Modele regresji i ich zastosowania
Labolatoria 4, 5, 6

Jan Solarz
243889

22 kwietnia 2021

Spis treści

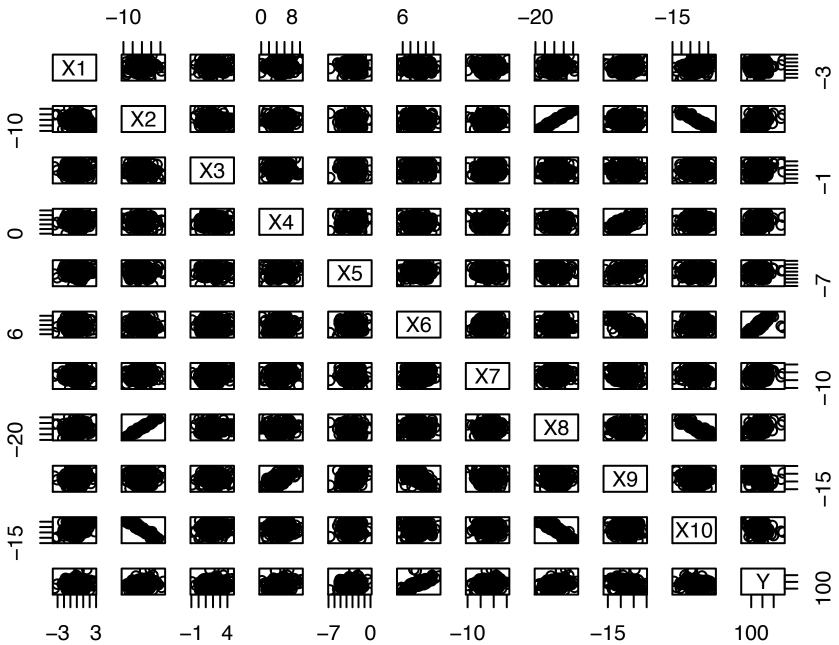
1 Zadania labolatoryjne

1.1 Zadanie 1

Analizy zaczynamy od zapoznania się z plikiem zawierającym 10 zmiennych objaśniających x i zmienną objaśnianą y w dwustu rekordach.

```
library(readxl)
regresja_wielokrotna <- read_excel("~/Downloads/regresja wielokrotna.xlsx")
mydata<-regresja_wielokrotna
attach(mydata)
```

```
pairs(cbind(mydata[1:11]))
```



- Najwiekszy wplyw na zmienna objasniana Y ma zmienna X6
- Silna wpoliniowosc zmiennych X2 z X8 i X10 , X8 z X10. czyli wszystkie korelacje miedzy soba w X2,X8,X10
- Tak pojawiaja sie wartosci odstajace w wykresach rozrzutów

1.2 Zadanie 2

```
cor(mydata[1:11])

##              X1           X2          X3          X4           X5          X6
## X1   1.00000000 -0.092964842  0.01819041  0.02498834  0.045798284 -0.04665757
## X2  -0.09296484  1.000000000 -0.11206291 -0.01226583 -0.028915615 -0.02372916
## X3   0.01819041 -0.112062905  1.00000000 -0.09353850 -0.015292795  0.04271405
## X4   0.02498834 -0.012265830 -0.09353850  1.00000000  0.140443718 -0.03405524
## X5   0.04579828 -0.028915615 -0.01529280  0.14044372  1.000000000  0.14840618
## X6  -0.04665757 -0.023729162  0.04271405 -0.03405524  0.148406183  1.00000000
## X7   0.01025066  0.028959963  0.14455031  0.09001609 -0.009798832  0.17388272
## X8   0.04512924  0.981162302  0.02386976 -0.02110227 -0.021747994 -0.02194675
## X9   0.05449456 -0.004255826 -0.09110212  0.69754423  0.331105098 -0.64710695
## X10  0.33618392 -0.961243387  0.09128925  0.01378357  0.041441572  0.02845195
## Y    0.02225308  0.291561058  0.11410456  0.24682540  0.167920613  0.80744019
##               X7           X8           X9          X10          Y
## X1   0.010250661  0.045129241  0.054494558  0.336183918 0.02225308
## X2   0.028959963  0.981162302 -0.004255826 -0.961243387 0.29156106
## X3   0.144550307  0.023869761 -0.091102124  0.091289247 0.11410456
```

```
## X4   0.090016089 -0.021102272  0.697544232  0.013783566  0.24682540
## X5  -0.009798832 -0.021747994  0.331105098  0.041441572  0.16792061
## X6   0.173882718 -0.021946755 -0.647106950  0.028451953  0.80744019
## X7   1.000000000  0.053673426 -0.060399680 -0.031137755  0.19726935
## X8   0.053673426  1.000000000 -0.009746542 -0.911924910  0.31485109
## X9  -0.060399680 -0.009746542  1.000000000  0.002997373 -0.33526500
## X10 -0.031137755 -0.911924910  0.002997373  1.000000000 -0.25865963
## Y    0.197269351  0.314851087 -0.335264997 -0.258659626  1.00000000
```

- X1 korelacja 0.33 z X10
- X2 korelacja 0.98 z X8, -0.96 z X10, 0.29 z Y
- X4 korelacja 0.69 z X9
- X5 korelacja 0.33 z X9
- X6 korelacja -0.64 z X9, 0.8 z Y
- X8 korelacja -0.91 z X10
- X9 korelacja -0.33 z Y
- Najwiekszy wplyw na Y ma X6. Duzo nizsza korelacje zauwazamy w X8, X9
- Wysoka wspoliniowosc wystepuje w parach (X2,X8), (X2,X10), (X8,X10), mniejsza w (X6,X9) i (X4,X9)

1.3 Zadanie 3

Na początek budujemy model regresji liniowej korzystając ze wszystkich zmiennych objaśniających

```
model <- lm( Y ~  X1+X2+X3+X4+X5+X6+X7+X8+X9+X10, data=mydata)
model.opis <- summary(model)
model.opis

##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 +
##     X10, data = mydata)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.810 -1.972 -1.048  0.070 97.059
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.52260    6.65182   0.079   0.9375
## X1           2.84868    4.02874   0.707   0.4804
```

```
## X2           1.82515    7.20785   0.253   0.8004
## X3           3.64880    3.61818   1.008   0.3145
## X4           3.95372    2.37698   1.663   0.0979 .
## X5           0.21928    2.46797   0.089   0.9293
## X6          11.00584    2.38215   4.620 7.08e-06 ***
## X7          -0.03279    0.27664  -0.119   0.9058
## X8          -0.14515    3.59911  -0.040   0.9679
## X9           0.13848    2.34504   0.059   0.9530
## X10         -0.73124    1.50367  -0.486   0.6273
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.14 on 189 degrees of freedom
## Multiple R-squared:  0.8534,Adjusted R-squared:  0.8456
## F-statistic:   110 on 10 and 189 DF,  p-value: < 2.2e-16
```

```
beta_0<-model.opis$coefficients[1]
beta_1<-model.opis$coefficients[2]
beta_2<-model.opis$coefficients[3]
beta_3<-model.opis$coefficients[4]
beta_4<-model.opis$coefficients[5]
beta_5<-model.opis$coefficients[6]
beta_6<-model.opis$coefficients[7]
beta_7<-model.opis$coefficients[8]
beta_8<-model.opis$coefficients[9]
beta_9<-model.opis$coefficients[10]
beta_10<-model.opis$coefficients[11]

cbind(beta_0,beta_1,beta_2,beta_3,beta_4,beta_5,beta_6,beta_7,beta_8,beta_9,beta_10)
```

```
##         beta_0   beta_1   beta_2   beta_3   beta_4    beta_5   beta_6
## [1,] 0.5226044 2.848677 1.825147 3.648797 3.953715 0.2192809 11.00584
##           beta_7     beta_8    beta_9    beta_10
## [1,] -0.03279499 -0.1451457 0.1384821 -0.7312405
```

```
model.opis$r.squared
```

```
## [1] 0.8533544
```

```
model.opis$adj.r.squared
```

```
## [1] 0.8455954
```

