10 Regresja wielokrotna i krokowa

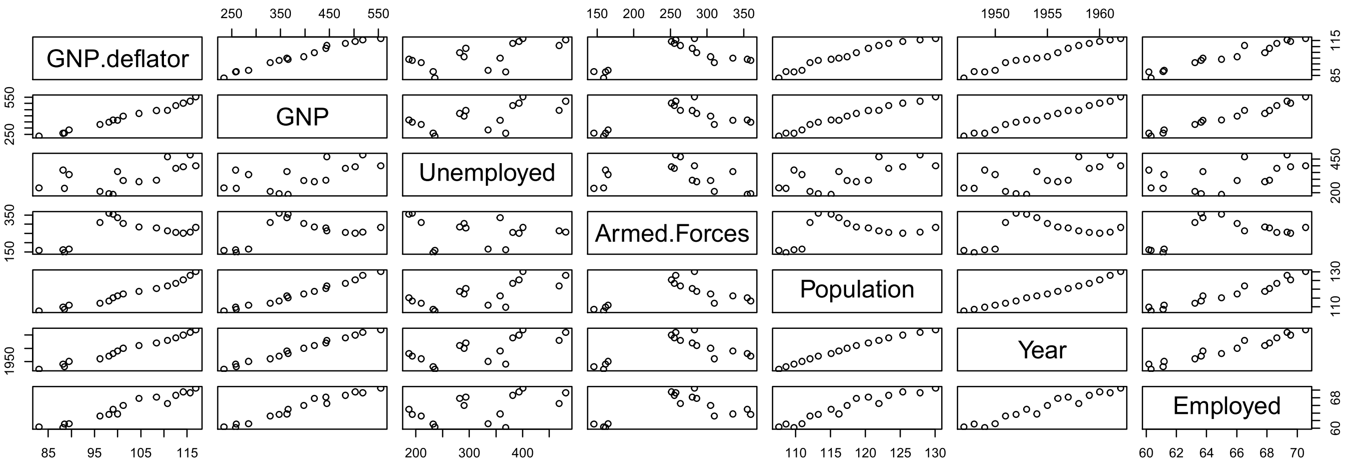
10.1 Przykład

Przykład. Zbiór danych `longley` zawiera 7 zmiennych makroekonomicznych. Chcemy modelować liczbę zatrudnionych za pomocą innych (niekoniecznie wszystkich) zmiennych przy użyciu modelu regresji wielokrotnej.

```
head(longley)
```

```
##      GNP.deflator     GNP Unemployed Armed.Forces Population Year Employed
## 1947         83.0 234.289      235.6        159.0    107.608 1947   60.323
## 1948         88.5 259.426      232.5        145.6    108.632 1948   61.122
## 1949         88.2 258.054      368.2        161.6    109.773 1949   60.171
## 1950         89.5 284.599      335.1        165.0    110.929 1950   61.187
## 1951         96.2 328.975      209.9        309.9    112.075 1951   63.221
## 1952         98.1 346.999      193.2        359.4    113.270 1952   63.639
```

```
pairs(longley)
```




```
# model pełny
model_1 <- lm(Employed ~ ., data = longley)
# model_1 <- lm(Employed ~ GNP.deflator + GNP + Unemployed +
#                          Armed.Forces + Population + Year,
#                data = longley)
model_1
```

```
##
## Call:
## lm(formula = Employed ~ ., data = longley)
##
## Coefficients:
##  (Intercept)  GNP.deflator           GNP    Unemployed  Armed.Forces
##   -3.482e+03     1.506e-02    -3.582e-02    -2.020e-02    -1.033e-02
##   Population          Year
##   -5.110e-02     1.829e+00
```

```
# estymacja parametrów
coef(model_1)
```

```
##   (Intercept)  GNP.deflator           GNP    Unemployed  Armed.Forces
## -3.482259e+03  1.506187e-02 -3.581918e-02 -2.020230e-02 -1.033227e-02
##    Population          Year
## -5.110411e-02  1.829151e+00
```

```r
confint(model_1)
```

```
##                      2.5 %        97.5 %
## (Intercept)  -5.496529e+03 -1.467988e+03
## GNP.deflator -1.770290e-01  2.071528e-01
## GNP          -1.115811e-01  3.994274e-02
## Unemployed   -3.125067e-02 -9.153930e-03
## Armed.Forces -1.517949e-02 -5.485050e-03
## Population   -5.625172e-01  4.603090e-01
## Year          7.987875e-01  2.859515e+00
```

```r
# podsumowanie modelu
# tj. reszty, estymacja punktowa, testy istotności dla współczynników regresji,
# R_adj^2, test istnotności modelu (test analizy wariancji w regresji)
summary(model_1)
```

```
##
## Call:
## lm(formula = Employed ~ ., data = longley)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41011 -0.15767 -0.02816  0.10155  0.45539
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.482e+03  8.904e+02  -3.911 0.003560 **
## GNP.deflator  1.506e-02  8.492e-02   0.177 0.863141
## GNP          -3.582e-02  3.349e-02  -1.070 0.312681
## Unemployed   -2.020e-02  4.884e-03  -4.136 0.002535 **
## Armed.Forces -1.033e-02  2.143e-03  -4.822 0.000944 ***
## Population   -5.110e-02  2.261e-01  -0.226 0.826212
## Year          1.829e+00  4.555e-01   4.016 0.003037 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3049 on 9 degrees of freedom
## Multiple R-squared:  0.9955, Adjusted R-squared:  0.9925
## F-statistic: 330.3 on 6 and 9 DF,  p-value: 4.984e-10
```

```r
# wartości dopasowane przez model
fitted(model_1)
```

```
##     1947     1948     1949     1950     1951     1952     1953     1954
## 60.05566 61.21601 60.12471 61.59711 62.91129 63.88831 65.15305 63.77418
##     1955     1956     1957     1958     1959     1960     1961     1962
## 66.00470 67.40161 68.18627 66.55206 68.81055 69.64967 68.98907 70.75776
```

```r
# reszty
residuals(model_1)
```

```
##        1947        1948        1949        1950        1951        1952
##  0.26734003 -0.09401394  0.04628717 -0.41011462  0.30971459 -0.24931122
##        1953        1954        1955        1956        1957        1958
## -0.16404896 -0.01318036  0.01430477  0.45539409 -0.01726893 -0.03905504
##        1959        1960        1961        1962
## -0.15554997 -0.08567131  0.34193151 -0.20675783
```

```r
# predykcja
new_data <- data.frame(GNP.deflator = 115.4,
                       GNP = 518.163,
                       Unemployed = 480.3,
                       Armed.Forces = 257.4,
                       Population = 127.857,
                       Year = 1963)
predict(model_1, new_data, interval = "prediction")
```

```
##        fit      lwr      upr
## 1 72.64695 70.55039 74.74351
```

```r
# redukcja modelu pełnego
summary(model_1)
```

```
##
## Call:
## lm(formula = Employed ~ ., data = longley)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41011 -0.15767 -0.02816  0.10155  0.45539
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.482e+03  8.904e+02  -3.911 0.003560 **
## GNP.deflator  1.506e-02  8.492e-02   0.177 0.863141
## GNP          -3.582e-02  3.349e-02  -1.070 0.312681
## Unemployed   -2.020e-02  4.884e-03  -4.136 0.002535 **
## Armed.Forces -1.033e-02  2.143e-03  -4.822 0.000944 ***
## Population   -5.110e-02  2.261e-01  -0.226 0.826212
## Year          1.829e+00  4.555e-01   4.016 0.003037 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3049 on 9 degrees of freedom
## Multiple R-squared:  0.9955, Adjusted R-squared:  0.9925
## F-statistic: 330.3 on 6 and 9 DF,  p-value: 4.984e-10
```

```r
model_2 <- lm(Employed ~ Unemployed + Armed.Forces + Year, data = longley)
# model_2 <- update(model_1, . ~ . - GNP.deflator - GNP - Population)
summary(model_2)
```

```
##
```

```
## Call:
## lm(formula = Employed ~ Unemployed + Armed.Forces + Year, data = longley)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.57285 -0.11989  0.04087  0.13979  0.75303
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.797e+03  6.864e+01 -26.183 5.89e-12 ***
## Unemployed   -1.470e-02  1.671e-03  -8.793 1.41e-06 ***
## Armed.Forces -7.723e-03  1.837e-03  -4.204  0.00122 **
## Year          9.564e-01  3.553e-02  26.921 4.24e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3321 on 12 degrees of freedom
## Multiple R-squared:  0.9928, Adjusted R-squared:  0.9911
## F-statistic: 555.2 on 3 and 12 DF,  p-value: 3.916e-13
```

Regresja krokowa

- Istnieje również inna metoda konstrukcji modeli z dużą liczbą zmiennych objaśniających niż konstruowanie pełnego modelu i szacowanie jego parametrów (tak jak robimy to w regresji wielokrotnej).
- Jest to procedura regresji krokowej, w której możemy odrzucić lub dodać zmienną na każdym kroku.
- Powiedzmy, że zaczynamy od modelu zawierającego tylko stałą - „regresja w przód” (możemy też zacząć od pełnego modelu - „regresja w tył”). W następnym kroku dodajemy najlepszą zmienną w sensie kryterium (np. test istotności, AIC, BIC). W następnym dodamy ponownie, ale możemy również sprawdzić co się dzieje, jakbyśmy usunęli z modelu zmienną dodaną w poprzednim kroku, itd.

