Jak naprawić popsutą zabawkę

Transformacje zmiennych w modelach liniowych

Piotr J. Sobczyk

> Data analysis is an artful science! It involves making subjective decisions using very objective tools!
>
> Znalezione w notatkach wykładu Stat 501 z PennState

Na dzisiejszych zajęciach poznamy kilka sposobów na radzenie sobie z danymi, które nie pasują do założeń modelu liniowego. Być może między zmiennymi objaśniającymi, a zmienną objaśniającą jest zależność, ale nie jest ona liniowa. W takim wypadku należy dokonać transformacji danych

Transformacje zmiennych objasniających

Przede wszystkim dodajemy nowe zmienne, np. x^2. Mogą być też bardziej skomplikowane przekształcenia i dyskretyzacja.

Weźmy dane dotyczące uczenia się języka obcego. Od momentu nauki sprawdzamy jaki procent słówek pozostaje w pamięci. Dostajemy następujące dane. Interesuje nas jaka jest zależność, chcielibyśmy potrafić przewidywać procent pamiętanych słów po danym czasie. Notabene na podobnej idei są oparte systemy takie jak supermemo.

```
wordrecall=read.csv("datasets/wordrecall.txt", sep='\t')
str(wordrecall)
```

```
## 'data.frame':    13 obs. of  2 variables:
##  $ time: int  1 5 15 30 60 120 240 480 720 1440 ...
##  $ prop: num  0.84 0.71 0.61 0.56 0.54 0.47 0.45 0.38 0.36 0.26 ...
```

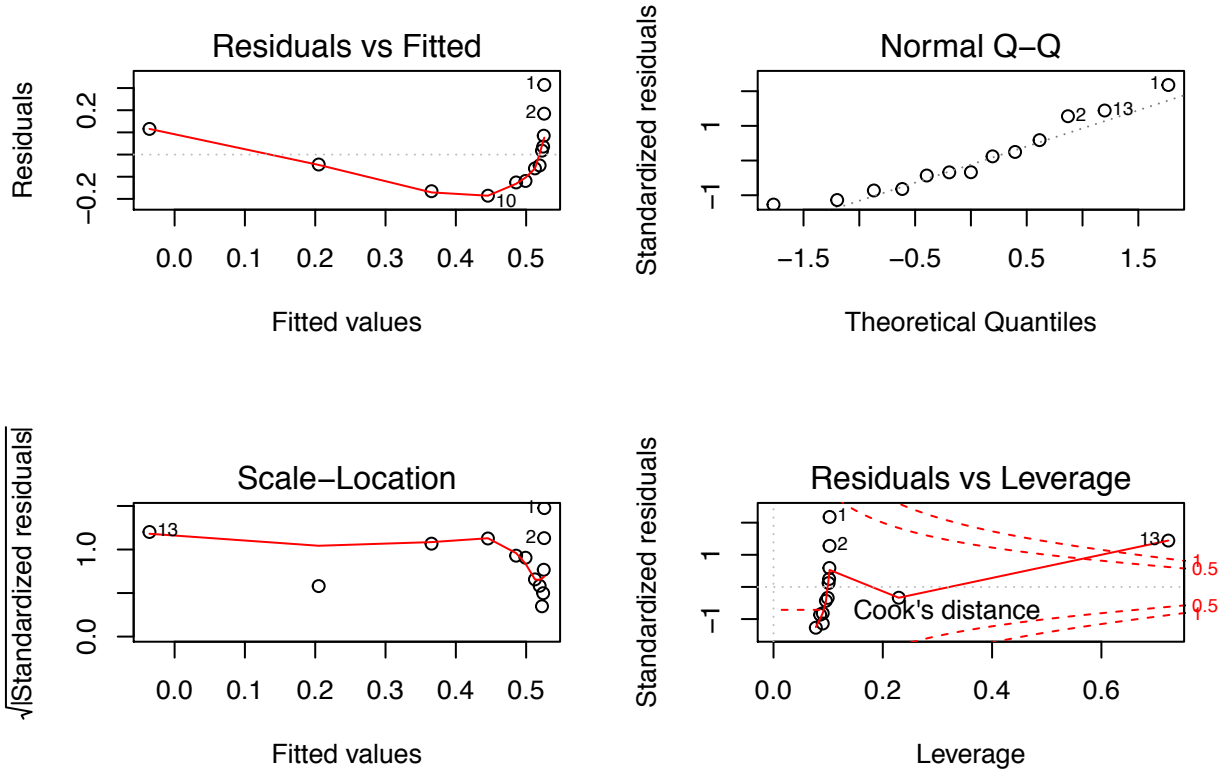
Dopasujemy model liniowy:

```
recall.lm=lm(prop~time, data=wordrecall)
summary(recall.lm)
```

```
##
## Call:
## lm(formula = prop ~ time, data = wordrecall)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18564 -0.11913 -0.04495  0.08496  0.31418
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.259e-01  4.881e-02  10.774 3.49e-07 ***
## time        -5.571e-05  1.457e-05  -3.825  0.00282 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1523 on 11 degrees of freedom
## Multiple R-squared:  0.5709, Adjusted R-squared:  0.5318
## F-statistic: 14.63 on 1 and 11 DF,  p-value: 0.002817
```

Niezłe R^2, istotne współczynniki. A co z diagnostyką?

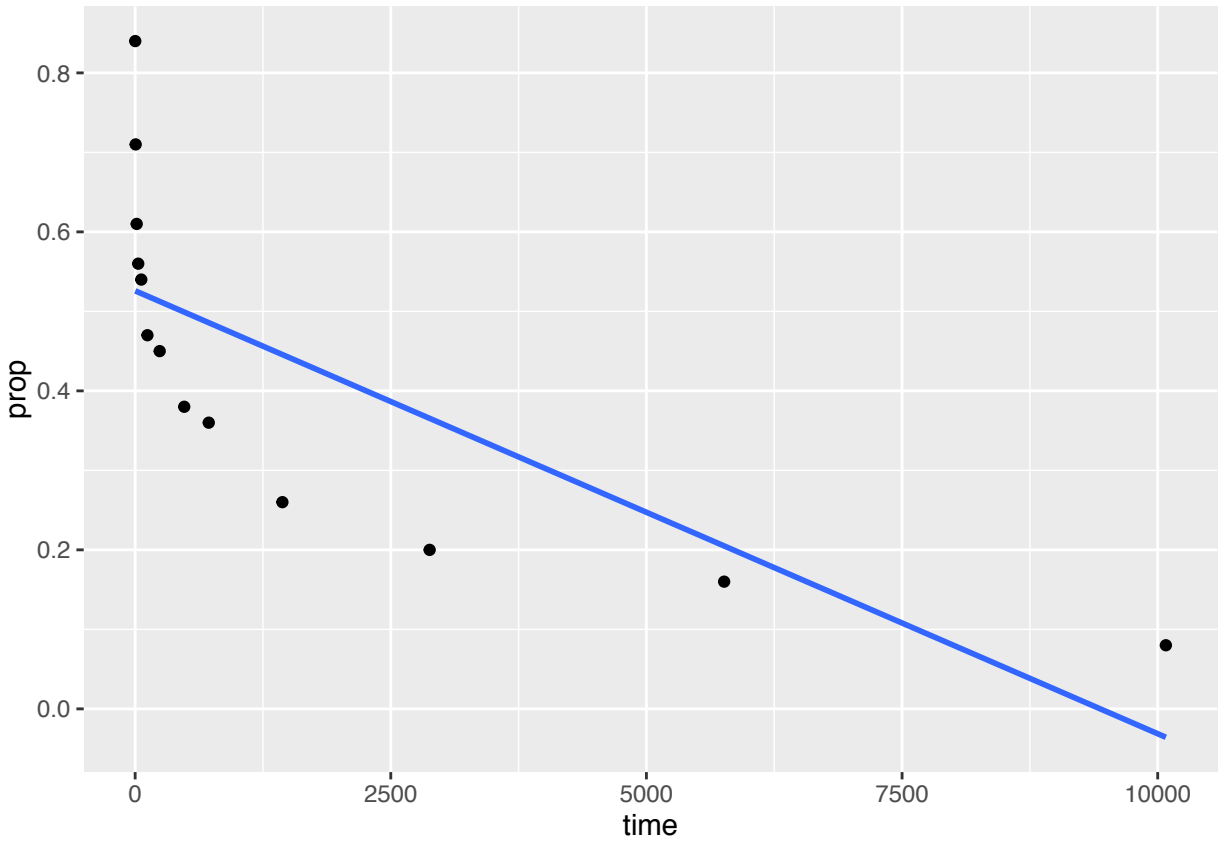
```
par(mfrow=c(2,2))
plot(recall.lm)
```




```
par(mfrow=c(1,1))
```