- Dzięki funkcji *lm* poznaliśmy estymatory najmniejszych kwadratów
- Zgodnie z podejrzeniami liniowy wpływ na zmienną Y ma zmienna *X6*, parametr p-value jest na bardzo niskim poziomie 7.08e-06
- parameter R.squared wynosi 0.8533544 a Adj.r.squared 0.8455954

1.4 Zadanie 4

Problem współniniowości

```
library(caret)

## Loading required package: lattice
## Loading required package: ggplot2

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1
--
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts()
--
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()

car::vif(model)

##          X1          X2          X3          X4          X5          X6
##   35.145297 1497.679681   27.256636   36.089952   11.758465   42.033020
##          X7          X8          X9         X10
##    1.093435 1469.489960   89.575184   73.671270

# Build the model
model1 <- lm(Y ~., data = mydata)
# Make predictions
predictions <- model1 %>% predict(mydata)
# Model performance
data.frame(
  RMSE = RMSE(predictions, mydata$Y),
  R2 = R2(predictions, mydata$Y)
)

##       RMSE        R2
## 1 9.861228 0.8533544

car::vif(model1)

##          X1          X2          X3          X4          X5          X6
##   35.145297 1497.679681   27.256636   36.089952   11.758465   42.033020
##          X7          X8          X9         X10
##    1.093435 1469.489960   89.575184   73.671270
```

```r
model2 <- lm(Y ~. -X2, data = mydata)
# Make predictions
predictions <- model2 %>% predict(mydata)
# Model performance
data.frame(
  RMSE = RMSE(predictions, mydata$Y),
  R2 = R2(predictions, mydata$Y)
)

##     RMSE        R2
## 1 9.8629 0.8533046

car::vif(model2)

##        X1        X3        X4        X5        X6        X7        X8        X9
## 11.331815  1.852655 35.844845 11.605348 41.780951  1.074035 64.180276 88.932719
##       X10
## 72.840327

model3 <- lm(Y ~. -X2-X9, data = mydata)
# Make predictions
predictions <- model3 %>% predict(mydata)
# Model performance
data.frame(
  RMSE = RMSE(predictions, mydata$Y),
  R2 = R2(predictions, mydata$Y)
)

##      RMSE        R2
## 1 9.86307 0.8532996

car::vif(model3)

##        X1        X3        X4        X5        X6        X7        X8       X10
## 11.276908  1.842674  1.048843  1.051652  1.098026  1.072346 63.522539 72.147455

model4 <- lm(Y ~. -X2-X9-X10, data = mydata)
# Make predictions
predictions <- model4 %>% predict(mydata)
# Model performance
data.frame(
  RMSE = RMSE(predictions, mydata$Y),
  R2 = R2(predictions, mydata$Y)
)

##       RMSE        R2
## 1 9.870298 0.8530845

car::vif(model4)
```

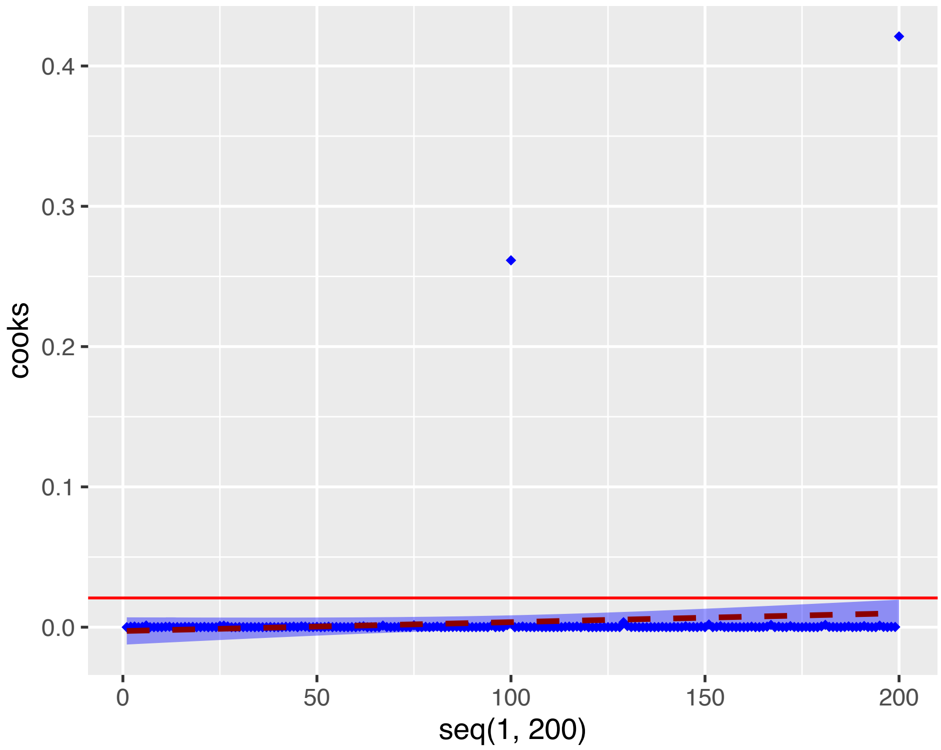
```
##        X1       X3       X4       X5       X6       X7       X8
## 1.008049 1.034192 1.047199 1.050985 1.065881 1.071434 1.007002

summary(model4)

##
## Call:
## lm(formula = Y ~ . - X2 - X9 - X10, data = mydata)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.072 -1.896 -1.096  0.045 97.594
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.09849    5.70330   0.368 0.713321
## X1           0.87524    0.67757   1.292 0.198003
## X3           2.44180    0.69990   3.489 0.000602 ***
## X4           4.10112    0.40209  10.199  < 2e-16 ***
## X5           0.32707    0.73273   0.446 0.655827
## X6          10.82853    0.37671  28.745  < 2e-16 ***
## X7          -0.03704    0.27195  -0.136 0.891801
## X8           1.13078    0.09356  12.086  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.07 on 192 degrees of freedom
## Multiple R-squared:  0.8531,Adjusted R-squared:  0.8477
## F-statistic: 159.3 on 7 and 192 DF,  p-value: < 2.2e-16

cor(predictions,Y)

## [1] 0.9236257
```

- Wyznaczamy wskaźnik podbicia wariancji dla każdej ze zmiennych objaśniających. Zmierzamy do tego aby VIF dla każdej ze zmiennncy był mniejszy od 10. Zaczynamy od modelu ze wszystkimi atrybutami, odrzucając najpierw ten z najwyższym VIF
- Odrzucamy najpierw X2- VIF 1492.67,następnie X9 i X10.
- Końcowy *model4* posiada R.squared o wartości 0.8531. Zauważamy tu silną liniowość na Y oprócz X6 również X3, X4 i X8.

