sygnału $\mathrm{gain} = 38.6$ i $\mathrm{gain} = 426.7$ znajdź przedziały ufności.

```
model.lin <- lm(log(gain)~density, gauge)
a <- model.lin$coefficients[[2]]
b <- model.lin$coefficients[[1]]

gain.observed <- 38.6
density.est <- a^{-1}*(log(gain.observed) - b)

#fh tells how much upper boundary of prediction interval differs from gain.observed
fh <- function(x) abs( predict(model.lin, newdata = data.frame(density = x), interval = "prediction")[3] - l
og(gain.observed))



#fh tells how much upper boundary of prediction interval differs from gain.observed
fl <- function(x) abs( predict(model.lin, newdata = data.frame(density = x), interval = "prediction")[2] - l
og(gain.observed))

density.low <- optimize(fl, interval = c(0.01, 1))$minimum
density.high <- optimize(fh, interval = c(0.01, 1))$minimum

c(density.est, density.low, density.high)
```

```
## [1] 0.5089113 0.4793798 0.5385366
```

```
gain.observed2 <- 426.7
density.est2 <- a^{-1}*(log(gain.observed2) - b)
fh2 <- function(x) abs( predict(model.lin, newdata = data.frame(density = x), interval = "prediction")[3] -
log(gain.observed2))
fl2 <- function(x) abs( predict(model.lin, newdata = data.frame(density = x), interval = "prediction")[2] -
log(gain.observed2))
density.low2 <- optimize(fl2, interval = c(0.01, 1))$minimum
density.high2 <- optimize(fh2, interval = c(0.01, 1))$minimum
c(density.est2, density.low2, density.high2)
```

```
## [1] -0.01276954  0.01006520  0.01699373
```

Wnioski i obserwacje: Widzimy, że procedura kalibracji daje nam wartość 0.53. Dla wartości sygnału $\mathrm{gain} = 38.6$ otrzymujemy gęstość 0.51 oraz przedział ufności $[0.5, 0.54]$. Dla $\mathrm{gain} = 426.7$ gęstość -0.01 oraz przedział ufności $[0.01, 0.02]$.

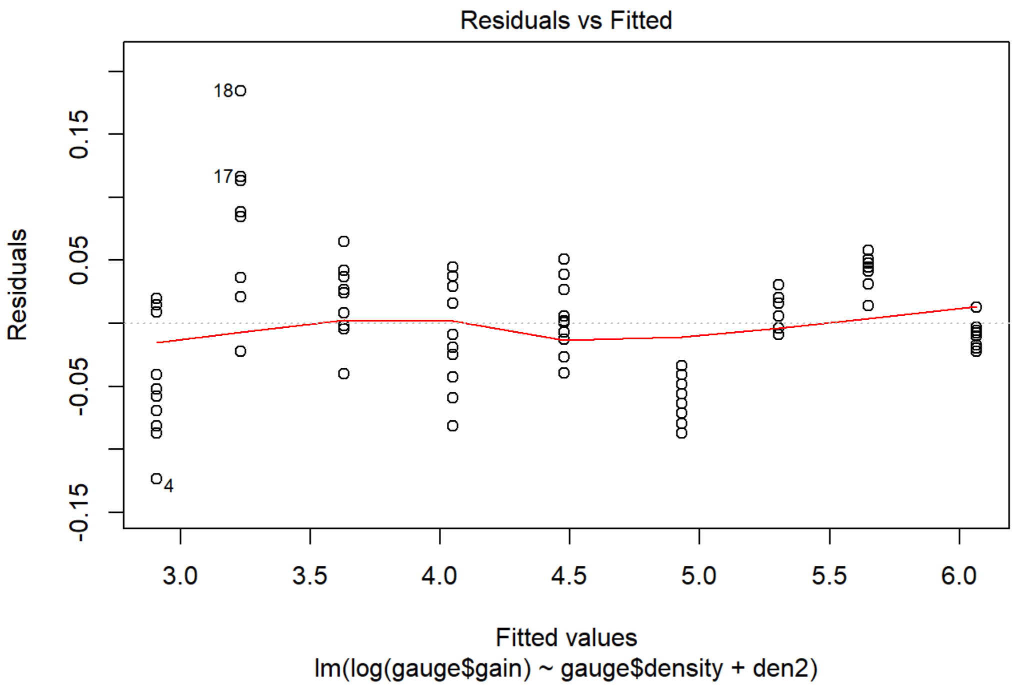
4. Rozważ model $\log(\mathrm{gain}) = a + b \cdot \mathrm{density} + c \cdot \mathrm{density}^2$. Jak wyglądają wykresy diagnostyczne?