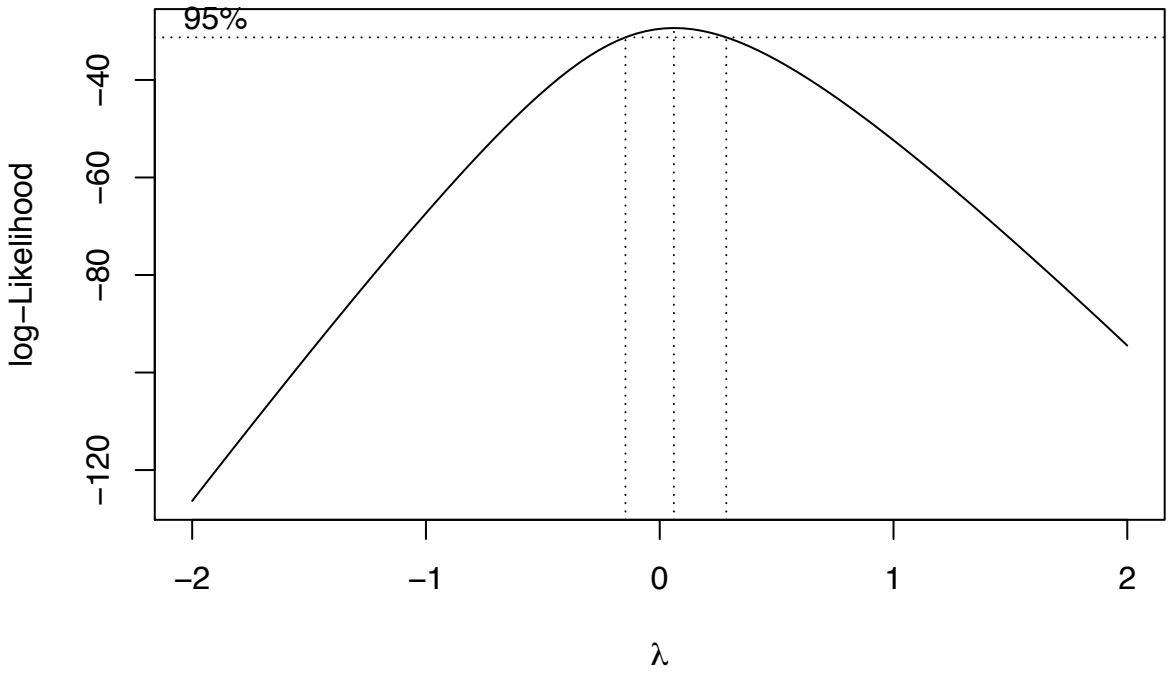
Residua są normalne. Mamy natomiast problem z kiepskim dopasowaniem i niejednorodną wariancją. Mamy jedną obserwację wpływową, ale być może jest ona wynikiem złego dopasowania, więc póki co nie będziemy się nią zajmować.

```
ggplot(wordrecall, aes(x=time, y=prop))+
  geom_point() + stat_smooth(method = "lm", se = F)
```



Przeszktałcamy zmienną x, ponieważ zależność nie wygląda na liniową. Na taką potrzebę wskazuje też wykres Fitted vs Residuals. Jedna z możliwych strategii to symetryzacja zmiennej **time**.

```
MASS::boxcox(time~1, data=wordrecall)
```

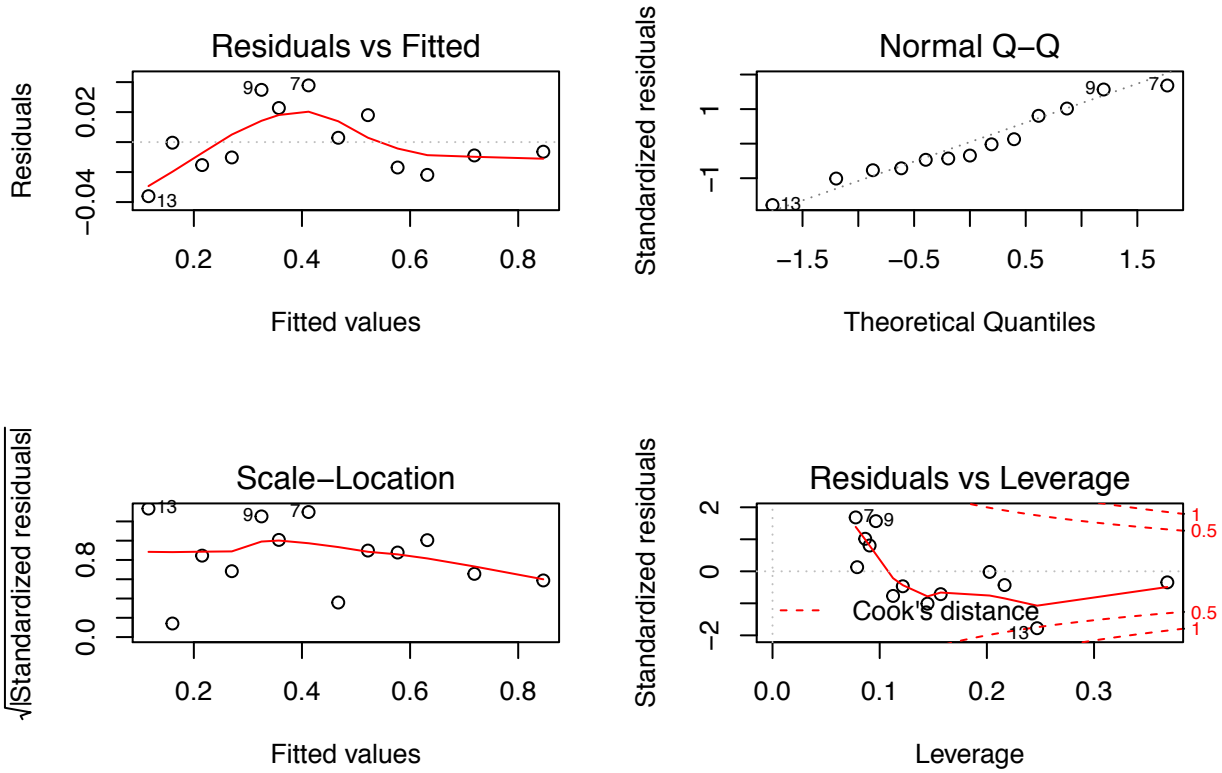



```
recall.lm2=lm(prop~log(time), data=wordrecall)
summary(recall.lm2)
```

```
##
## Call:
## lm(formula = prop ~ log(time), data = wordrecall)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.036077 -0.015330 -0.006415  0.017967  0.037799
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.846415   0.014195   59.63 3.65e-15 ***
## log(time)   -0.079227   0.002416  -32.80 2.53e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02339 on 11 degrees of freedom
## Multiple R-squared:  0.9899, Adjusted R-squared:  0.989
## F-statistic:  1076 on 1 and 11 DF,  p-value: 2.525e-12
```