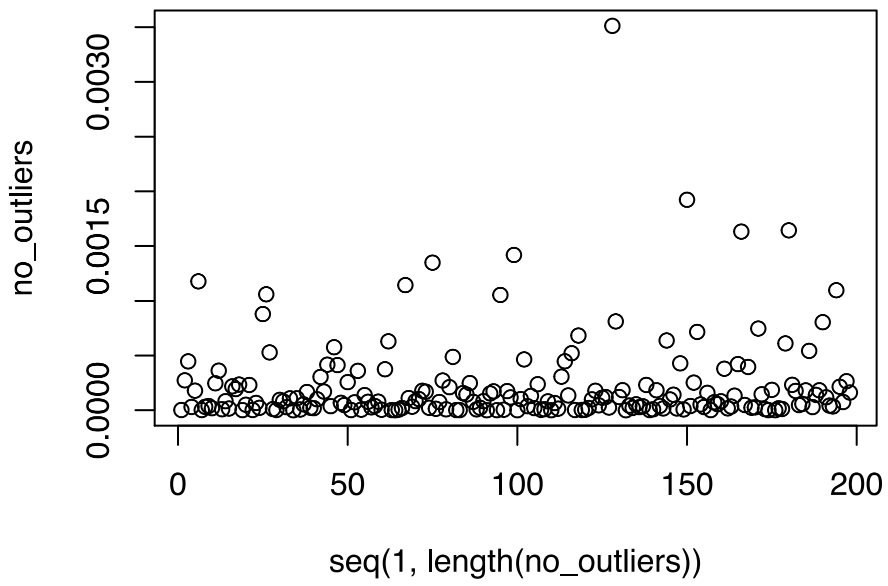
1.5 Zadanie 5

Usunięcie wartości wpływowych

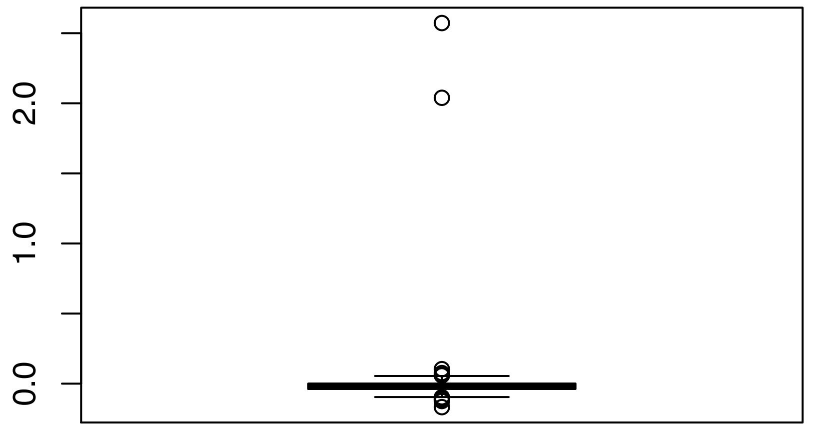
```r
library(base)
p=8
n=200
cooks<-cooks.distance(model4)
sort(cooks)[190:200]

##         195         67          6         75         99        167
## 0.001096444 0.001143393 0.001177646 0.001348909 0.001418963 0.001632445
##         181        151        129        100        200
## 0.001643652 0.001923286 0.003511861 0.261475614 0.421056076

ggplot(mydata, aes(x =seq(1,200), y = cooks)) +
  geom_point(shape=18, color="blue")+
  geom_hline(yintercept = 4/(n-p),col="red")+
  geom_smooth(method=lm,  linetype="dashed",
              color="darkred", fill="blue")

## `geom_smooth()` using formula 'y ~ x'
```




```r
no_outliers <- cooks[cooks< (4 /(n-p))]
length(no_outliers)

## [1] 198

plot(seq(1,length(no_outliers)),no_outliers)
```




```
dffits <- as.data.frame(dffits(model4))
boxplot(dffits)
```




```
y=4/(n-p)

library(MASS)
```

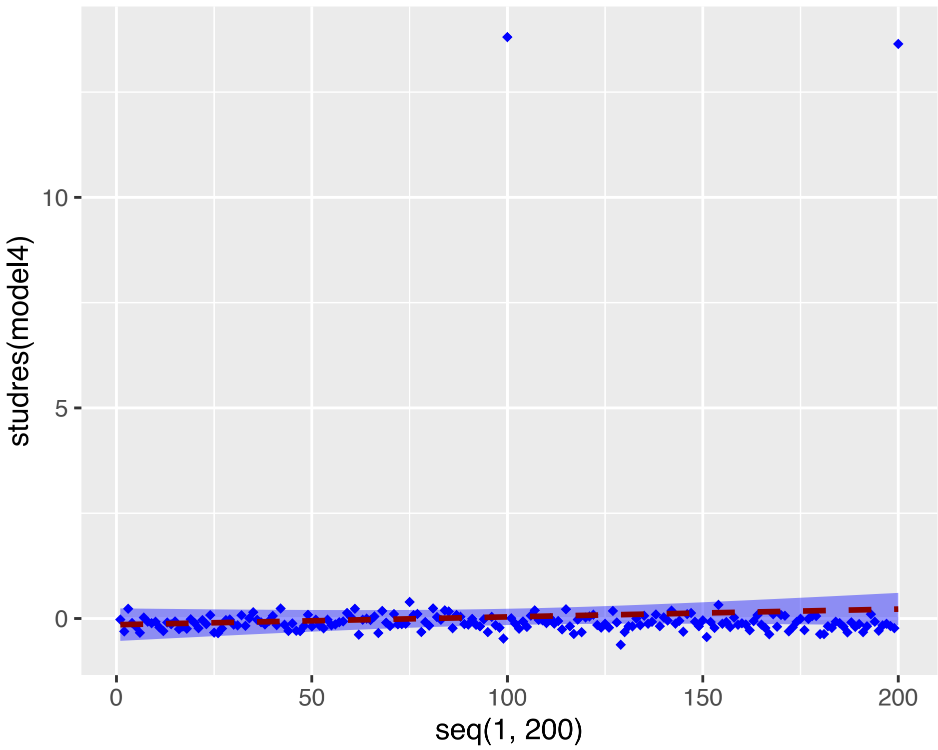
```
##
## Attaching package: 'MASS'
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
sort(studres(model4))[190:200]
```

```
##       107        84       115         3        61        42        81
##  0.1929974  0.1937862  0.2190568  0.2312174  0.2312531  0.2371080  0.2396129
##       154        75       200       100
##  0.3242153  0.3920606 13.6434551 13.8062864
```

```r
ggplot(mydata, aes(x =seq(1,200), y = studres(model4))) +
  geom_point(shape=18, color="blue")+
  geom_smooth(method=lm,  linetype="dashed",
              color="darkred", fill="blue")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



- Po zbadaniu wpływu kolejnych obserwacji za pomocą odległości Cooka widzimy że obserwacje z indeksami 100 i 200 znacznie odbiegają wartościami od reszty. Zauważamy to również na wykresach.
- Korzystając ze studentyzowanych reziduów i DFFITS dochodzimu do tych samych wniosków, wartości zbyt wpływowe, odbiegające od reszty należa do prób 100 i 200

1.6 Zadanie 6

```r
mydata_2<-mydata[-c(100,200),]


model_2 <- lm( Y ~  ., data=mydata_2)
model.opis <- summary(model_2)
model.opis

##
## Call:
## lm(formula = Y ~ ., data = mydata_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68735 -0.21365 -0.00748  0.16698  1.01993
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.240284   0.191867   1.252   0.2120
## X1           0.785045   0.116366   6.746 1.84e-10 ***
## X2           1.471536   0.207702   7.085 2.75e-11 ***
## X3           2.741074   0.104403  26.255  < 2e-16 ***
## X4           4.064837   0.068492  59.348  < 2e-16 ***
## X5           0.112932   0.071147   1.587   0.1141
## X6          10.923599   0.068654 159.110  < 2e-16 ***
## X7           0.008906   0.007977   1.116   0.2657
## X8           0.254224   0.103778   2.450   0.0152 *
## X9          -0.067554   0.067595  -0.999   0.3189
## X10         -0.014412   0.043571  -0.331   0.7412
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2923 on 187 degrees of freedom
## Multiple R-squared:  0.9999,Adjusted R-squared:  0.9998
## F-statistic: 1.303e+05 on 10 and 187 DF,  p-value: < 2.2e-16

beta_0<-model.opis$coefficients[1]
beta_1<-model.opis$coefficients[2]
beta_2<-model.opis$coefficients[3]
beta_3<-model.opis$coefficients[4]
beta_4<-model.opis$coefficients[5]
beta_5<-model.opis$coefficients[6]
beta_6<-model.opis$coefficients[7]
beta_7<-model.opis$coefficients[8]
beta_8<-model.opis$coefficients[9]
beta_9<-model.opis$coefficients[10]
beta_10<-model.opis$coefficients[11]

cbind(beta_0,beta_1,beta_2,beta_3,beta_4,beta_5,beta_6,beta_7,beta_8,beta_9,beta_10)
```