Przykład (cd.).

```r
model_1 <- lm(Employed ~ ., data = longley)
model_2 <- lm(Employed ~ Unemployed + Armed.Forces + Year, data = longley)
# AIC (z wyrazem stałym)
AIC(model_1, model_2)
```

```
##         df      AIC
## model_1  8 14.18670
## model_2  5 15.52741
```

```r
n <- nrow(longley)
p <- 6
n * log(mean(model_1$residuals^2)) + 2 * (p + 1) + n * log(2 * pi) + n + 2
```

```
## [1] 14.1867
```

```r
p <- 3
n * log(mean(model_2$residuals^2)) + 2 * (p + 1) + n * log(2 * pi) + n + 2
```

```
## [1] 15.52741
```

```r
# BIC (z wyrazem stałym)
AIC(model_1, model_2, k = log(nrow(longley)))
```

```
##         df      AIC
```

```
## model_1  8 20.36741
## model_2  5 19.39035
```

```r
# AIC i BIC (bez wyrazu stałego)
extractAIC(model_1)[2]
```

```
## [1] -33.21933
```

```r
extractAIC(model_1, k = log(n))[2]
```

```
## [1] -27.81121
```

```r
p <- 6
n * log(mean(model_1$residuals^2)) + 2 * (p + 1)
```

```
## [1] -33.21933
```

```r
n * log(mean(model_1$residuals^2)) + log(n) * (p + 1)
```

```
## [1] -27.81121
```

```r
extractAIC(model_2)[2]
```

```
## [1] -31.87863
```

```r
extractAIC(model_2, k = log(n))[2]
```

```
## [1] -28.78827
```

```r
p <- 3
n * log(mean(model_2$residuals^2)) + 2 * (p + 1)
```

```
## [1] -31.87863
```

```r
n * log(mean(model_2$residuals^2)) + log(n) * (p + 1)
```

```
## [1] -28.78827
```

```r
# regresja krokowa
step(model_1)
```

```
## Start:  AIC=-33.22
## Employed ~ GNP.deflator + GNP + Unemployed + Armed.Forces + Population +
##     Year
##
##                Df Sum of Sq     RSS     AIC
## - GNP.deflator  1   0.00292 0.83935 -35.163
## - Population    1   0.00475 0.84117 -35.129
## - GNP           1   0.10631 0.94273 -33.305
## <none>                      0.83642 -33.219
## - Year          1   1.49881 2.33524 -18.792
## - Unemployed    1   1.59014 2.42656 -18.178
## - Armed.Forces  1   2.16091 2.99733 -14.798
##
## Step:  AIC=-35.16
## Employed ~ GNP + Unemployed + Armed.Forces + Population + Year
##
##                Df Sum of Sq    RSS     AIC
## - Population    1   0.01933 0.8587 -36.799
```

```
## <none>                        0.8393 -35.163
## - GNP           1    0.14637 0.9857 -34.592
## - Year          1    1.52725 2.3666 -20.578
## - Unemployed    1    2.18989 3.0292 -16.628
## - Armed.Forces  1    2.39752 3.2369 -15.568
##
## Step:  AIC=-36.8
## Employed ~ GNP + Unemployed + Armed.Forces + Year
##
##                Df Sum of Sq    RSS     AIC
## <none>                       0.8587 -36.799
## - GNP           1     0.4647 1.3234 -31.879
## - Year          1     1.8980 2.7567 -20.137
## - Armed.Forces  1     2.3806 3.2393 -17.556
## - Unemployed    1     4.0491 4.9077 -10.908
```

```
##
## Call:
## lm(formula = Employed ~ GNP + Unemployed + Armed.Forces + Year,
##     data = longley)
##
## Coefficients:
##  (Intercept)          GNP   Unemployed  Armed.Forces          Year
##   -3.599e+03   -4.019e-02   -2.088e-02    -1.015e-02     1.887e+00
```

```r
# step(model_1, direction = "backward")
step(model_1, k = log(nrow(longley)))
```

```
## Start:  AIC=-27.81
## Employed ~ GNP.deflator + GNP + Unemployed + Armed.Forces + Population +
##     Year
##
##                Df Sum of Sq     RSS     AIC
## - GNP.deflator  1   0.00292 0.83935 -30.528
## - Population    1   0.00475 0.84117 -30.493
## - GNP           1   0.10631 0.94273 -28.669
## <none>                      0.83642 -27.811
## - Year          1   1.49881 2.33524 -14.156
## - Unemployed    1   1.59014 2.42656 -13.542
## - Armed.Forces  1   2.16091 2.99733 -10.162
##
## Step:  AIC=-30.53
## Employed ~ GNP + Unemployed + Armed.Forces + Population + Year
##
##                Df Sum of Sq    RSS     AIC
## - Population    1   0.01933 0.8587 -32.936
## - GNP           1   0.14637 0.9857 -30.729
## <none>                      0.8393 -30.528
## - Year          1   1.52725 2.3666 -16.715
## - Unemployed    1   2.18989 3.0292 -12.765
## - Armed.Forces  1   2.39752 3.2369 -11.705
##
## Step:  AIC=-32.94
```

```
## Employed ~ GNP + Unemployed + Armed.Forces + Year
##
##                Df Sum of Sq    RSS     AIC
## <none>                      0.8587 -32.936
## - GNP           1    0.4647 1.3234 -28.788
## - Year          1    1.8980 2.7567 -17.046
## - Armed.Forces  1    2.3806 3.2393 -14.466
## - Unemployed    1    4.0491 4.9077  -7.818

##
## Call:
## lm(formula = Employed ~ GNP + Unemployed + Armed.Forces + Year,
##     data = longley)
##
## Coefficients:
##  (Intercept)           GNP    Unemployed  Armed.Forces          Year
##   -3.599e+03    -4.019e-02    -2.088e-02    -1.015e-02     1.887e+00
```

```
model_0 <- lm(Employed ~ 1, data = longley)
step(model_0, direction = "forward", scope = formula(model_1))
```

```
## Start:  AIC=41.17
## Employed ~ 1
##
##                Df Sum of Sq     RSS     AIC
## + GNP           1   178.973   6.036 -11.597
## + Year          1   174.552  10.457  -2.806
## + GNP.deflator  1   174.397  10.611  -2.571
## + Population    1   170.643  14.366   2.276
## + Unemployed    1    46.716 138.293  38.509
## + Armed.Forces  1    38.691 146.318  39.411
## <none>                      185.009  41.165
##
## Step:  AIC=-11.6
## Employed ~ GNP
##
##                Df Sum of Sq    RSS      AIC
## + Unemployed    1   2.45708 3.5791 -17.9598
## + Population    1   2.16178 3.8744 -16.6913
## + Year          1   1.12520 4.9109 -12.8980
## <none>                      6.0361 -11.5972
## + GNP.deflator  1   0.21194 5.8242 -10.1691
## + Armed.Forces  1   0.07665 5.9595  -9.8017
##
## Step:  AIC=-17.96
## Employed ~ GNP + Unemployed
##
##                Df Sum of Sq    RSS     AIC
## + Armed.Forces  1   0.82235 2.7567 -20.137
## <none>                      3.5791 -17.960
## + Year          1   0.33980 3.2393 -17.556
## + Population    1   0.09682 3.4822 -16.399
## + GNP.deflator  1   0.01884 3.5602 -16.044
```

```
##
## Step:  AIC=-20.14
## Employed ~ GNP + Unemployed + Armed.Forces
##
##                Df Sum of Sq     RSS     AIC
## + Year          1   1.89803 0.85868 -36.799
## + Population    1   0.39011 2.36660 -20.578
## <none>                      2.75671 -20.137
## + GNP.deflator  1   0.07288 2.68383 -18.566
##
## Step:  AIC=-36.8
## Employed ~ GNP + Unemployed + Armed.Forces + Year
##
##                Df Sum of Sq     RSS     AIC
## <none>                      0.85868 -36.799
## + Population    1  0.019332 0.83935 -35.163
## + GNP.deflator  1  0.017507 0.84117 -35.129

##
## Call:
## lm(formula = Employed ~ GNP + Unemployed + Armed.Forces + Year,
##     data = longley)
##
## Coefficients:
##  (Intercept)           GNP    Unemployed  Armed.Forces          Year
##   -3.599e+03    -4.019e-02    -2.088e-02    -1.015e-02     1.887e+00
```

```r
step(model_0, direction = "forward", scope = formula(model_1), k = log(nrow(longley)))
```

```
## Start:  AIC=41.94
## Employed ~ 1
##
##                Df Sum of Sq     RSS     AIC
## + GNP           1   178.973   6.036 -10.052
## + Year          1   174.552  10.457  -1.261
## + GNP.deflator  1   174.397  10.611  -1.025
## + Population    1   170.643  14.366   3.822
## + Unemployed    1    46.716 138.293  40.054
## + Armed.Forces  1    38.691 146.318  40.956
## <none>                      185.009  41.938
##
## Step:  AIC=-10.05
## Employed ~ GNP
##
##                Df Sum of Sq    RSS      AIC
## + Unemployed    1   2.45708 3.5791 -15.6420
## + Population    1   2.16178 3.8744 -14.3736
## + Year          1   1.12520 4.9109 -10.5802
## <none>                      6.0361 -10.0520
## + GNP.deflator  1   0.21194 5.8242  -7.8513
## + Armed.Forces  1   0.07665 5.9595  -7.4839
##
## Step:  AIC=-15.64
```

```
## Employed ~ GNP + Unemployed
##
##                Df Sum of Sq    RSS     AIC
## + Armed.Forces  1   0.82235 2.7567 -17.046
## <none>                      3.5791 -15.642
## + Year          1   0.33980 3.2393 -14.466
## + Population    1   0.09682 3.4822 -13.308
## + GNP.deflator  1   0.01884 3.5602 -12.954
##
## Step:  AIC=-17.05
## Employed ~ GNP + Unemployed + Armed.Forces
##
##                Df Sum of Sq     RSS     AIC
## + Year          1   1.89803 0.85868 -32.936
## <none>                      2.75671 -17.046
## + Population    1   0.39011 2.36660 -16.715
## + GNP.deflator  1   0.07288 2.68383 -14.703
##
## Step:  AIC=-32.94
## Employed ~ GNP + Unemployed + Armed.Forces + Year
##
##                Df Sum of Sq     RSS     AIC
## <none>                      0.85868 -32.936
## + Population    1  0.019332 0.83935 -30.528
## + GNP.deflator  1  0.017507 0.84117 -30.493
```

```
##
## Call:
## lm(formula = Employed ~ GNP + Unemployed + Armed.Forces + Year,
##     data = longley)
##
## Coefficients:
##  (Intercept)           GNP    Unemployed  Armed.Forces          Year
##   -3.599e+03    -4.019e-02    -2.088e-02    -1.015e-02     1.887e+00
```