Wielka różnica! Teraz R^2 jest bardzo wysokie! Sprawdźmy jeszcze czy naszej regresji nie zepsuliśmy.

```
par(mfrow=c(2,2))
plot(recall.lm2)
```

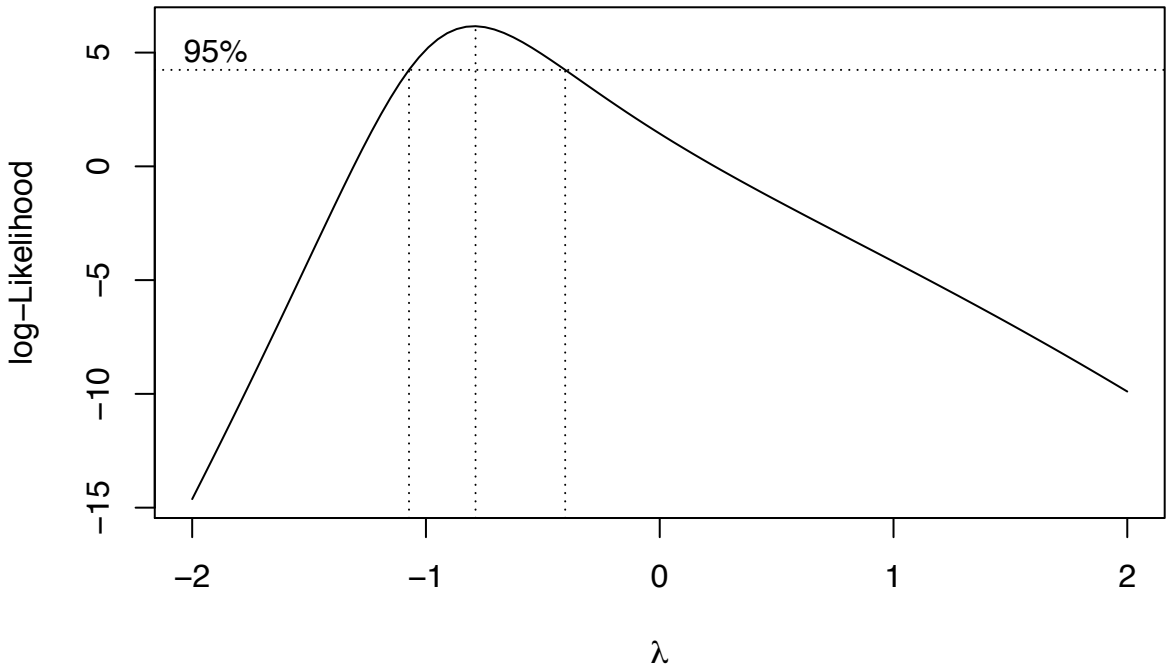



```
par(mfrow=c(1,1))
```

Lekko niepokoić może nas obserwacja 13 z duża miarą Cooka.

Czy moglibyśmy dokonać transformacji zmiennej objaśnianej?

```
MASS::boxcox(prop~time, data=wordrecall) -> boxcox.result
```



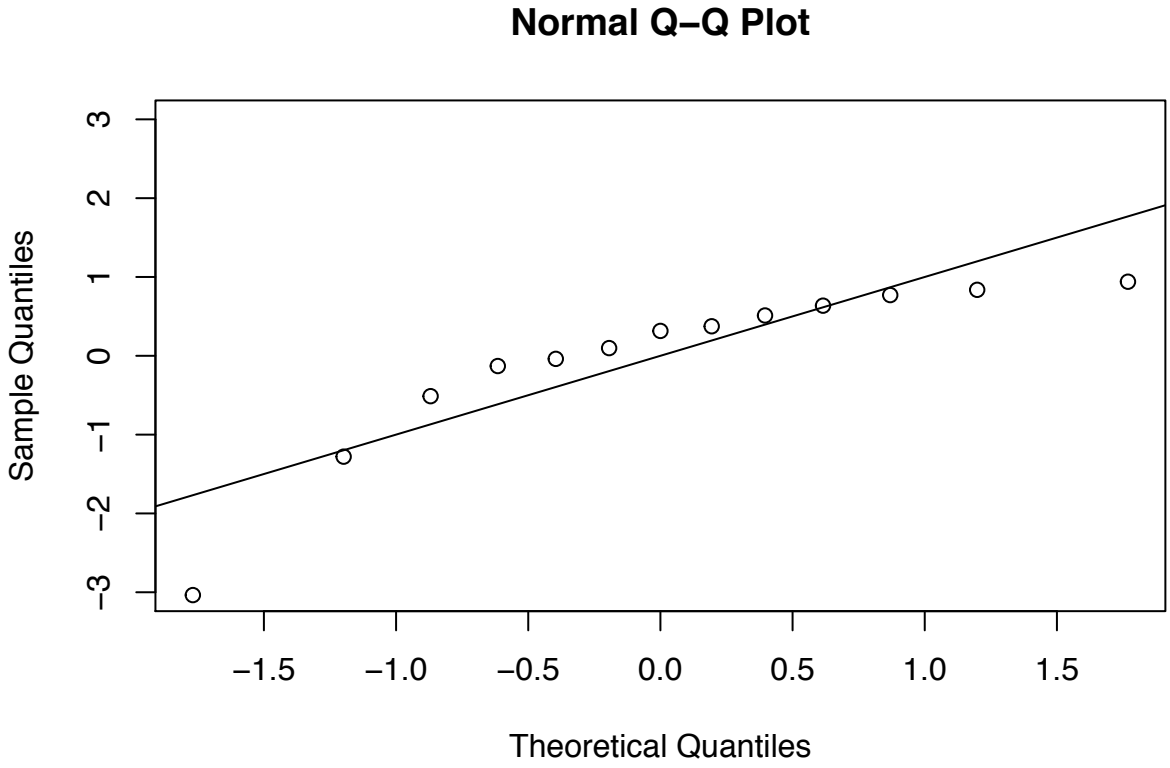

```
lambda=boxcox.result$x[which.max(boxcox.result$y)]
recall.lm3=lm(I((prop^lambda-1)/lambda)~log(time), data=wordrecall)
summary(recall.lm3)
```

```
##
## Call:
## lm(formula = I((prop^lambda - 1)/lambda) ~ log(time), data = wordrecall)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6144 -0.1647  0.3974  0.8180  1.2259
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.2081     0.8326   1.451  0.17467
## log(time)    -0.6085     0.1417  -4.295  0.00127 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.372 on 11 degrees of freedom
## Multiple R-squared:  0.6264, Adjusted R-squared:  0.5925
## F-statistic: 18.44 on 1 and 11 DF,  p-value: 0.001267
```

Jest lepiej jeśli chodzi o R^2, a jak wyglądają wykresy diagnostyczne?

```
qqnorm(y=MASS::stdres(recall.lm3), ylim = c(-3,3))
abline(a=0, b=1, add=TRUE)
```

```
## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "add"
## is not a graphical parameter
```



Residua nie są normalne. Trochę możemy się tego spodziewać, skoro przed transformacją były normalne.

Transformacje zmiennej objaśnianej

Uwaga. Nie robimy tego po to, żeby zmienna objaśniana miała rozkład normalny. To nie jest nasze założenie! Założenie dotyczy rozkładu reszt.

Przekształcenie boxcoxa.

$$y' = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \text{dla } \lambda \neq 0 \\ log(\lambda) & \text{dla } \lambda = 0 \end{cases}$$

Sprawdźmy jak działają przekształcenia na danych dotyczących czasu trwania ciąży w ssaków.