```
##         beta_0    beta_1   beta_2   beta_3   beta_4    beta_5  beta_6
## [1,] 0.240284 0.7850445 1.471536 2.741074 4.064837 0.1129322 10.9236
##            beta_7    beta_8      beta_9     beta_10
## [1,] 0.008906206 0.2542243 -0.06755406 -0.01441161
```

```
model.opis$r.squared
```

```
## [1] 0.9998565
```

```
model.opis$adj.r.squared
```

```
## [1] 0.9998488
```

- Bardzo wysokie wskazniki R2 i R adjusted co wskazuje za niemal 100 procentowe pokrywanie wartosci objasnianych przez model. Odpowiednio 0.9998565 i 0.9998488
- Usuniecie wartosci odstajacych zdecydowanie polepszylo dopasowanie modelu
- Zmienne X1,X2,X3,X4 i X6 *** maja p-value bliskie zeru - liniowosc wzgledem zmiennej objasnianej

1.7 Zadanie 7

Opcja *forward*

```
#define intercept-only model
intercept_only <- lm(Y ~ 1, data=mydata_2)

#define model with all predictors
all <- lm(Y ~ ., data=mydata_2)

#perform forward stepwise regression
forward <- step(intercept_only, direction='forward', scope=formula(all), trace=0)

#view results of forward stepwise regression
forward$anova
```

```
##   Step Df     Deviance Resid. Df   Resid. Dev       AIC
## 1      NA           NA       197 111299.36644 1255.6789
## 2 + X6 -1 8.953010e+04       196  21769.26205  934.5975
## 3 + X8 -1 1.099115e+04       195  10778.10764  797.4070
## 4 + X4 -1 9.909523e+03       194    868.58475  300.7624
## 5 + X3 -1 8.476970e+02       193     20.88778 -435.3223
## 6 + X1 -1 3.737500e-01       192     20.51403 -436.8973
## 7 + X2 -1 3.994453e+00       191     16.51958 -477.7767
## 8 + X5 -1 3.553794e-01       190     16.16420 -480.0827
```

```
#view final model
forward$coefficients
```

```
## (Intercept)          X6          X8          X4          X3          X1
##  0.21177139 10.99290122  0.28150187  3.99899680  2.71840557  0.74625250
##          X2          X5
##  1.43183894  0.04358071
```

```r
summary(forward)
```

```
##
## Call:
## lm(formula = Y ~ X6 + X8 + X4 + X3 + X1 + X2 + X5, data = mydata_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.70034 -0.20767 -0.01516  0.16683  1.00971
##
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  0.21177    0.15731    1.346   0.1798
## X6          10.99290    0.01077 1020.906  < 2e-16 ***
## X8           0.28150    0.10169    2.768   0.0062 **
## X4           3.99900    0.01161  344.585  < 2e-16 ***
## X3           2.71841    0.10297   26.399  < 2e-16 ***
## X1           0.74625    0.10235    7.291 8.06e-12 ***
## X2           1.43184    0.20363    7.032 3.58e-11 ***
## X5           0.04358    0.02132    2.044   0.0423 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2917 on 190 degrees of freedom
## Multiple R-squared:  0.9999,Adjusted R-squared:  0.9998
## F-statistic: 1.869e+05 on 7 and 190 DF,  p-value: < 2.2e-16
```

```r
summary(forward)$r.squared
```

```
## [1] 0.9998548
```

```r
summary(forward)$adj.r.squared
```

```
## [1] 0.9998494
```

Opcja *backward*

```r
#define intercept-only model
intercept_only <- lm(Y ~ 1, data=mydata_2)

#define model with all predictors
all <- lm(Y ~ ., data=my_data2)
```

```
## Error in is.data.frame(data): nie znaleziono obiektu 'my_data2'
```

```r
#perform backward stepwise regression
backward <- step(all, direction='backward', scope=formula(all), trace=0)

#view results of backward stepwise regression
backward$anova

##    Step Df    Deviance Resid. Df Resid. Dev       AIC
## 1       NA          NA       187   15.97401 -476.4262
## 2 - X10  1 0.009345701       188   15.98335 -478.3104
## 3  - X9  1 0.080975588       189   16.06433 -479.3098
## 4  - X7  1 0.099867386       190   16.16420 -480.0827

#view final model
backward$coefficients

## (Intercept)          X1          X2          X3          X4          X5
##  0.21177139  0.74625250  1.43183894  2.71840557  3.99899680  0.04358071
##          X6          X8
## 10.99290122  0.28150187

summary(backward)

##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X8, data = mydata_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.70034 -0.20767 -0.01516  0.16683  1.00971
##
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  0.21177    0.15731    1.346   0.1798
## X1           0.74625    0.10235    7.291 8.06e-12 ***
## X2           1.43184    0.20363    7.032 3.58e-11 ***
## X3           2.71841    0.10297   26.399  < 2e-16 ***
## X4           3.99900    0.01161  344.585  < 2e-16 ***
## X5           0.04358    0.02132    2.044   0.0423 *
## X6          10.99290    0.01077 1020.906  < 2e-16 ***
## X8           0.28150    0.10169    2.768   0.0062 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2917 on 190 degrees of freedom
## Multiple R-squared:  0.9999,Adjusted R-squared:  0.9998
## F-statistic: 1.869e+05 on 7 and 190 DF,  p-value: < 2.2e-16

summary(backward)$r.squared
```

```
## [1] 0.9998548
```

```r
summary(backward)$adj.r.squared
```

```
## [1] 0.9998494
```

```r
summary(all)
```

```
##
## Call:
## lm(formula = Y ~ ., data = mydata_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68735 -0.21365 -0.00748  0.16698  1.01993
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.240284   0.191867   1.252   0.2120
## X1           0.785045   0.116366   6.746 1.84e-10 ***
## X2           1.471536   0.207702   7.085 2.75e-11 ***
## X3           2.741074   0.104403  26.255  < 2e-16 ***
## X4           4.064837   0.068492  59.348  < 2e-16 ***
## X5           0.112932   0.071147   1.587   0.1141
## X6          10.923599   0.068654 159.110  < 2e-16 ***
## X7           0.008906   0.007977   1.116   0.2657
## X8           0.254224   0.103778   2.450   0.0152 *
## X9          -0.067554   0.067595  -0.999   0.3189
## X10         -0.014412   0.043571  -0.331   0.7412
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2923 on 187 degrees of freedom
## Multiple R-squared:  0.9999,Adjusted R-squared:  0.9998
## F-statistic: 1.303e+05 on 10 and 187 DF,  p-value: < 2.2e-16
```

```r
predict(forward, newdata=mydata_2, interval="confidence")[1:10,]
```

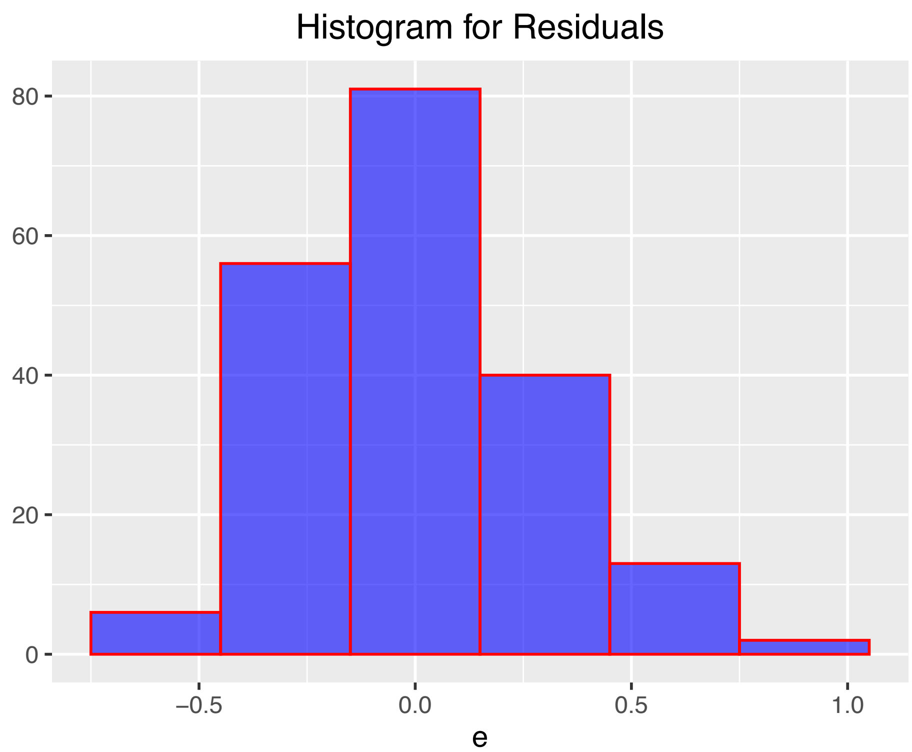
```
##         fit      lwr      upr
## 1  140.1615 140.0534 140.2696
## 2  118.2700 118.1724 118.3676
## 3  133.8411 133.7020 133.9802
## 4  110.2973 110.1852 110.4094
## 5  151.8307 151.7435 151.9179
## 6  109.0819 108.9131 109.2508
## 7  148.5690 148.4626 148.6754
```