10.2 Zadania

Zadanie 1. Zbiór danych w pliku Automobile.csv zawiera dane charakteryzujące różne typy samochodów.

```
##   symboling normalized.losses        make fuel.type aspiration num.of.doors
## 1         3                NA alfa-romero       gas        std          two
## 2         3                NA alfa-romero       gas        std          two
## 3         1                NA alfa-romero       gas        std          two
## 4         2               164        audi       gas        std         four
## 5         2               164        audi       gas        std         four
## 6         2                NA        audi       gas        std          two
##    body.style drive.wheels engine.location wheel.base length width height
## 1 convertible          rwd           front       88.6  168.8  64.1   48.8
## 2 convertible          rwd           front       88.6  168.8  64.1   48.8
## 3   hatchback          rwd           front       94.5  171.2  65.5   52.4
## 4       sedan          fwd           front       99.8  176.6  66.2   54.3
## 5       sedan          4wd           front       99.4  176.6  66.4   54.3
```

```
## 6       sedan          fwd           front       99.8  177.3  66.3   53.1
##   curb.weight engine.type num.of.cylinders engine.size fuel.system bore stroke
## 1        2548        dohc             four         130        mpfi 3.47   2.68
## 2        2548        dohc             four         130        mpfi 3.47   2.68
## 3        2823        ohcv              six         152        mpfi 2.68   3.47
## 4        2337         ohc             four         109        mpfi 3.19   3.40
## 5        2824         ohc             five         136        mpfi 3.19   3.40
## 6        2507         ohc             five         136        mpfi 3.19   3.40
##   compression.ratio horsepower peak.rpm city.mpg highway.mpg price
## 1               9.0        111     5000       21          27 13495
## 2               9.0        111     5000       21          27 16500
## 3               9.0        154     5000       19          26 16500
## 4              10.0        102     5500       24          30 13950
## 5               8.0        115     5500       18          22 17450
## 6               8.5        110     5500       19          25 15250
```

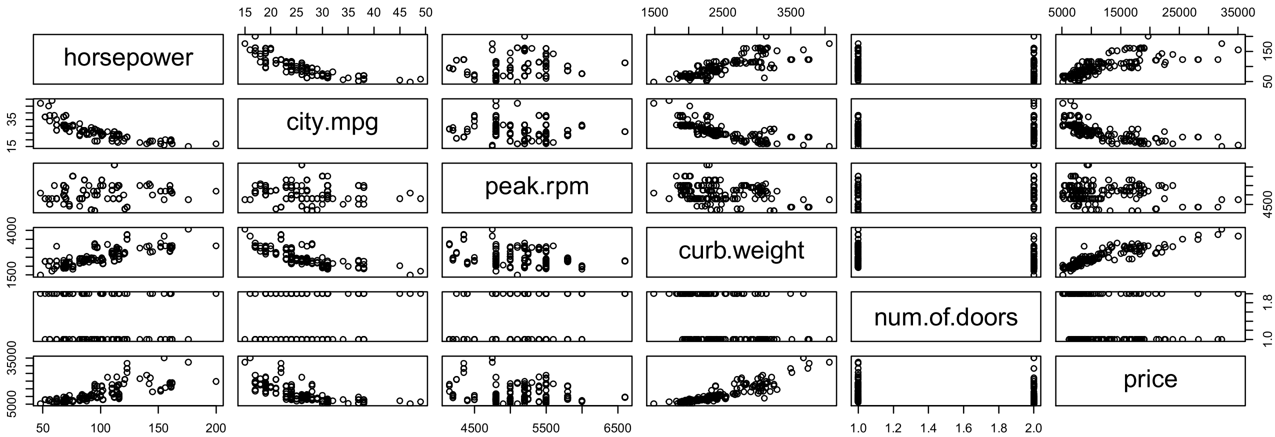
1. W tym zestawie danych występuja braki danych. Usuń wszystkie obserwacje, dla których nie mamy pełnych informacji o wszystkich zmiennych zawartych w zbiorze danych, używając funkcji `na.omit()`.

```
## wymiar nowych danych
```

```
## [1] 159  26
```

2. Interesuje nas zbudowanie modelu opisującego cenę samochodów w zależności od pewnych ich cech. Weźmy pod uwagę następujące zmienne: `horsepower`, `city.mpg`, `peak.rpm`, `curb.weight` i `num.of.doors` jako zmienne niezależne.