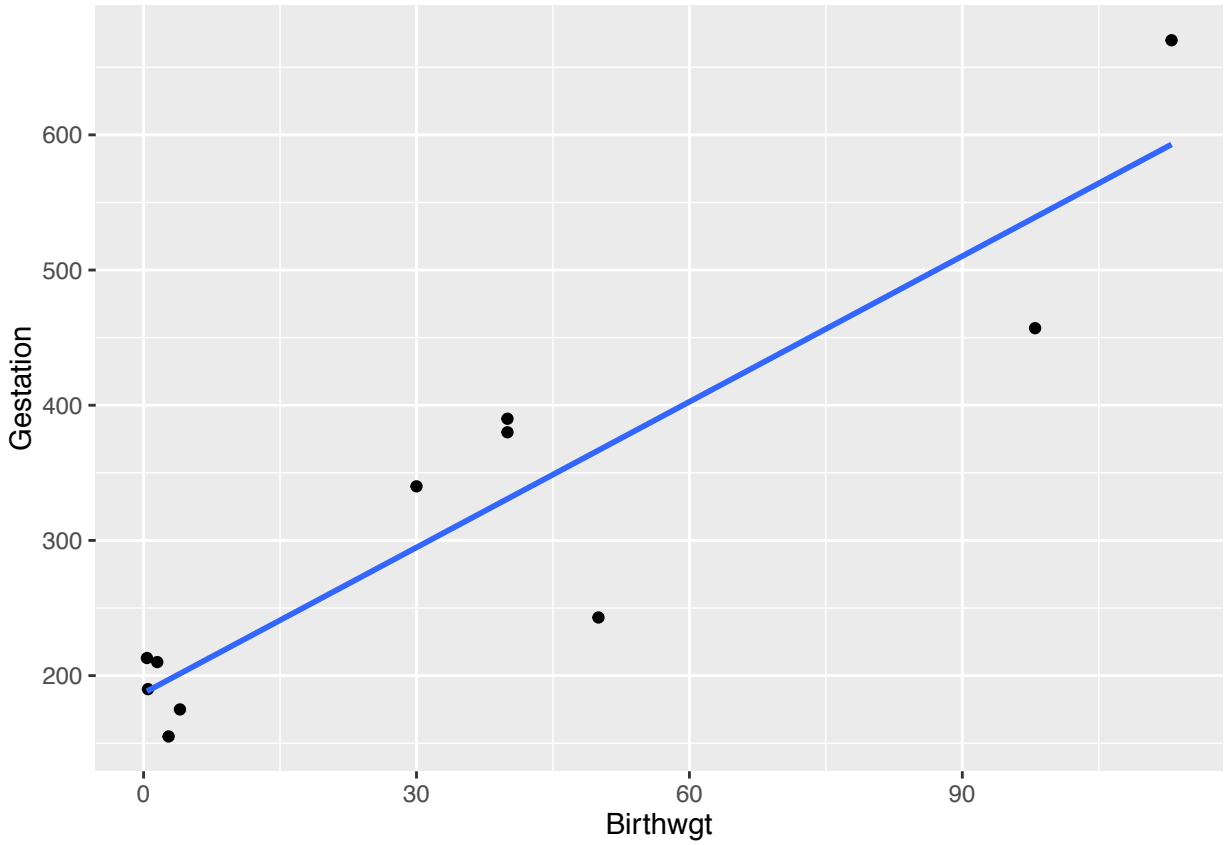
```
# codepages <- setNames(iconvlist(), iconvlist())
# x <- lapply(codepages, function(enc) try(read.csv("datasets/mammgest.txt",
#                     fileEncoding=enc,
#                     nrows=3, header=TRUE, sep="\t"), silent=T))
# unique(do.call(rbind, sapply(x, dim)))
# maybe_ok <- sapply(x, function(x) isTRUE(all.equal(dim(x), c(3,3))))
# codepages[maybe_ok]
# x[maybe_ok]
mamgest=read.csv("datasets/mammgest.txt", sep="\t", header=T, fileEncoding = "UTF-16")
str(mamgest)
```

```
## 'data.frame':    11 obs. of  3 variables:
##  $ Mammal   : Factor w/ 11 levels "Bear","Camel",..: 6 10 3 9 1 7 8 2 11 5 ...
##  $ Birthwgt : num  2.75 4 0.48 1.5 0.37 50 30 40 40 98 ...
##  $ Gestation: int  155 175 190 210 213 243 340 380 390 457 ...
```

```
gestation.lm=lm(Gestation~Birthwgt, data=mamgest)
summary(gestation.lm)
```

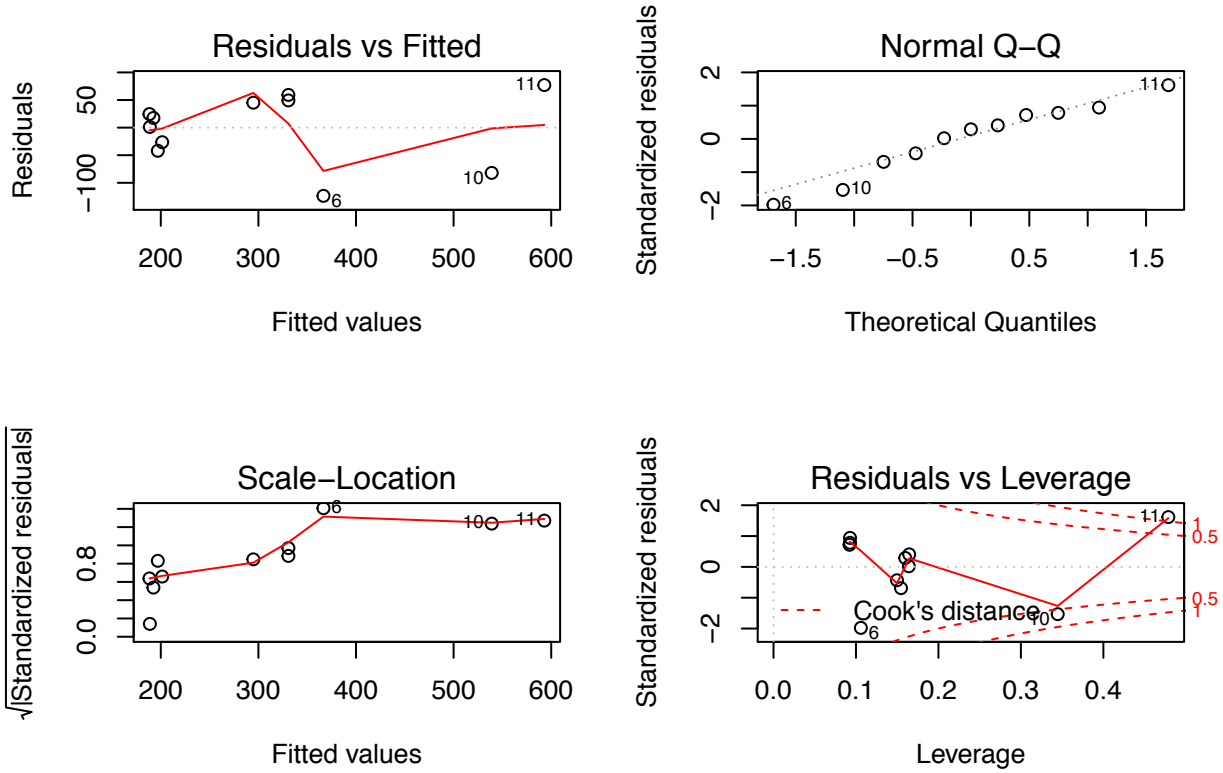
```
##
## Call:
## lm(formula = Gestation ~ Birthwgt, data = mamgest)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -123.65  -34.20   17.53   47.22   77.09
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 187.0837    26.9426   6.944 6.73e-05 ***
## Birthwgt      3.5914     0.5247   6.844 7.52e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 66.09 on 9 degrees of freedom
## Multiple R-squared:  0.8388, Adjusted R-squared:  0.8209
## F-statistic: 46.84 on 1 and 9 DF,  p-value: 7.523e-05
```

```
ggplot(mamgest, aes(x=Birthwgt, y=Gestation))+
  geom_point() + stat_smooth(method = "lm", se = F)
```