```
## 8  139.8768 139.7526 140.0011
## 9  120.1604 120.0713 120.2496
## 10 121.0857 120.9921 121.1794
```

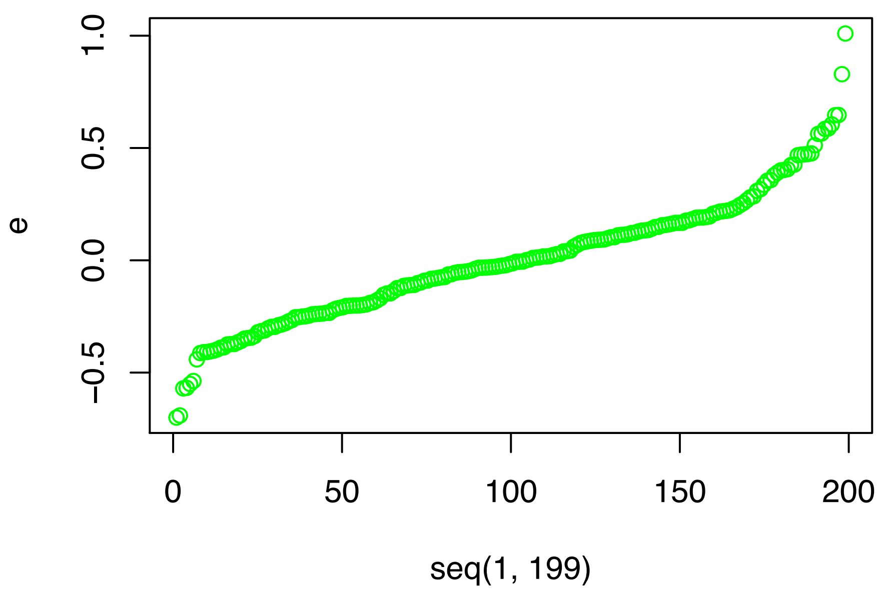
- Estymaotry wynoszą przy opcji forward 0.21177139 X6: 10.99290122 X8: 0.28150187 X4: 3.99899680 X3: 2.71840557 X1: 0.74625250 X2: 1.43183894 X5: 0.04358071. Przy opcji backward dokładnie tyle samo, zostały jednak dobrene w innej kolejności
- Liniowy wpływ na zmienna Y mają zmienne X1 o p-value 8.06e-12, X2 3.58e-11, X3 ¡ 2e-16, X4 ¡ 2e-16, X6 ¡ 2e-16
- Przy uwzglednieniu wszystkich zmiennych objaśniających liniowy wpływ równiez mają X1, X2, X3, X4 i X6. P-value całego testu wynosi 2.2e-16, jest bardzo niska, więc model ma jak najbardziej sens
- R.squared 0.9998548, Adj.r.squared 0.9998494

1.8 Zadanie 8

```r
e=mydata_2$Y-predict(forward,mydata_2)

par(mfrow=c(1,1))
qplot(e,
      binwidth = 0.3,
      main = "Histogram for Residuals",
      xlab = "e",
      fill=I("blue"),
      col=I("red"),
      alpha=I(.6))+
  theme(plot.title = element_text(hjust = 0.5))
```

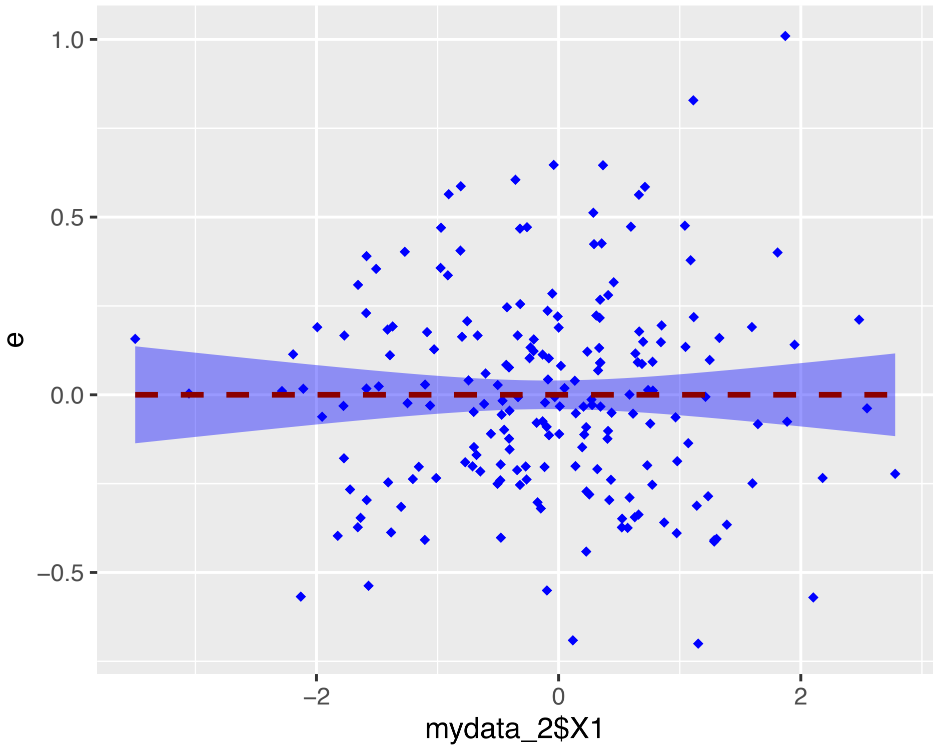



```
qqplot(seq(1,199),e, col="green")
```

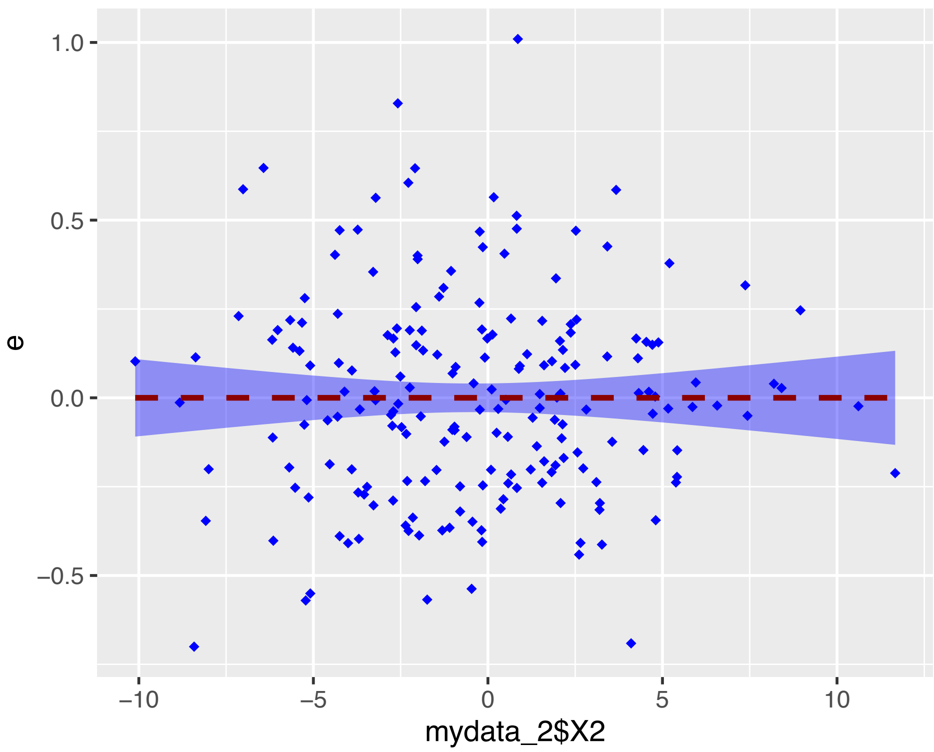