   - Dopasuj model regresji liniowej do tych danych.




```
##
## Call:
## lm(formula = price ~ horsepower + city.mpg + peak.rpm + curb.weight +
##     num.of.doors, data = auto_wna)
##
## Coefficients:
##     (Intercept)       horsepower          city.mpg          peak.rpm
##      -2.185e+04        2.743e+01         7.733e+01         4.847e-01
##     curb.weight  num.of.doorstwo
##       1.053e+01        5.516e+02
```

   - Jakie są wartości estymatorów współczynników regresji i przedziały ufności? Które zmienne są stymulantami a które destymulantami?

```
##     (Intercept)      horsepower        city.mpg        peak.rpm     curb.weight
##   -2.185283e+04    2.742792e+01    7.732533e+01    4.847128e-01    1.053105e+01
## num.of.doorstwo
##    5.515964e+02
```

```
##                         2.5 %        97.5 %
## (Intercept)     -3.161301e+04 -12092.655361
## horsepower      -2.484962e+00     57.340811
## city.mpg        -5.460472e+01    209.255385
## peak.rpm        -5.605008e-01      1.529926
## curb.weight      8.731667e+00     12.330436
## num.of.doorstwo -3.595924e+02   1462.785120
```

- Które współczynniki są statystycznie istotne w skontruowanym modelu? Jakie jest dopasowanie modelu?

```
##
## Call:
## lm(formula = price ~ horsepower + city.mpg + peak.rpm + curb.weight +
##     num.of.doors, data = auto_wna)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8235.7 -1413.0   -89.7   937.4  9759.4
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -2.185e+04  4.940e+03  -4.423 1.84e-05 ***
## horsepower       2.743e+01  1.514e+01   1.811    0.072 .
## city.mpg         7.733e+01  6.678e+01   1.158    0.249
## peak.rpm         4.847e-01  5.291e-01   0.916    0.361
## curb.weight      1.053e+01  9.108e-01  11.562  < 2e-16 ***
## num.of.doorstwo  5.516e+02  4.612e+02   1.196    0.234
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2597 on 153 degrees of freedom
## Multiple R-squared:  0.8109, Adjusted R-squared:  0.8048
## F-statistic: 131.2 on 5 and 153 DF,  p-value: < 2.2e-16
```

- Oblicz wartości dopasowane przez model oraz wartości reszt.

```
##         4         5         7         9        11        12        13        14
## 10077.611 15098.844 15249.651 18466.353 11280.669 10729.073 14240.554 14268.165
##        19        20        21        22        23        24        25        26
##  1791.835  5909.721  5726.711  5847.073  5383.121  8428.218  5789.850  6021.533
##        27        29        30        31        32        33        34        35
##  6021.533 11536.412 16171.344  4444.837  5244.631  5294.264  6441.563  6610.060
##        36        37        38        39        40        41        42        43
##  6627.140  6774.575  9504.115 10062.260  9668.630 10384.741 11543.572 10188.311
##        48        51        52        53        54        55        60        61
## 29256.003  5210.874  5393.510  5446.165  5315.811  5368.466 10456.347 10168.027
##        62        63        65        66        68        69        70        71
## 10456.347 10168.027 10325.993 13449.171 22347.106 24821.903 22688.081 25032.524
```

```
##        73        77        78        79        80        81        82        86 
## 25296.608  6289.377  6099.232  6731.096  8607.246 11283.398  9985.406  9823.459 
##        87        88        89        90        91        92        93        94 
## 10244.701 11079.326 11079.326  5402.039  7254.692  5707.439  5366.464  6272.134 
##        95        96        97        98        99       100       101       102 
##  6054.964  6865.855  5713.989  6409.038  6655.234  9890.130  9658.447 18744.853 
##       103       104       105       106       107       108       109       112 
## 20861.595 18530.917 19417.779 21076.356 20133.890 16504.197 18597.259 17028.549 
##       113       116       117       118       119       120       121       122 
## 19176.467 17083.405 19176.467 19110.371  6289.377  8428.218  5789.850  6021.533 
##       123       124       126       133       134       135       136       137 
##  8148.806 11536.412 16011.320 13875.944 13713.997 14391.966 14377.453 16793.526 
##       138       139       140       141       142       143       144       145 
## 16652.641  6952.124  7170.026  8433.753  7786.395  7757.106  9899.018  9695.244 
##       146       147       148       149       150       151       152       153 
## 11807.036  9004.096 11032.764  9986.506 13204.058  6336.440  6606.347  5791.474 
##       154       155       156       157       158       159       160       161 
##  8582.203  8378.212 17013.674  6628.622  6923.491  8451.542  8760.844  7384.128 
##       162       163       164       165       166       167       168       169 
##  6905.744  7095.303  8029.625  8398.212 10833.086 11201.673 12811.703 12769.579 
##       170       171       172       173       174       175       176       177 
## 12927.545 14275.519 14644.106 17392.711  9443.991 10767.381 10216.073 10216.073 
##       178       179       180       181       183       184       185       186 
## 10679.439 18522.082 18865.999 19465.659  9123.382  8925.756  8603.379  8405.753 
##       187       188       189       191       195       196       197       198 
##  9069.209  9476.020  9787.757  9078.471 16336.304 17621.093 16655.844 17782.666 
##       199       200       201       202       203       204       205 
## 18444.109 19623.587 16757.546 18682.970 17599.813 19270.000 17606.661
```

```
##           4           5           7           9          11          12 
##  3872.38860  2351.15555  2460.34881  5408.64732  5149.33069  6195.92704 
##          13          14          19          20          21          22 
##  6729.44647  6836.83500  3359.16525   385.27925   848.28880  -275.07295 
##          23          24          25          26          27          29 
##   993.87903  -471.21802   439.14971   670.46658  1587.46658 -2615.41226 
##          30          31          32          33          34          35 
## -3207.34381  2034.16262  1610.36919   104.73611    87.43730   518.94048 
##          36          37          38          39          40          41 
##   667.86006   520.42534 -1609.11460  -967.26032  -823.62974   -89.74124 
##          42          43          48          51          52          53 
##  1401.42812   156.68902  2993.99694   -15.87397   701.49018  1348.83492 
##          54          55          60          61          62          63 
##  1379.18922  2026.53396 -1611.34731 -1673.02725   138.65269    76.97275 
##          65          66          68          69          70          71 
##   919.00698  4830.82889  3204.89401  3426.09694  5487.91869  6567.47591 
##          73          77          78          79          80          81 
##  9759.39224  -900.37711    89.76754   -62.09554  -918.24589 -1324.39806 
##          82          86          87          88          89          90 
## -1486.40630 -2834.45885 -2055.70091 -1800.32640 -1800.32640    96.96127 
##          91          92          93          94          95          96 
##  -155.69190   941.56078  1482.53610  1076.86568  1244.03608   933.14513 
##          97          98          99         100         101         102 
```

```
##  1785.01141  1589.96202  1593.76616  -941.13028  -109.44715 -5245.85340
##         103         104         105         106         107         108
## -6462.59473 -5031.91727 -2218.77858 -1377.35637 -1734.89007 -4604.19683
##         109         112         113         116         117         118
## -5397.25921 -1448.54881 -2276.46704  -453.40466 -1226.46704  -960.37140
##         119         120         121         122         123         124
##  -717.37711  -471.21802   439.14971   670.46658  -539.80580 -2615.41226
##         126         133         134         135         136         137
##  6006.68036 -2025.94445 -1543.99700   648.03403  1132.54676  1356.47410
##         138         139         140         141         142         143
##  1967.35945 -1834.12417  -117.02643  -830.75259  -660.39477    17.89435
##         144         145         146         147         148         149
##    60.98200  -462.24445  -548.03567 -1541.09590  -834.76357 -1973.50592
##         150         151         152         153         154         155
## -1510.05754  -988.44041  -268.34691   696.52572 -1664.20289  -480.21208
##         156         157         158         159         160         161
## -8235.67421   309.37827   274.50883  -553.54226  -972.84358   353.87195
##         162         163         164         165         166         167
##  1452.25582  2162.69690    28.37473  -160.21207 -1535.08600 -1663.67279
##         168         169         170         171         172         173
## -4362.70319 -3130.57898 -2938.54475 -3076.51933 -3095.10613   276.28947
##         174         175         176         177         178         179
##  -495.99066   -69.38117  -228.07252   681.92748   568.56122 -1964.08195
##         180         181         183         184         185         186
## -2867.99868 -3775.65894 -1348.38201  -950.75627  -608.37882  -210.75307
##         187         188         189         191         195         196
##  -574.20931    18.98040   207.24268   901.52930 -3396.30442 -4206.09269
##         197         198         199         200         201         202
##  -670.84394 -1267.66643   -24.10880  -673.58655    87.45352   362.02963
##         203         204         205
##  3885.18734  3199.99996  5018.33919
```