Wydaje się, że wszystko jest ok. Współczynniki duże, R^2 bardzo wysokie. Ale „diagonostyka głupcze".

```
par(mfrow=c(2,2))
plot(gestation.lm)
```




```
par(mfrow=c(1,1))
```

Być może jest mały problem z normalnością residuów. Dopasowanie jest w miarę ok, ale wariancja wydaje się być niejednorodna. Zobaczmy co mówią testy.

```
#normalnosc reszt
shapiro.test(MASS::stdres(gestation.lm))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  MASS::stdres(gestation.lm)
## W = 0.95548, p-value = 0.7144
```

```
#jednorodnosc wariancji
bptest(gestation.lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  gestation.lm
## BP = 3.7739, df = 1, p-value = 0.05206
```

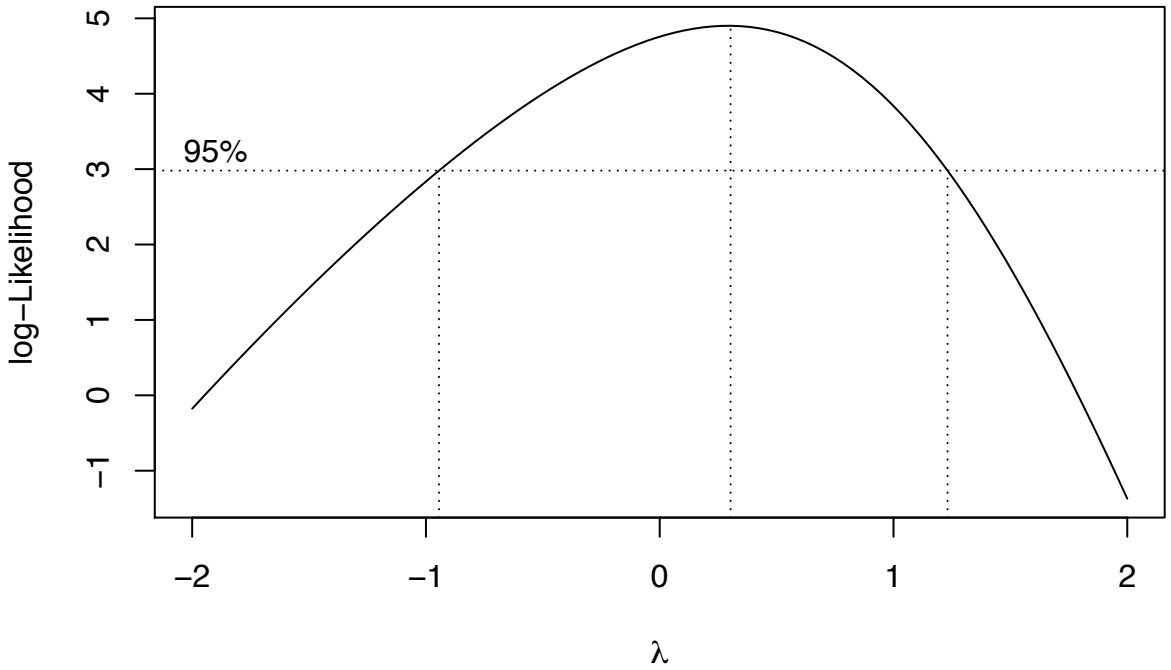
```
gqtest(gestation.lm)
```

```
##
##  Goldfeld-Quandt test
##
## data:  gestation.lm
```

```
## GQ = 21.789, df1 = 4, df2 = 3, p-value = 0.01487
```

Mamy jedną obserwację wpływową, ale być może jest ona wynikiem złego dopasowania, więc póki co nie będziemy się nią zajmować.

```
MASS::boxcox(Gestation~Birthwgt, data=mamgest) -> boxcox.result
```

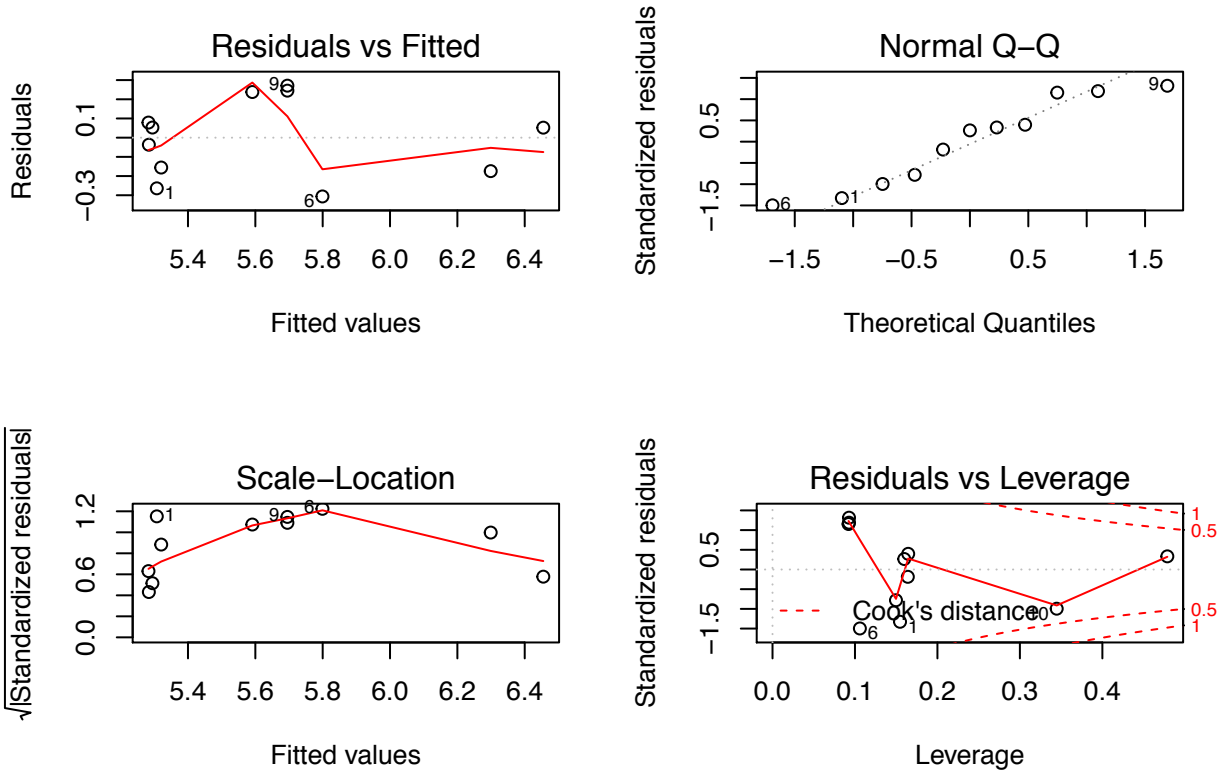



```
lambda=boxcox.result$x[which.max(boxcox.result$y)]
mamgest2=cbind(mamgest, GestationTransf=(mamgest$Gestation^lambda-1)/lambda)
mamgest.lm2=lm(log(Gestation)~Birthwgt, data=mamgest)
summary(mamgest.lm2)
```

```
##
## Call:
## lm(formula = log(Gestation) ~ Birthwgt, data = mamgest)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3063 -0.1650  0.0521  0.1582  0.2709
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.278817   0.088177  59.866  5.1e-13 ***
## Birthwgt    0.010410   0.001717   6.062 0.000188 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2163 on 9 degrees of freedom
## Multiple R-squared:  0.8033, Adjusted R-squared:  0.7814
## F-statistic: 36.75 on 1 and 9 DF,  p-value: 0.0001878
```

Jest lepiej jeśli chodzi o R^2, a jak wyglądają wykresy diagnostyczne?

```
par(mfrow=c(2,2))
plot(mamgest.lm2)
```




```
par(mfrow=c(1,1))
```

R^2 jest niższe, ale wariancja jest teraz jednorodna. Dodatkowo pozbyliśmy się obserwacji odstających. Sprawdźmy jeszcze czy nie zepsuliśmy normalności residuów

```
#normalnosc reszt
shapiro.test(MASS::stdres(mamgest.lm2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  MASS::stdres(mamgest.lm2)
## W = 0.91948, p-value = 0.3143
```

```
#jednorodnosc wariancji
bptest(mamgest.lm2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  mamgest.lm2
## BP = 0.034762, df = 1, p-value = 0.8521
```

```
gqtest(mamgest.lm2)
```

```
##
##  Goldfeld-Quandt test
##
## data:  mamgest.lm2
## GQ = 4.8827, df1 = 4, df2 = 3, p-value = 0.1118
```

Dlaczego transformacje (logarytmiczne) pomagają?