```
par(mfrow=c(3,3))
ggplot(mydata_2,aes(x = mydata_2$X1, y = e)) +
  geom_point(shape=18, color="blue")+
  geom_smooth(method=lm,  linetype="dashed",
```

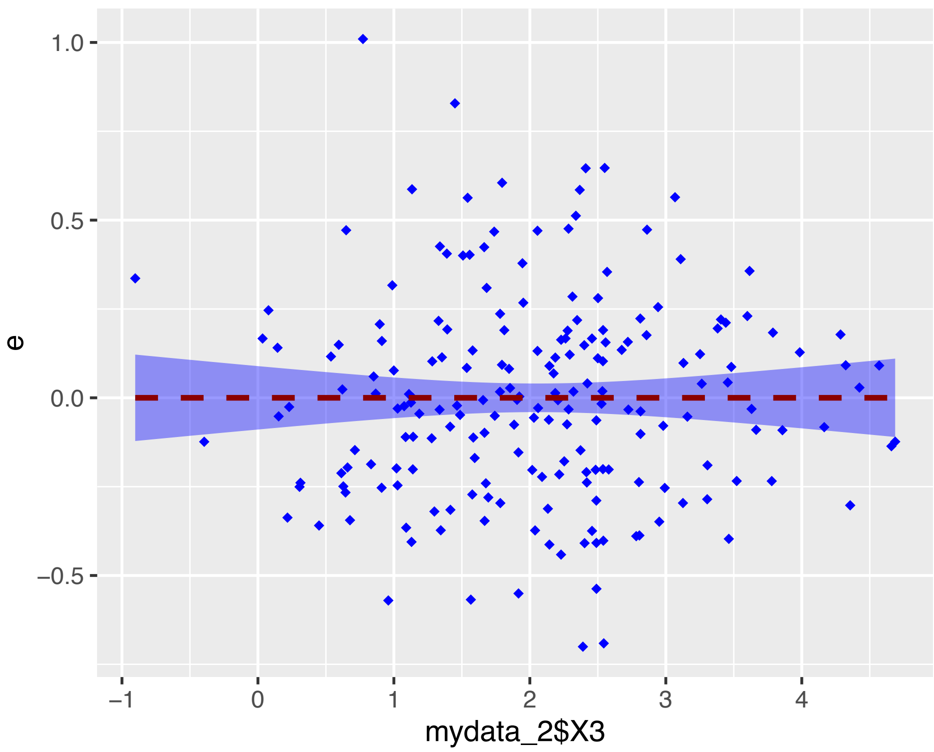
```
              color="darkred", fill="blue")

## Warning: Use of `mydata_2$X1` is discouraged. Use `X1` instead.
## Warning: Use of `mydata_2$X1` is discouraged. Use `X1` instead.
## `geom_smooth()` using formula 'y ~ x'
```



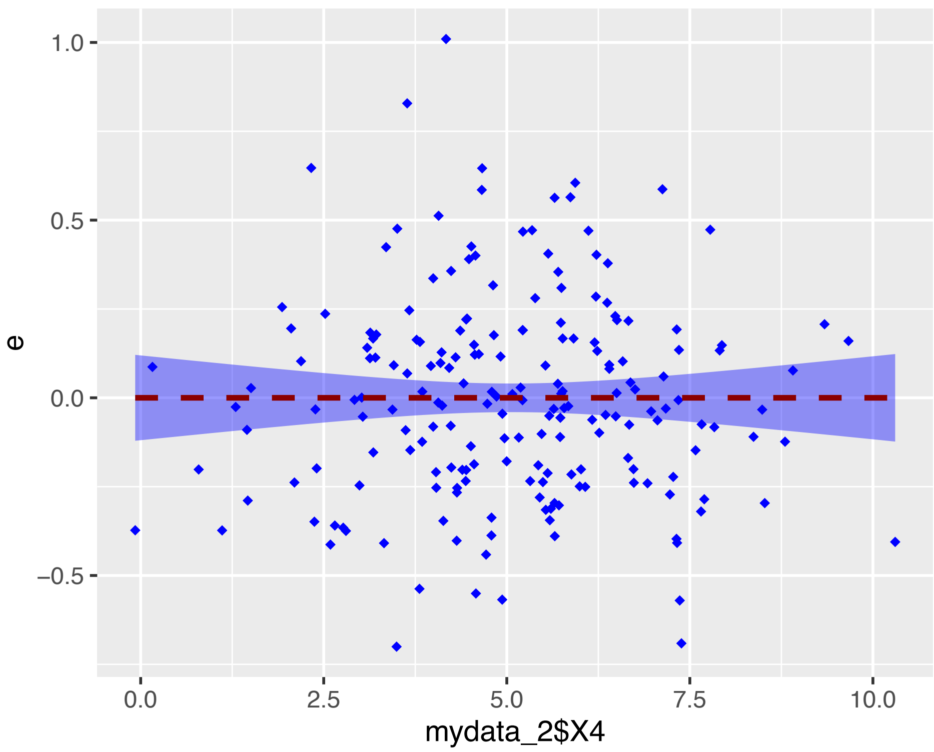

```
ggplot(mydata_2,aes(x = mydata_2$X2, y = e)) +
  geom_point(shape=18, color="blue")+
  geom_smooth(method=lm,  linetype="dashed",
              color="darkred", fill="blue")

## Warning: Use of `mydata_2$X2` is discouraged. Use `X2` instead.
## Warning: Use of `mydata_2$X2` is discouraged. Use `X2` instead.
## `geom_smooth()` using formula 'y ~ x'
```



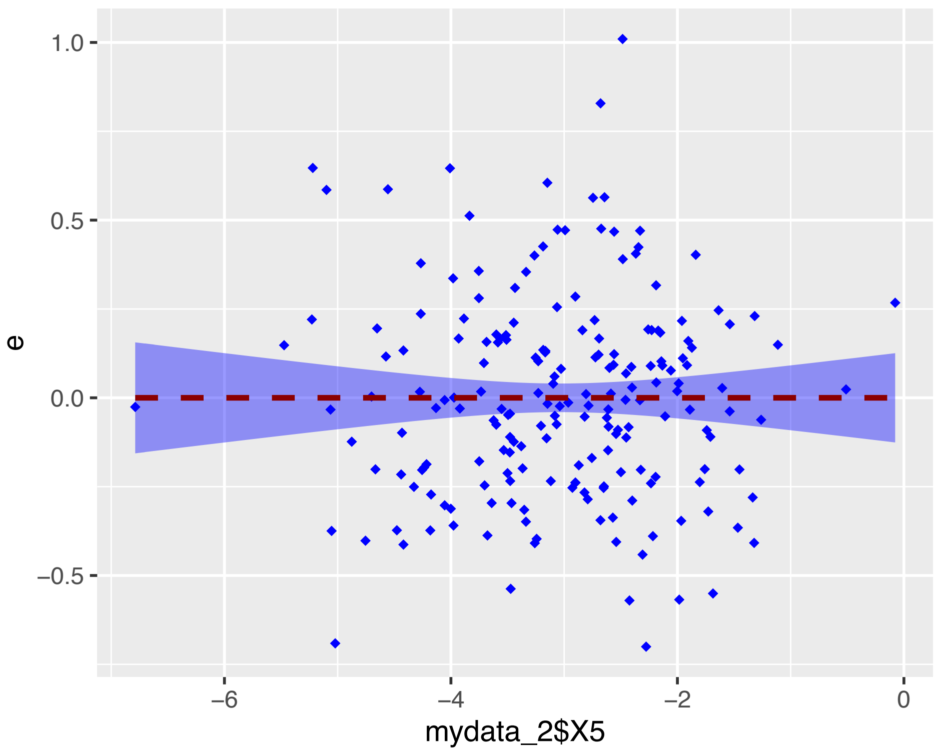