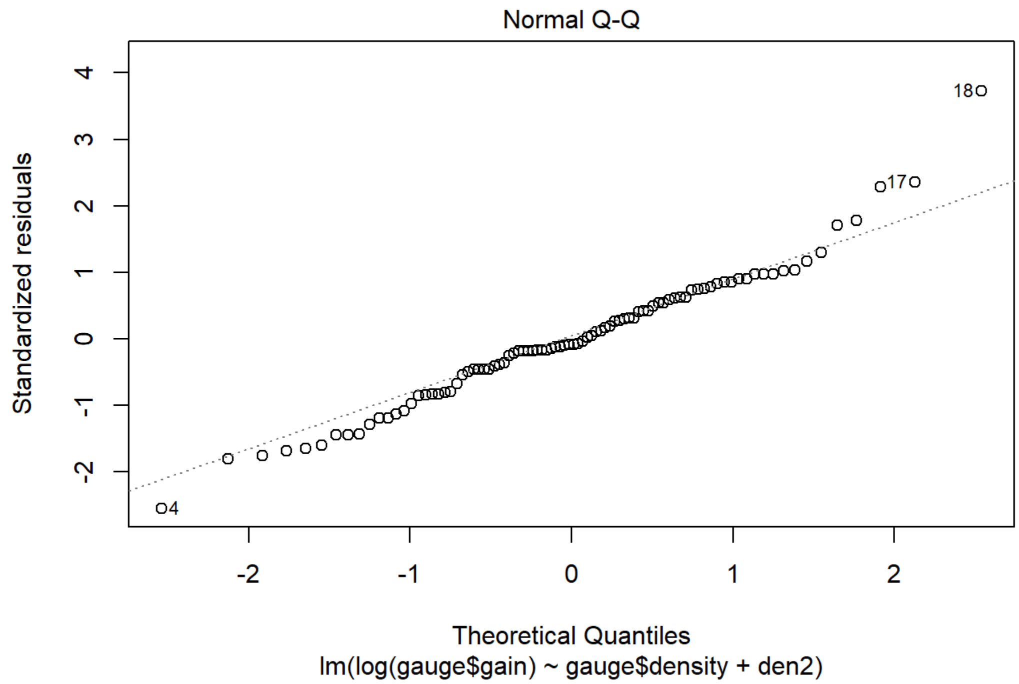
```
den2 <- (gauge$density)^2
model.den2 <- lm(log(gauge$gain)~gauge$density+den2)
summary(model.den2)
```

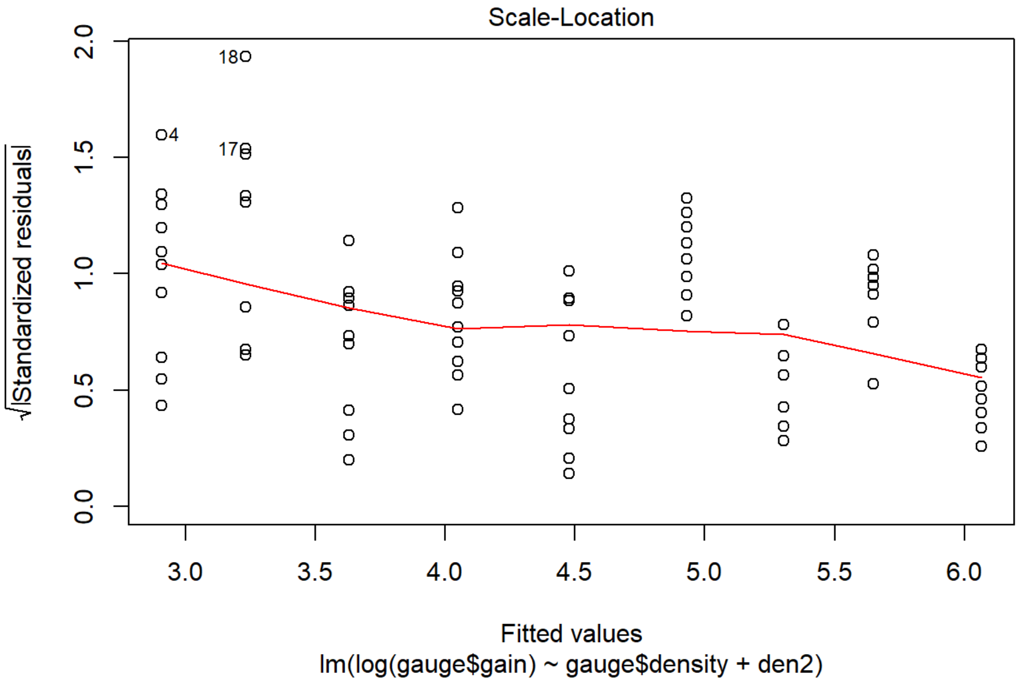
```
##
## Call:
## lm(formula = log(gauge$gain) ~ gauge$density + den2)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.123698 -0.026099 -0.003998  0.030825  0.184439
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.07006    0.01263 480.569  < 2e-16 ***
## gauge$density -5.34675    0.08899 -60.083  < 2e-16 ***
## den2           1.07634    0.12471   8.631 2.56e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05014 on 87 degrees of freedom
## Multiple R-squared:  0.9977, Adjusted R-squared:  0.9977
## F-statistic: 1.927e+04 on 2 and 87 DF,  p-value: < 2.2e-16
```