Chcemy uchwycić zależność y od x. W modeli liniowym zakładamy, że jest ona liniowa. Co jeśli jest inaczej?

$$y = \beta_0 e^{\beta_1 x}$$

W takim wypadku nałożenie logarytmu przekształca model multikplikatywny na liniowy

$$log(y) = \beta_0 + \beta_1 x$$

Podobnie dla danych zgodnych z zależnością potęgową mamy

$$y = \beta_0 x^{\beta_1} log(y) = \beta_0 + \beta_1 log(x)$$

Zaś w przypadku zależności:

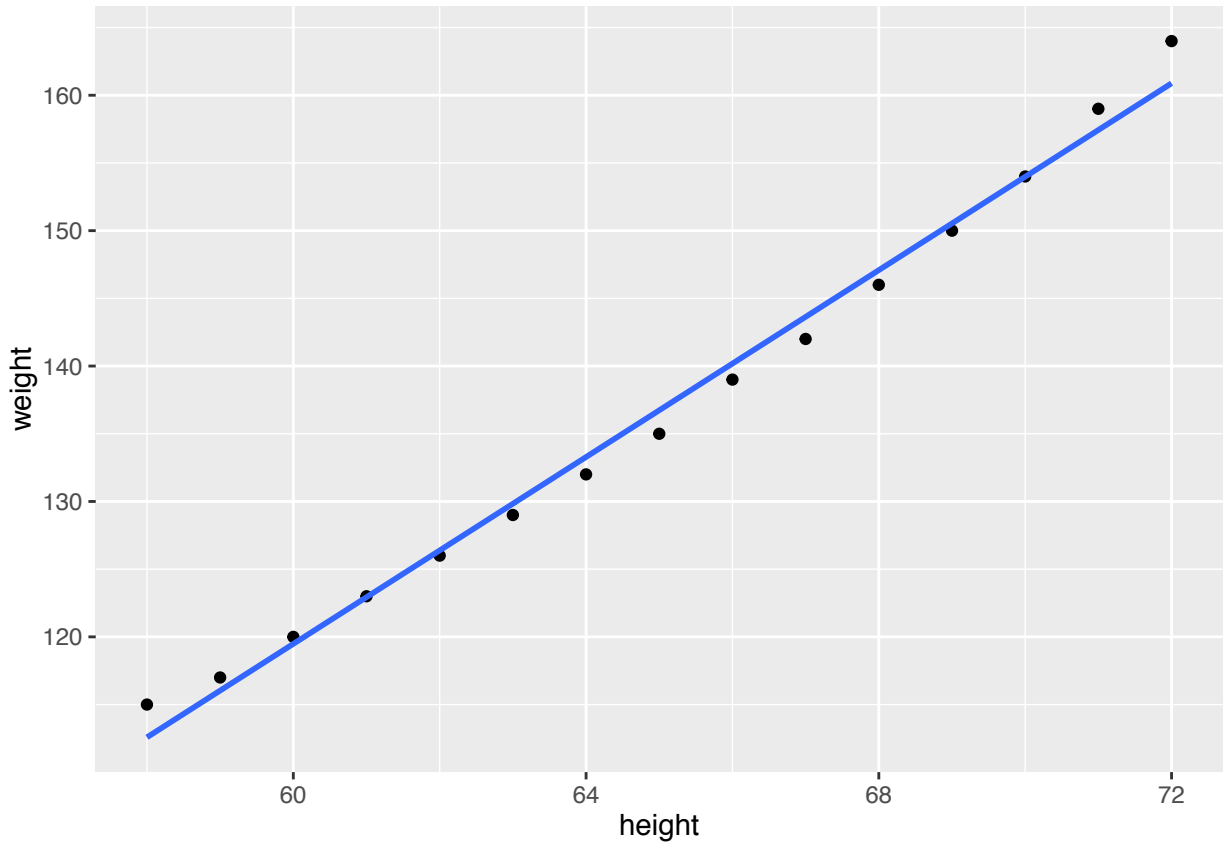
$$y = \frac{x}{\beta_0 + x\beta_1}\frac{1}{y} = \beta_1 + \beta_0\frac{1}{x}$$

Reguły kciuka dla transformacji zmiennych:

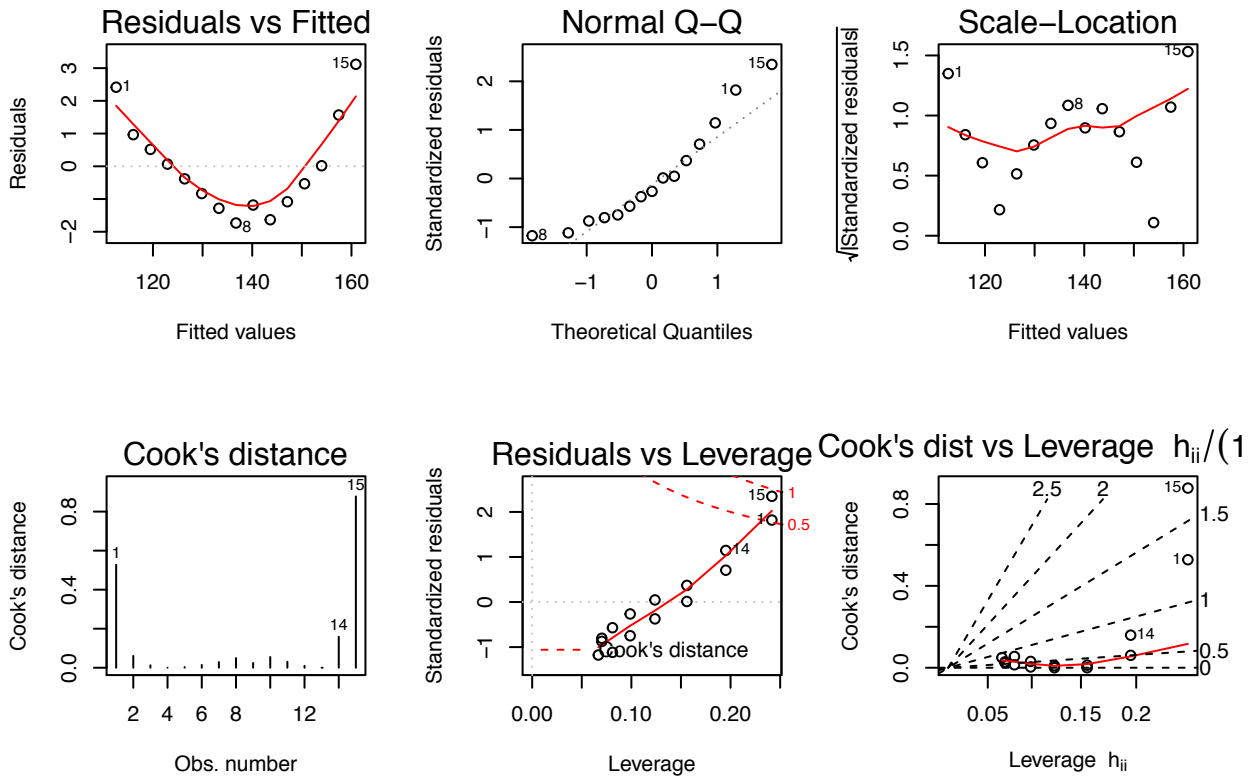
1. Stężenia - przekształcenie logarytmiczne
2. Procenty - arc sin
3. Przekształcenie powinno mieć logiczne uzasadnienie. Pytanie kontrolne: czy narysowanie wykresu z danymi w tej skali jest przekonywające?