```
ggplot(mydata_2,aes(x = mydata_2$X3, y = e)) +
  geom_point(shape=18, color="blue")+
  geom_smooth(method=lm,  linetype="dashed",
              color="darkred", fill="blue")

## Warning: Use of `mydata_2$X3` is discouraged. Use `X3` instead.
## Warning: Use of `mydata_2$X3` is discouraged. Use `X3` instead.
## `geom_smooth()` using formula 'y ~ x'
```



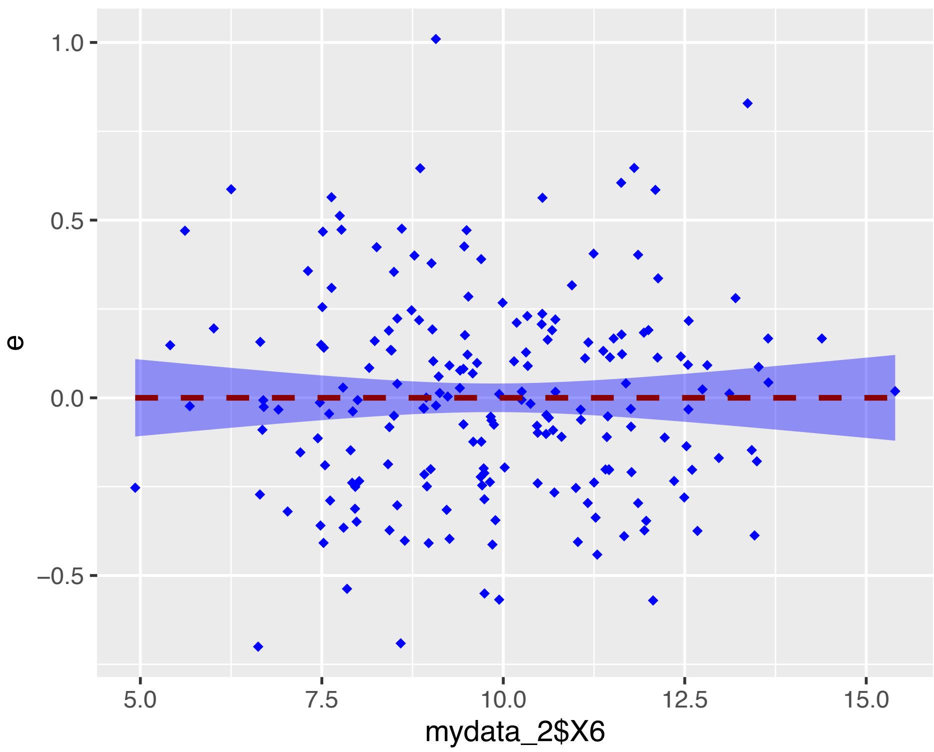

```
ggplot(mydata_2,aes(x = mydata_2$X4, y = e)) +
  geom_point(shape=18, color="blue")+
  geom_smooth(method=lm,  linetype="dashed",
              color="darkred", fill="blue")

## Warning: Use of `mydata_2$X4` is discouraged. Use `X4` instead.
## Warning: Use of `mydata_2$X4` is discouraged. Use `X4` instead.
## `geom_smooth()` using formula 'y ~ x'
```



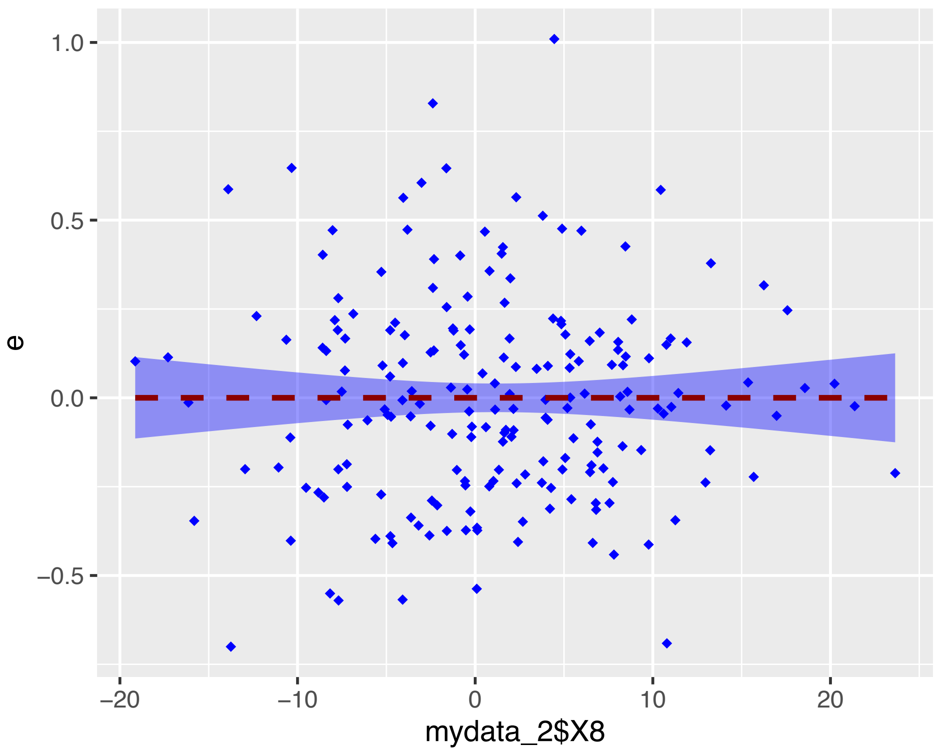

```
ggplot(mydata_2,aes(x = mydata_2$X5, y = e)) +
  geom_point(shape=18, color="blue")+
  geom_smooth(method=lm,  linetype="dashed",
              color="darkred", fill="blue")

## Warning: Use of `mydata_2$X5` is discouraged. Use `X5` instead.
## Warning: Use of `mydata_2$X5` is discouraged. Use `X5` instead.
## `geom_smooth()` using formula 'y ~ x'
```



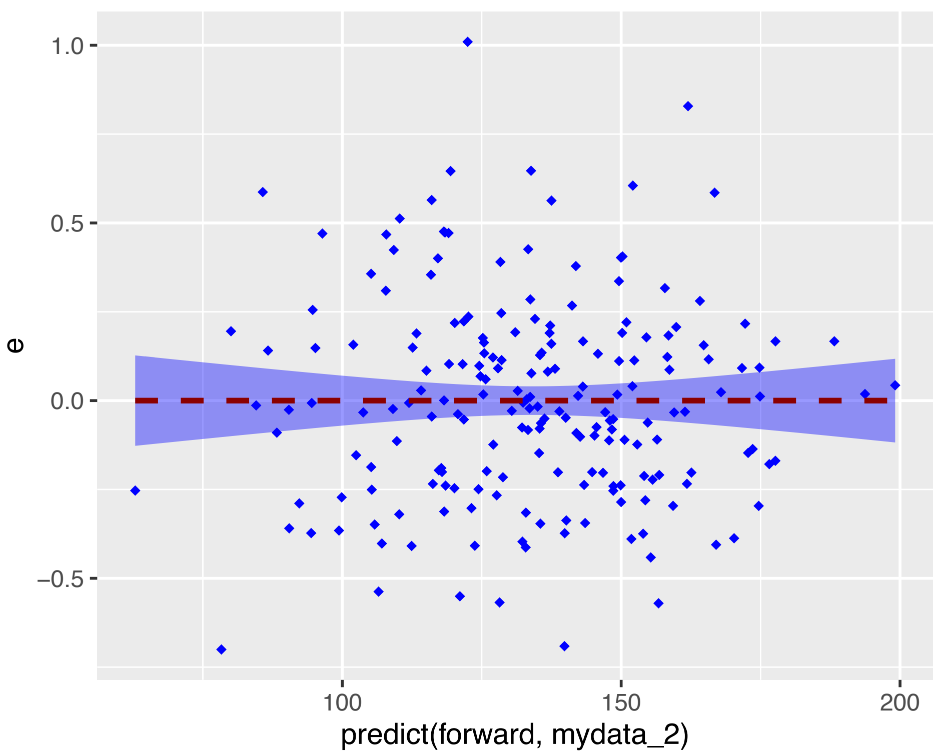