3. Spróbuj zredukować model korzystając z regresji krokowej ("backward", "forward", AIC, BIC).

```
## Start:  AIC=2506.06
## price ~ horsepower + city.mpg + peak.rpm + curb.weight + num.of.doors
##
##                Df Sum of Sq        RSS    AIC
## - peak.rpm      1   5661937 1037719493 2504.9
## - city.mpg      1   9044038 1041101594 2505.4
## - num.of.doors  1   9647889 1041705445 2505.5
## <none>                      1032057556 2506.1
## - horsepower    1  22134795 1054192350 2507.4
## - curb.weight   1 901782660 1933840216 2603.9
##
## Step:  AIC=2504.93
## price ~ horsepower + city.mpg + curb.weight + num.of.doors
##
##                Df  Sum of Sq        RSS    AIC
## - city.mpg      1    6994707 1044714200 2504.0
## - num.of.doors  1    9518068 1047237561 2504.4
## <none>                       1037719493 2504.9
```

```
## - horsepower    1   32461892 1070181386 2507.8
## - curb.weight   1 1136974423 2174693916 2620.6
##
## Step:  AIC=2504
## price ~ horsepower + curb.weight + num.of.doors
##
##                Df  Sum of Sq        RSS    AIC
## - num.of.doors  1   12661847 1057376047 2503.9
## <none>                       1044714200 2504.0
## - horsepower    1   26482698 1071196898 2506.0
## - curb.weight   1 1155965636 2200679836 2620.5
##
## Step:  AIC=2503.91
## price ~ horsepower + curb.weight
##
##               Df  Sum of Sq        RSS    AIC
## <none>                      1057376047 2503.9
## - horsepower   1   42071205 1099447251 2508.1
## - curb.weight  1 1249455315 2306831362 2625.9

##
## Call:
## lm(formula = price ~ horsepower + curb.weight, data = auto_wna)
##
## Coefficients:
## (Intercept)   horsepower  curb.weight
##  -14608.000       27.404        9.519

## Start:  AIC=2524.47
## price ~ horsepower + city.mpg + peak.rpm + curb.weight + num.of.doors
##
##                Df Sum of Sq        RSS    AIC
## - peak.rpm      1   5661937 1037719493 2520.3
## - city.mpg      1   9044038 1041101594 2520.8
## - num.of.doors  1   9647889 1041705445 2520.9
## - horsepower    1  22134795 1054192350 2522.8
## <none>                      1032057556 2524.5
## - curb.weight   1 901782660 1933840216 2619.2
##
## Step:  AIC=2520.28
## price ~ horsepower + city.mpg + curb.weight + num.of.doors
##
##                Df  Sum of Sq        RSS    AIC
## - city.mpg      1    6994707 1044714200 2516.3
## - num.of.doors  1    9518068 1047237561 2516.7
## - horsepower    1   32461892 1070181386 2520.1
## <none>                       1037719493 2520.3
## - curb.weight   1 1136974423 2174693916 2632.8
##
## Step:  AIC=2516.27
## price ~ horsepower + curb.weight + num.of.doors
##
```

```
##                Df  Sum of Sq        RSS    AIC
## - num.of.doors  1   12661847 1057376047 2513.1
## - horsepower    1   26482698 1071196898 2515.2
## <none>                       1044714200 2516.3
## - curb.weight   1 1155965636 2200679836 2629.7
##
## Step:  AIC=2513.12
## price ~ horsepower + curb.weight
##
##               Df  Sum of Sq        RSS    AIC
## <none>                      1057376047 2513.1
## - horsepower   1   42071205 1099447251 2514.3
## - curb.weight  1 1249455315 2306831362 2632.1
```

```
##
## Call:
## lm(formula = price ~ horsepower + curb.weight, data = auto_wna)
##
## Coefficients:
## (Intercept)   horsepower  curb.weight
##  -14608.000       27.404        9.519
```

```
## Start:  AIC=2760.9
## price ~ 1
##
##                Df  Sum of Sq        RSS    AIC
## + curb.weight   1 4359325314 1099447251 2508.1
## + horsepower    1 3151941203 2306831362 2625.9
## + city.mpg      1 2616073039 2842699526 2659.2
## + peak.rpm      1  161334765 5297437800 2758.1
## + num.of.doors  1  143528709 5315243857 2758.7
## <none>                       5458772565 2760.9
##
## Step:  AIC=2508.12
## price ~ curb.weight
##
##                Df Sum of Sq        RSS    AIC
## + horsepower    1  42071205 1057376047 2503.9
## + num.of.doors  1  28250353 1071196898 2506.0
## + peak.rpm      1  21371766 1078075485 2507.0
## <none>                      1099447251 2508.1
## + city.mpg      1   1628352 1097818899 2509.9
##
## Step:  AIC=2503.91
## price ~ curb.weight + horsepower
##
##                Df Sum of Sq        RSS    AIC
## <none>                      1057376047 2503.9
## + num.of.doors  1  12661847 1044714200 2504.0
## + city.mpg      1  10138486 1047237561 2504.4
## + peak.rpm      1   3133537 1054242509 2505.4
```

```
##
```

```
## Call:
## lm(formula = price ~ curb.weight + horsepower, data = auto_wna)
##
## Coefficients:
## (Intercept)  curb.weight   horsepower
##  -14608.000        9.519       27.404
```

```
## Start:  AIC=2763.97
## price ~ 1
##
##                Df  Sum of Sq        RSS    AIC
## + curb.weight   1 4359325314 1099447251 2514.3
## + horsepower    1 3151941203 2306831362 2632.1
## + city.mpg      1 2616073039 2842699526 2665.3
## <none>                       5458772565 2764.0
## + peak.rpm      1  161334765 5297437800 2764.3
## + num.of.doors  1  143528709 5315243857 2764.8
##
## Step:  AIC=2514.26
## price ~ curb.weight
##
##                Df Sum of Sq        RSS    AIC
## + horsepower    1  42071205 1057376047 2513.1
## <none>                      1099447251 2514.3
## + num.of.doors  1  28250353 1071196898 2515.2
## + peak.rpm      1  21371766 1078075485 2516.2
## + city.mpg      1   1628352 1097818899 2519.1
##
## Step:  AIC=2513.12
## price ~ curb.weight + horsepower
##
##                Df Sum of Sq        RSS    AIC
## <none>                      1057376047 2513.1
## + num.of.doors  1  12661847 1044714200 2516.3
## + city.mpg      1  10138486 1047237561 2516.7
## + peak.rpm      1   3133537 1054242509 2517.7
```

```
##
## Call:
## lm(formula = price ~ curb.weight + horsepower, data = auto_wna)
##
## Coefficients:
## (Intercept)  curb.weight   horsepower
##  -14608.000        9.519       27.404
```

4. Dokonaj redukcji modelu metodą eliminacji wstecznej, tak aby w kolejnych krokach z pełnego modelu stopniowo usuwać najmniej istotną zmienną niezależną, aż otrzymamy model ze wszystkimi istotnymi zmiennymi niezależnymi. Jakie było zachowanie odpowiedniego współczynnika determinacji w kolejnych modelach?

```
##                    Estimate   Std. Error   t value     Pr(>|t|)
## (Intercept)     -18434.71356 3236.8236673 -5.695310 6.068463e-08
## horsepower          31.64419   14.4173920  2.194862 2.967071e-02
```

```
## city.mpg            67.03355   65.7941206  1.018838 3.098778e-01
## curb.weight         10.09671    0.7772917 12.989598 1.600643e-26
## num.of.doorstwo    547.85114  460.9648332  1.188488 2.364708e-01
```

```
## [1] 0.8049611
```

```
##                      Estimate   Std. Error    t value     Pr(>|t|)
## (Intercept)     -15395.704186 1257.0623108 -12.247368 1.496423e-24
## horsepower          22.757081   11.4806990   1.982203 4.922435e-02
## curb.weight          9.917956    0.7573256  13.096027 7.371627e-27
## num.of.doorstwo    623.608183  454.9840593   1.370615 1.724765e-01
```

```
## [1] 0.8049132
```

```
##                 Estimate   Std. Error   t value     Pr(>|t|)
## (Intercept) -14607.99973 1121.1401301 -13.02959 1.000781e-26
## horsepower      27.40398   10.9995177   2.49138 1.377104e-02
## curb.weight      9.51894    0.7011011  13.57713 3.237394e-28
```

```
## [1] 0.8038145
```