Regresja wielomianowa

```
data(women)
ggplot(women, aes(x=height, y=weight)) +
  geom_point() +
  stat_smooth(method = "lm", se = F, fullrange = T)
```




```
fit = lm(weight ~ height, women)
par(mfrow=c(2,3))
plot(fit, 1:6)
```




```
par(mfrow=c(1,1))
```

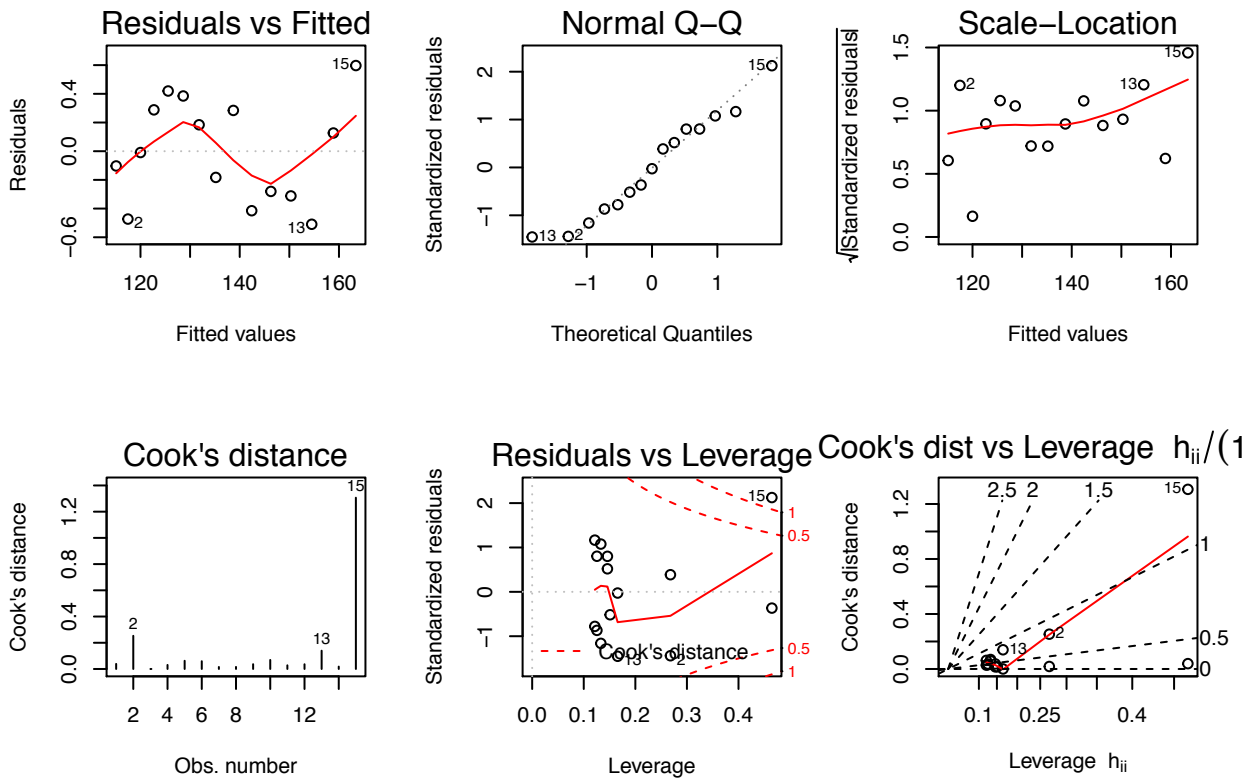
Bardzo złe residua, mimo, że na pierwszy rzut oka, wszystko wygląda dobrze. Problem z normalnością residuów wskazuje na problem ze skalą zmiennej objaśnianej. Dwa punkty wpływowe - najmniejszy i największy x.

Być może zależność nie jest liniowa. Spróbujmy dodać do modelu wyższe rzędy.

```
fit2 = lm(weight ~ height + I(height^2), women)
summary(fit2)
```

```
##
## Call:
## lm(formula = weight ~ height + I(height^2), data = women)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.50941 -0.29611 -0.00941  0.28615  0.59706
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 261.87818   25.19677  10.393 2.36e-07 ***
## height       -7.34832    0.77769  -9.449 6.58e-07 ***
## I(height^2)   0.08306    0.00598  13.891 9.32e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3841 on 12 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9994
## F-statistic: 1.139e+04 on 2 and 12 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,3))
plot(fit2, 1:6)
```

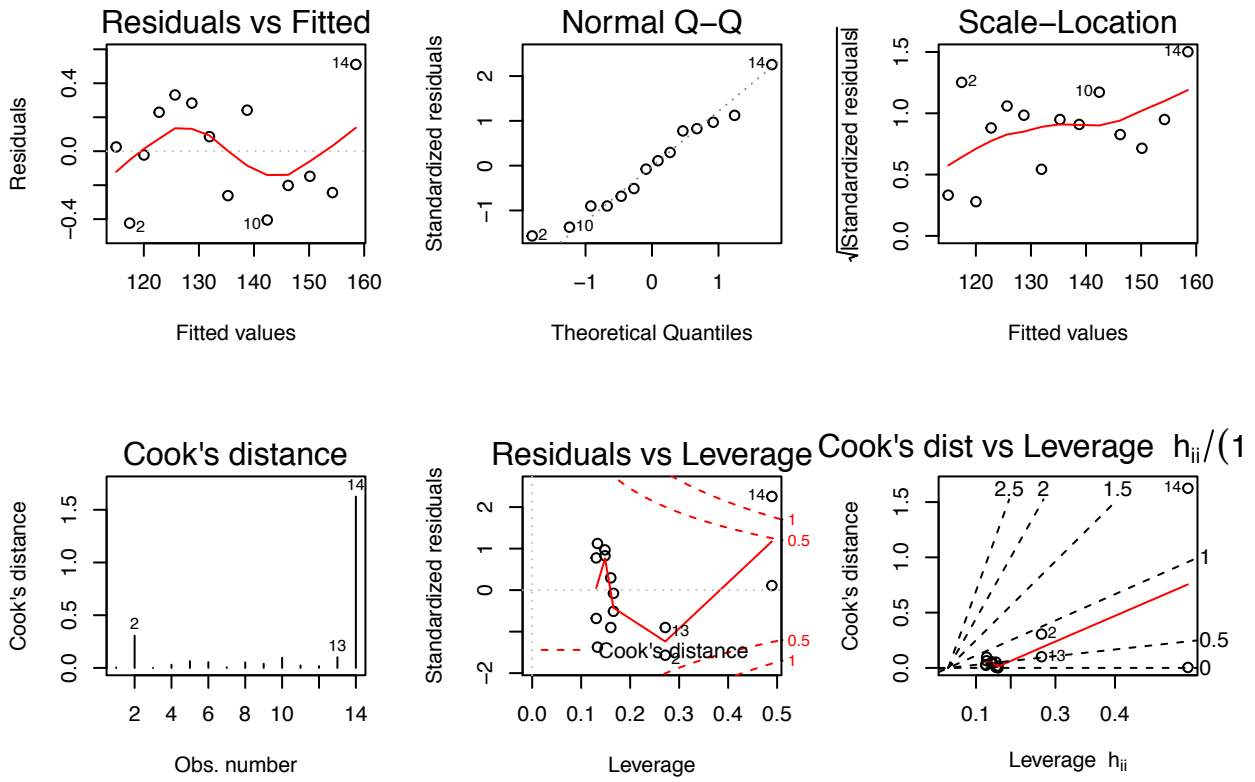



```
par(mfrow=c(1,1))
```

```
fit3 = lm(weight ~ height + I(height^2), women, subset = -15)
summary(fit3)
```

```
##
## Call:
## lm(formula = weight ~ height + I(height^2), data = women, subset = -15)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42390 -0.23317  0.00124  0.23839  0.51071
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 229.118681  24.367499   9.403 1.36e-06 ***
## height       -6.310027   0.757658  -8.328 4.45e-06 ***
## I(height^2)   0.074863   0.005871  12.751 6.22e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3168 on 11 degrees of freedom
## Multiple R-squared:  0.9996, Adjusted R-squared:  0.9995
## F-statistic: 1.278e+04 on 2 and 11 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,3))
plot(fit3, 1:6)
```




```
par(mfrow=c(1,1))
```