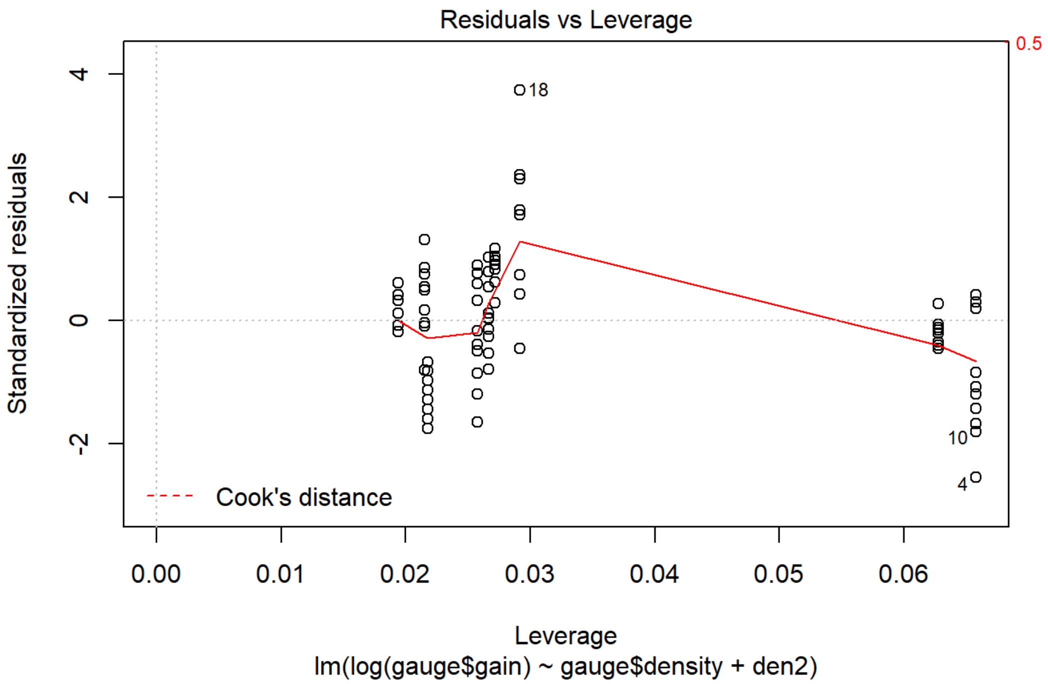
```
coef1<- model.den2$coefficients[[1]]
coef2 <- model.den2$coefficients[[2]]
coef3 <- model.den2$coefficients[[3]]

plot(model.den2)
```




Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(log(gauge$gain) ~ gauge$density + den2)
18
17
4


Scale-Location
√|Standardized residuals|
Fitted values
lm(log(gauge$gain) ~ gauge$density + den2)
18
4
17




```
coef1
```

```
## [1] 6.070059
```

```
coef2
```

```
## [1] -5.346754
```

```
coef3
```

```
## [1] 1.076345
```

Wnioski i obserwacje: Model sie dopasowuje najlepiej ze wszytskich.