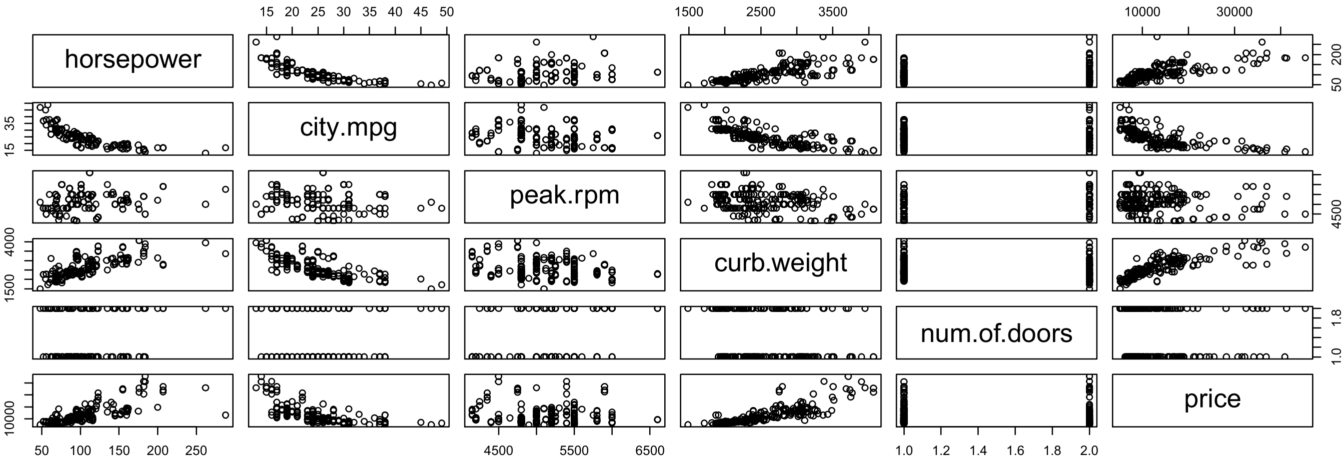
5. Zamiast usuwać obserwacje z brakującymi danymi, jak to zrobiliśmy w punkcie 1, uzupełnij je za pomocą średniej i mediany sąsiednich wartości dla zmiennych ilościowych i porządkowych, odpowiednio. Aby to zrobić, użyj funkcji `impute()` dostępnej w pakiecie `Hmisc`. W przypadku takich danych postępuj zgodnie z instrukcjami w punktach 2-4.

```
##    horsepower       city.mpg        peak.rpm     curb.weight   num.of.doors
##  Min.   : 48.0   Min.   :13.00   Min.   :4150   Min.   :1488   four:114    
##  1st Qu.: 70.0   1st Qu.:19.00   1st Qu.:4800   1st Qu.:2145   two : 89    
##  Median : 95.0   Median :24.00   Median :5200   Median :2414   NA's:  2    
##  Mean   :104.3   Mean   :25.22   Mean   :5125   Mean   :2556               
##  3rd Qu.:116.0   3rd Qu.:30.00   3rd Qu.:5500   3rd Qu.:2935               
##  Max.   :288.0   Max.   :49.00   Max.   :6600   Max.   :4066               
##  NA's   :2                       NA's   :2                                 
##      price      
##  Min.   : 5118  
##  1st Qu.: 7775  
##  Median :10295  
##  Mean   :13207  
##  3rd Qu.:16500  
##  Max.   :45400  
##  NA's   :4      
```

```
##    horsepower       city.mpg        peak.rpm     curb.weight    num.of.doors  
##  Min.   : 48.0   Min.   :13.00   Min.   :4150   Min.   :1488   Min.   :1.000  
##  1st Qu.: 70.0   1st Qu.:19.00   1st Qu.:4800   1st Qu.:2145   1st Qu.:1.000  
##  Median : 95.0   Median :24.00   Median :5200   Median :2414   Median :1.000  
##  Mean   :104.3   Mean   :25.22   Mean   :5125   Mean   :2556   Mean   :1.434  
##  3rd Qu.:116.0   3rd Qu.:30.00   3rd Qu.:5500   3rd Qu.:2935   3rd Qu.:2.000  
##  Max.   :288.0   Max.   :49.00   Max.   :6600   Max.   :4066   Max.   :2.000  
##      price      
##  Min.   : 5118  
##  1st Qu.: 7788  
##  Median :10595  
##  Mean   :13207  
```

```
##   3rd Qu.:16500
##   Max.   :45400
```

```
## 2.
```




```
##
## Call:
## lm(formula = price ~ horsepower + city.mpg + peak.rpm + curb.weight +
##     num.of.doors, data = auto_sel)
##
## Coefficients:
##  (Intercept)    horsepower      city.mpg      peak.rpm   curb.weight
##   -2.745e+04     6.722e+01     1.413e+02     6.572e-01     1.017e+01
## num.of.doors
##    5.050e+02
```

```
##   (Intercept)    horsepower      city.mpg      peak.rpm   curb.weight
## -2.744867e+04  6.721715e+01  1.413170e+02  6.572019e-01  1.017053e+01
##  num.of.doors
##  5.049619e+02
```

```
##                      2.5 %        97.5 %
## (Intercept)  -4.198964e+04 -12907.700033
## horsepower    3.717412e+01     97.260183
## city.mpg     -2.781769e+01    310.451628
## peak.rpm     -8.865075e-01      2.200911
## curb.weight   7.723371e+00     12.617692
## num.of.doors -7.993339e+02   1809.257683
```

```
##
## Call:
## lm(formula = price ~ horsepower + city.mpg + peak.rpm + curb.weight +
##     num.of.doors, data = auto_sel)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -20128  -2083   -138   1379  16751
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.745e+04  7.374e+03  -3.722 0.000257 ***
```

```
## horsepower    6.722e+01  1.524e+01   4.412 1.68e-05 ***
## city.mpg      1.413e+02  8.577e+01   1.648 0.101007
## peak.rpm      6.572e-01  7.828e-01   0.840 0.402185
## curb.weight   1.017e+01  1.241e+00   8.196 3.00e-14 ***
## num.of.doors  5.050e+02  6.614e+02   0.763 0.446100
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4182 on 199 degrees of freedom
## Multiple R-squared:  0.7245, Adjusted R-squared:  0.7176
## F-statistic: 104.7 on 5 and 199 DF,  p-value: < 2.2e-16
```

```
##         1         2         3         4         5         6         7         8
## 13190.535 13190.535 18595.135 10687.189 15666.159 12752.293 15674.800 16793.559
##         9        10        11        12        13        14        15        16
## 19869.950 21242.310 11770.668 11265.706 15017.431 15071.849 17879.986 23950.589
##        17        18        19        20        21        22        23        24
## 25981.131 26606.168  1915.053  6244.963  6095.970  6055.273  5207.372  9066.510
##        25        26        27        28        29        30        31        32
##  5627.928  5851.680  5851.680  9202.291 11431.313 17868.134  4961.068  5493.989
##        33        34        35        36        37        38        39        40
##  5262.202  6583.307  6746.035  6790.282  6932.669  9710.564 10249.602  9897.198
##        41        42        43        44        45        46        47        48
## 10588.794 12118.960 10751.530  8613.936  6244.963  6095.970 14094.645 31481.352
##        49        50        51        52        53        54        55        56
## 31481.352 36468.874  4879.841  5122.863  5173.716  5075.575  5126.428 10901.649
##        57        58        59        60        61        62        63        64
## 10901.649 10952.501 14266.179 10293.020 10042.321 10293.020 10042.321 10348.195
##        65        66        67        68        69        70        71        72
## 10194.879 14248.699 12497.433 23041.219 25431.293 23342.770 25634.704 26895.516
##        73        74        75        76        77        78        79        80
## 26841.098 30025.164 28648.577 20891.531  6482.436  5898.968  6509.199  9239.409
##        81        82        83        84        85        86        87        88
## 12327.501  9972.292 18091.886 18986.893 19037.745  9843.640 10250.461 12158.167
##        89        90        91        92        93        94        95        96
## 12158.167  5209.645  7285.990  5504.590  5203.039  6077.705  5840.218  6623.349
##        97        98        99       100       101       102       103       104
##  5538.667  6209.922  6419.938 10445.677 10221.925 20570.930 22615.206 20497.595
##       105       106       107       108       109       110       111       112
## 21652.170 24749.818 22343.766 16262.390 18641.371 18398.202 20587.154 16687.335
##       113       114       115       116       117       118       119       120
## 19200.750 18823.146 21146.533 16821.769 19200.750 20658.924  6482.436  9066.510
##       121       122       123       124       125       126       127       128
##  5627.928  5851.680  7906.127 11431.313 17939.328 17726.673 21785.066 21785.066
##       129       130       131       132       133       134       135       136
## 22232.570 33335.099 12912.585 12207.254 14406.377 14277.725 14904.733 14918.468
##       137       138       139       140       141       142       143       144
## 19174.481 19066.170  6649.940  6595.560  7816.024  8060.598  7690.124 10265.437
##       145       146       147       148       149       150       151       152
##  9370.990 12591.604  8970.057 11293.731  9585.642 13874.161  6017.883  6011.994
##       153       154       155       156       157       158       159       160
##  5252.769  7947.960  7484.397 15824.233  6320.445  6605.219  7720.595  8285.863
```

```
##       161       162       163       164       165       166       167       168
##  7583.197  6454.802  6637.872  7579.096  7935.065 12137.600 12493.569 13737.767
##       169       170       171       172       173       174       175       176
## 13697.085 13849.643 15151.471 15507.440 18161.948  9755.364 10382.977 10367.737
##       177       178       179       180       181       182       183       184
## 10367.737 10815.240 20894.504 21160.008 21629.888 21691.982  8435.412  9007.282
##       185       186       187       188       189       190       191       192
##  7960.962  8532.831  9173.575  9398.655 10459.079  9541.391  9205.763 13813.593
##       193       194       195       196       197       198       199       200
## 11477.725 12186.006 17134.813 18375.618 17510.052 18598.299 20668.854 21807.954
##       201       202       203       204       205
## 17541.634 20989.177 18855.344 19728.717 18095.125
```

```
##            1            2            3            4            5            6
##    304.46471   3309.46471  -2095.13493   3262.81115   1783.84136   2497.70662
##            7            8            9           10           11           12
##   2035.19953   2126.44112   4005.05039  -8035.18068   4659.33202   5659.29390
##           13           14           15           16           17           18
##   5952.56858   6033.15126   6685.01422   6809.41069  15333.86915  10273.83162
##           19           20           21           22           23           24
##   3235.94685     50.03666    479.02995   -483.27332   1169.62848  -1109.50973
##           25           26           27           28           29           30
##    601.07204    840.32035   1757.32035   -644.29131  -2510.31292  -4904.13419
##           31           32           33           34           35           36
##   1517.93247   1361.01059    136.79764    -54.30670    382.96480    504.71800
##           37           38           39           40           41           42
##    362.33057  -1815.56413  -1154.60228  -1052.19836   -293.79448    826.03976
##           43           44           45           46           47           48
##   -406.53002  -1828.93587   6962.16601   7111.15930  -3046.64478    768.64785
##           49           50           51           52           53           54
##   4068.64785   -468.87404    315.15897    972.13669   1621.28403   1619.42467
##           55           56           57           58           59           60
##   2268.57202     43.35142    943.35142   2692.49876   1378.82149  -1448.02008
##           61           62           63           64           65           66
##  -1547.32148    301.97992    202.67852    446.80464   1050.12055   4031.30137
##           67           68           69           70           71           72
##   5846.56664   2510.78150   2816.70670   4833.23024   5965.29608   7288.48419
##           73           74           75           76           77           78
##   8214.90152  10934.83623  16751.42259  -4388.53133  -1093.43563    290.03237
##           79           80           81           82           83           84
##    159.80051  -1550.40876  -2368.50141  -1473.29184  -5462.88588  -4117.89261
##           85           86           87           88           89           90
##  -4548.74526  -2854.63960  -2061.46085  -2879.16706  -2879.16706    289.35500
##           91           92           93           94           95           96
##   -186.98964   1144.40960   1645.96086   1271.29519   1458.78207   1175.65118
##           97           98           99          100          101          102
##   1960.33333   1789.07828   1829.06180  -1496.67652   -672.92483  -7071.92964
##          103          104          105          106          107          108
##  -8216.20638  -6998.59499  -4453.16993  -5050.81818  -3944.76604  -4362.39001
##          109          110          111          112          113          114
##  -5441.37080  -5958.20153  -6727.15363  -1107.33491  -2300.75000  -2128.14643
##          115          116          117          118          119          120
```