Zmiany są bardzo małe. Obserwacja 15 jest wpływowa, ale nie zmienia dużo.

Czy warto dodawać 3 potęgę? Niekonicznie, grozi nam zbytnie dopasowanie się do modelu (overfitting). Zasada dotycząca umieszczania kolejnych wielomianów od x jest następująca, jeśli używamy wielomianu o stopniu k, to powinniśmy też używać wielomianów o wykładnikach $< k$.

Metody radzenia sobie z obserwacjami odstającymi i wpływowymi

Dzięki diagnostyce regresji możemy zidentyfikować dwa rodzaje problematycznych obserwacji.

Outliers

Pierwszy to obserwacje odstające (ang. outliers). Są to punkty, do których regresja się źle dopasowała. Nie muszą być one szkodliwe dla naszej regresji.

```
which(rstudent(fit2)>2)
```

```
## 15
## 15
```

Dlaczego 2?

```
2*(1-pnorm(2))
```

```
## [1] 0.04550026
```

Influential observations

Drugi rodzaj to obserwacje wpływowe, czyli takie, których obecność wpływa na zmianę wartości współczynników bardziej niż pozostałe punkty. Te obserwacje też nie muszą być szkodliwe dla regresji. Wpływowość mierzymy poprzez wartość dźwigni (leverage).

```
which(hatvalues(fit2)>2*ncol(fit2$model)/nrow(fit2$model))
```

```
##  1 15
##  1 15
```

Zauważmy, że wpływowość zależy wyłącznie od macierzy X, a nie od y. Czyli jest to wartość niezależna od dopasowania modelu liniowego.

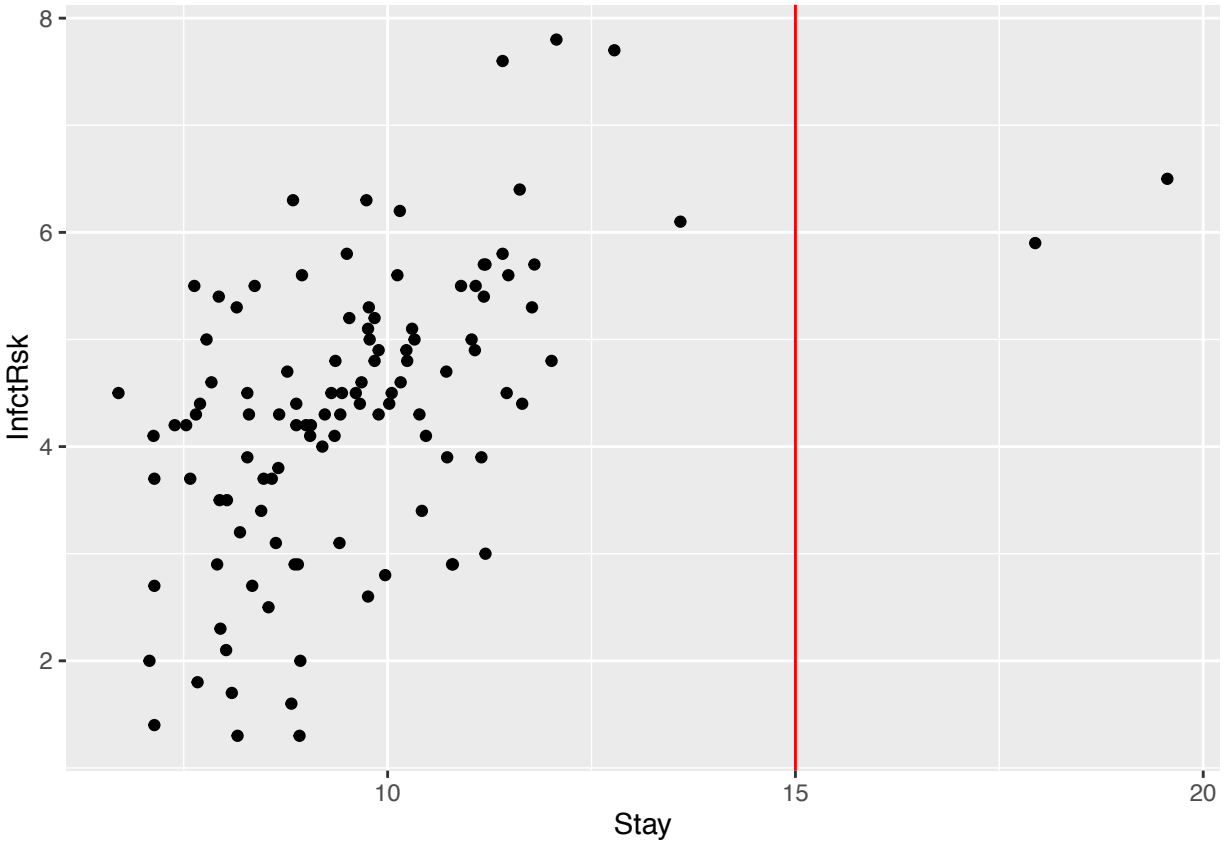
Bardzo dużym problemem są obserwacje, które są jednocześnie odstające i wpływowe. Ich niedoposowanie psuje wtedy cały model (patrz zestaw drugi z Anscombe quartet).

Co zrobić w przypadku problemów z konkretną obserwacją? Być może jest to niepoprawna dana. Należy sprawdzić czy nie została na przykład przypadkiem niepoprawnie wprowadzona.

Zanim usunie się daną, trzeba się zastanowić nad innym możliwymi rozwiązaniami. Może zależność nie jest liniowa i należy dokonać transformacji zmiennych objaśniających? Może brakuje nam jakiejś zmiennej objaśniającej?

Jeśli usuwamy obserwacje wpływowe o dużej wartości x, to powinniśmy wyraźnie zaznaczyć, że nasz model działa tylko na wartościach x z pewnego zakresu.

```
infekcje=read.csv("datasets/hospital_infct_03.txt", sep="\t", fileEncoding = "UTF-16")
ggplot(infekcje, aes(x=Stay, y=InfctRsk)) + geom_point() + geom_vline(xintercept = 15, col="red")
```



Usunięcie obserwacji musi być uzasadnione! Nie jest wystarczającym powodem fakt, że model źle się dopasowuje (czyli „podkręcenie" wartości R^2). Jeśli usunięcie punktu nie wpływa na wynik analizy, to lepiej

go zostawić (oczywiście umieszczając informację o obserwacji odstającej)

> First, foremost, and finally — it's okay to use your common sense and knowledge about the situation.
>
> from Stat 501 z PennStat