```
## -4071.53284  -191.76922 -1250.75000 -2508.92444  -910.43563 -1109.50973
##         121         122         123         124         125         126
##   601.07204   840.32035  -297.12692 -2510.31292 -5175.32791  4291.32669
##         127         128         129         130         131         132
## 10742.93382 12242.93382 14795.43045 -20127.96980 -3617.58498 -2312.25366
##         133         134         135         136         137         138
## -2556.37703 -2107.72480   135.26695   591.53174 -1024.48083  -446.16966
##         139         140         141         142         143         144
## -1531.93993   457.44008  -213.02365  -934.59825    84.87648  -305.43659
##         145         146         147         148         149         150
##  -137.98996 -1332.60375 -1507.05739 -1095.73069 -1572.64158 -2180.16114
##         151         152         153         154         155         156
##  -669.88303   326.00563  1235.23079 -1029.95994   413.60262 -7046.23285
##         157         158         159         160         161         162
##   617.55549   592.78062   177.40529  -497.86258   154.80285  1903.19766
##         163         164         165         166         167         168
##  2620.12810   478.90385   302.93526 -2839.60004 -2955.56863 -5288.76733
##         169         170         171         172         173         174
## -4058.08520 -3860.64317 -3952.47114 -3958.43973  -492.94834  -807.36385
##         175         176         177         178         179         180
##   315.02272  -379.73665   530.26335   432.75999 -4336.50360 -5162.00787
##         181         182         183         184         185         186
## -5939.88827 -5941.98192  -660.41202 -1032.28159    34.03827  -337.83131
##         187         188         189         190         191         192
##  -678.57476    96.34520  -464.07883  2053.60918   774.23670  -518.59328
##         193         194         195         196         197         198
##  2367.27500   103.99429 -4194.81287 -4960.61766 -1525.05205 -2083.29888
##         199         200         201         202         203         204
## -2248.85442 -2857.95390  -696.63411 -1944.17655  2629.65563  2741.28261
##         205
##  4529.87534
```

```
## 3.
```

```
## Start:  AIC=3424.69
## price ~ horsepower + city.mpg + peak.rpm + curb.weight + num.of.doors
##
##                Df  Sum of Sq        RSS    AIC
## - num.of.doors  1   10192607 3490187999 3423.3
## - peak.rpm      1   12324997 3492320389 3423.4
## <none>                       3479995392 3424.7
## - city.mpg      1   47472671 3527468063 3425.5
## - horsepower    1  340402702 3820398094 3441.8
## - curb.weight   1 1174580109 4654575501 3482.3
##
## Step:  AIC=3423.29
## price ~ horsepower + city.mpg + peak.rpm + curb.weight
##
##               Df  Sum of Sq        RSS    AIC
## - peak.rpm     1   12030940 3502218939 3422.0
## <none>                      3490187999 3423.3
## - city.mpg     1   48682445 3538870444 3424.1
```

```
## - horsepower   1  440974262 3931162261 3445.7
## - curb.weight  1 1240381716 4730569715 3483.6
##
## Step:  AIC=3422
## price ~ horsepower + city.mpg + curb.weight
##
##               Df  Sum of Sq        RSS    AIC
## <none>                      3502218939 3422.0
## - city.mpg     1   38636782 3540855721 3422.2
## - horsepower   1  556659511 4058878450 3450.2
## - curb.weight  1 1750422882 5252641821 3503.1

##
## Call:
## lm(formula = price ~ horsepower + city.mpg + curb.weight, data = auto_sel)
##
## Coefficients:
## (Intercept)   horsepower     city.mpg  curb.weight
##  -21384.432       75.415      121.380        9.261

## Start:  AIC=3444.63
## price ~ horsepower + city.mpg + peak.rpm + curb.weight + num.of.doors
##
##                 Df  Sum of Sq        RSS    AIC
## - num.of.doors  1   10192607 3490187999 3439.9
## - peak.rpm      1   12324997 3492320389 3440.0
## - city.mpg      1   47472671 3527468063 3442.1
## <none>                       3479995392 3444.6
## - horsepower    1  340402702 3820398094 3458.4
## - curb.weight   1 1174580109 4654575501 3498.9
##
## Step:  AIC=3439.91
## price ~ horsepower + city.mpg + peak.rpm + curb.weight
##
##               Df  Sum of Sq        RSS    AIC
## - peak.rpm     1   12030940 3502218939 3435.3
## - city.mpg     1   48682445 3538870444 3437.4
## <none>                      3490187999 3439.9
## - horsepower   1  440974262 3931162261 3459.0
## - curb.weight  1 1240381716 4730569715 3496.9
##
## Step:  AIC=3435.29
## price ~ horsepower + city.mpg + curb.weight
##
##               Df  Sum of Sq        RSS    AIC
## - city.mpg     1   38636782 3540855721 3432.2
## <none>                      3502218939 3435.3
## - horsepower   1  556659511 4058878450 3460.2
## - curb.weight  1 1750422882 5252641821 3513.1
##
## Step:  AIC=3432.22
## price ~ horsepower + curb.weight
```

```
##
##                Df  Sum of Sq        RSS    AIC
## <none>                       3540855721 3432.2
## - horsepower   1  580023407 4120879128 3458.0
## - curb.weight  1 1834490017 5375345738 3512.5
```

```
##
## Call:
## lm(formula = price ~ horsepower + curb.weight, data = auto_sel)
##
## Coefficients:
## (Intercept)   horsepower  curb.weight
##  -15818.459       64.615        8.722
```

```
## Start:  AIC=3678.97
## price ~ 1
##
##                Df  Sum of Sq        RSS    AIC
## + curb.weight   1 8510293560 4.1209e+09 3451.3
## + horsepower    1 7255826951 5.3753e+09 3505.8
## + city.mpg      1 5627042447 7.0041e+09 3560.1
## + peak.rpm      1  128478511 1.2503e+10 3678.9
## <none>                       1.2631e+10 3679.0
## + num.of.doors  1   22223129 1.2609e+10 3680.6
##
## Step:  AIC=3451.35
## price ~ curb.weight
##
##                Df Sum of Sq        RSS    AIC
## + horsepower    1 580023407 3540855721 3422.2
## + peak.rpm      1 188393930 3932485198 3443.8
## + num.of.doors  1 172156795 3948722333 3444.6
## + city.mpg      1  62000678 4058878450 3450.2
## <none>                      4120879128 3451.3
##
## Step:  AIC=3422.25
## price ~ curb.weight + horsepower
##
##                Df Sum of Sq        RSS    AIC
## + city.mpg      1  38636782 3502218939 3422.0
## <none>                      3540855721 3422.2
## + num.of.doors  1  11184104 3529671617 3423.6
## + peak.rpm      1   1985277 3538870444 3424.1
##
## Step:  AIC=3422
## price ~ curb.weight + horsepower + city.mpg
##
##                Df Sum of Sq        RSS    AIC
## <none>                      3502218939 3422.0
## + peak.rpm      1  12030940 3490187999 3423.3
## + num.of.doors  1   9898550 3492320389 3423.4
```

```
##
```

```
## Call:
## lm(formula = price ~ curb.weight + horsepower + city.mpg, data = auto_sel)
##
## Coefficients:
## (Intercept)  curb.weight   horsepower     city.mpg
##  -21384.432        9.261       75.415      121.380
```

```
## Start:  AIC=3682.29
## price ~ 1
##
##                Df  Sum of Sq        RSS    AIC
## + curb.weight   1 8510293560 4.1209e+09 3458.0
## + horsepower    1 7255826951 5.3753e+09 3512.5
## + city.mpg      1 5627042447 7.0041e+09 3566.7
## <none>                       1.2631e+10 3682.3
## + peak.rpm      1  128478511 1.2503e+10 3685.5
## + num.of.doors  1   22223129 1.2609e+10 3687.3
##
## Step:  AIC=3457.99
## price ~ curb.weight
##
##                Df Sum of Sq        RSS    AIC
## + horsepower    1 580023407 3540855721 3432.2
## + peak.rpm      1 188393930 3932485198 3453.7
## + num.of.doors  1 172156795 3948722333 3454.6
## <none>                      4120879128 3458.0
## + city.mpg      1  62000678 4058878450 3460.2
##
## Step:  AIC=3432.22
## price ~ curb.weight + horsepower
##
##                Df Sum of Sq        RSS    AIC
## <none>                      3540855721 3432.2
## + city.mpg      1  38636782 3502218939 3435.3
## + num.of.doors  1  11184104 3529671617 3436.9
## + peak.rpm      1   1985277 3538870444 3437.4
```

```
##
## Call:
## lm(formula = price ~ curb.weight + horsepower, data = auto_sel)
##
## Coefficients:
## (Intercept)  curb.weight   horsepower
##  -15818.459        8.722       64.615
```

```
## 4.
```

```
##                  Estimate   Std. Error    t value     Pr(>|t|)
## (Intercept) -2.635706e+04 7226.3754586 -3.6473412 3.379537e-04
## horsepower   7.139625e+01   14.2029379  5.0268648 1.104343e-06
## city.mpg     1.430558e+02   85.6502655  1.6702319 9.643773e-02
## curb.weight  9.855041e+00    1.1689345  8.4307899 6.749078e-15
## peak.rpm     6.492573e-01    0.7819454  0.8303102 4.073535e-01
```

```
## [1] 0.7181583

##                  Estimate   Std. Error   t value     Pr(>|t|)
## (Intercept) -21384.431936 4040.8655146 -5.292042 3.151748e-07
## horsepower      75.415001   13.3424794  5.652248 5.374600e-08
## city.mpg       121.380259   81.5119348  1.489110 1.380257e-01
## curb.weight      9.261327    0.9240072 10.023003 1.962328e-19

## [1] 0.7185938

##                 Estimate   Std. Error    t value     Pr(>|t|)
## (Intercept) -15818.45917 1540.0560178 -10.271353 3.508074e-20
## horsepower      64.61495   11.2328166   5.752337 3.223308e-08
## curb.weight      8.72178    0.8525618  10.230085 4.645579e-20

## [1] 0.7168977
```

6. Korzystając z ostatecznych modeli uzyskanych dla obu zbiorów danych, wykonaj prognozę ceny samochodu, dla którego zmienne `curb.weight` i `horsepower` są równe 2823 i 154, odpowiednio. Który model daje lepszą prognozę, gdyby cena tego samochodu wynosiła 1650? Jak możemy to wyjaśnić?

```
##        fit      lwr      upr
## 1 16484.18 11243.94 21724.42

##        fit      lwr      upr
## 1 18753.83 10437.85 27069.81

## [1] 0.8038145

## [1] 0.7168977
```

11 Regresja logistyczna i Poissona

11.1 Przykłady

Regresja logistyczna

Przykład. Rozważmy przykład dotyczący badania szansy ponownego ataku serca w ciągu roku od pierwszego ataku, w zależności od *treatment of anger* oraz *trait anxiety*. Zmienna zależna ma wartość 1, jeśli nastąpił ponowny atak, a 0 w przeciwnym razie.

```
y <- c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
x1 <- c(1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0)
x2 <- c(70, 80, 50, 60, 40, 65, 75, 80, 70, 60, 65, 50, 45, 35, 40, 50, 55, 45, 50, 60)
data_set <- data.frame(y, x1, x2)
head(data_set)
```

```
##   y x1 x2
## 1 1  1 70
## 2 1  1 80
## 3 1  1 50
## 4 1  0 60
## 5 1  0 40
## 6 1  0 65
```