```
ggplot(mydata_2,aes(x = mydata_2$X6, y = e)) +
  geom_point(shape=18, color="blue")+
  geom_smooth(method=lm,  linetype="dashed",
              color="darkred", fill="blue")

## Warning: Use of `mydata_2$X6` is discouraged. Use `X6` instead.
## Warning: Use of `mydata_2$X6` is discouraged. Use `X6` instead.
## `geom_smooth()` using formula 'y ~ x'
```




```
ggplot(mydata_2,aes(x = mydata_2$X8, y = e)) +
  geom_point(shape=18, color="blue")+
  geom_smooth(method=lm,  linetype="dashed",
              color="darkred", fill="blue")

## Warning: Use of `mydata_2$X8` is discouraged. Use `X8` instead.
## Warning: Use of `mydata_2$X8` is discouraged. Use `X8` instead.
## `geom_smooth()` using formula 'y ~ x'
```




```
par(mfrow=c(1,1))
ggplot(mydata_2,aes(x = predict(forward,mydata_2), y = e)) +
  geom_point(shape=18, color="blue")+
  geom_smooth(method=lm,  linetype="dashed",
              color="darkred", fill="blue")

## `geom_smooth()` using formula 'y ~ x'
```



- Wartości residuów sa bliskie zeru, oscylują między -1,1 potwierdzają poprawność modelu
- Nie ma korelacji między zmiennymi objaśniającymi i e oraz między wartościami prognozowaymi i e, korelacja jest równa prawie zeru, stąd model regresji liniowej poprawnie opisuje zależność między zmiennymi.

1.9 Zadanie 9

```
summary(forward)

##
## Call:
## lm(formula = Y ~ X6 + X8 + X4 + X3 + X1 + X2 + X5, data = mydata_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.70034 -0.20767 -0.01516  0.16683  1.00971
##
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  0.21177    0.15731    1.346   0.1798
## X6          10.99290    0.01077 1020.906  < 2e-16 ***
## X8           0.28150    0.10169    2.768   0.0062 **
## X4           3.99900    0.01161  344.585  < 2e-16 ***
## X3           2.71841    0.10297   26.399  < 2e-16 ***
## X1           0.74625    0.10235    7.291 8.06e-12 ***
## X2           1.43184    0.20363    7.032 3.58e-11 ***
## X5           0.04358    0.02132    2.044   0.0423 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2917 on 190 degrees of freedom
## Multiple R-squared:  0.9999,Adjusted R-squared:  0.9998
## F-statistic: 1.869e+05 on 7 and 190 DF,  p-value: < 2.2e-16

predict(forward, c(1,2,3,4,5,6,7))

## Error in eval(predvars, data, env): liczbowy argument 'envir' nie posiada długości
1

0.21177139 + 6* 10.99290122  +8* 0.28150187 + 4* 3.99899680 + 3* 2.71840557 + 1* 0.74625

## [1] 96.40023
```

Wartość prognozowana wynosi 96.4

2 Zadania teoretyczne

2.1

Cel: 1) $P := I_n - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T$ jest macierzą symetryczną czyli $P = P^T$

2) $P^2 = P$

1)
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \vdots \\ 0 & \vdots & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} - \frac{1}{n}\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}(1, 1, \dots, 1) =$$

$$= \begin{pmatrix} 1 & 0 & & \\ 0 & 1 & & 0 \\ & & \ddots & \\ & & 0 & 1 \end{pmatrix} - \frac{1}{n}\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & & & \vdots \\ \vdots & & 1 & 1 \\ 1 & \cdots & & 1 \end{pmatrix} = \begin{pmatrix} 1-\frac{1}{n} & -\frac{1}{n} & \cdots & -\frac{1}{n} \\ -\frac{1}{n} & 1-\frac{1}{n} & & \\ \vdots & & \ddots & \\ -\frac{1}{n} & & & 1-\frac{1}{n} \end{pmatrix} := P$$

$$P^T = \begin{pmatrix} 1-\frac{1}{n} & \cdots & -\frac{1}{n} \\ -\frac{1}{n} & 1-\frac{1}{n} & \vdots \\ \vdots & & 1-\frac{1}{n} \\ -\frac{1}{n} & & \end{pmatrix} = P$$, czyli symetryczna

2)
$$\begin{array}{ccc} & n\times n & n\times n \\ P^2 = P \cdot P = & \begin{pmatrix} 1-\frac{1}{n} & \cdots & -\frac{1}{n} \\ -\frac{1}{n} & \ddots & \vdots \\ \vdots & & 1-\frac{1}{n} \\ -\frac{1}{n} & \cdots & \end{pmatrix} & \begin{pmatrix} 1-\frac{1}{n} & \cdots & -\frac{1}{n} \\ -\frac{1}{n} & \ddots & \vdots \\ \vdots & & \\ -\frac{1}{n} & \cdots & 1-\frac{1}{n} \end{pmatrix} = \end{array}$$

$$= \begin{pmatrix} \left(1-\frac{1}{n}\right)^2 + (n-1)\left(-\frac{1}{n}\right)^2 & \cdots & 2\left(1-\frac{1}{n}\right)\left(-\frac{1}{n}\right) + (n-2)\left(-\frac{1}{n}\right)^2 \\ 2\left(1-\frac{1}{n}\right)\left(-\frac{1}{n}\right) + (n-2)\left(-\frac{1}{n}\right)^2 & \ddots & \vdots \\ \text{---}\ \ ||\ \ \text{---} & & \\ \vdots & & \left(1-\frac{1}{n}\right)^2 + (n-1)\left(-\frac{1}{n}\right)^2 \\ \text{---}\ \ ||\ \ \text{---} & & \end{pmatrix} =$$

$$= \left\{ \begin{aligned} &\left(1-\frac{1}{n}\right)^2 + (n-1)\left(-\frac{1}{n}\right)^2 = 1 - \frac{2}{n} + \frac{1}{n^2} + \frac{1}{n^2} - \frac{1}{n^2} = 1 - \frac{1}{n} \\ &2\left(1-\frac{1}{n}\right)\left(-\frac{1}{n}\right) + (n-2)\left(-\frac{1}{n}\right)^2 = -\frac{2}{n} + \frac{2}{n^2} + \frac{1}{n} - \frac{2}{n^2} = -\frac{1}{n} \end{aligned} \right\} = \begin{pmatrix} 1-\frac{1}{n} & \cdots & -\frac{1}{n} \\ -\frac{1}{n} & \ddots & \\ \vdots & & \\ -\frac{1}{n} & & 1-\frac{1}{n} \end{pmatrix} = P$$

z.3
zm. objaśniana $(y_1, \dots, y_n)$
zm. objaśniająca $(x_1, \dots, x_n)$

$$X = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}$$

$$X^T X = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \end{pmatrix} \cdot \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} = \begin{pmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{pmatrix}$$

$$H = X \cdot (X^T X)^{-1} \cdot X^T = \quad \Bigg| \quad (X^T X)^{-1} = \frac{1}{n\sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i\right)^2} \begin{pmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{pmatrix}$$

$$= \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \cdot \frac{1}{n\sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i\right)^2} \begin{pmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \end{pmatrix} =$$

$n \times 2 \qquad 2 \times n$

$$= \frac{1}{n\sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i\right)^2} \begin{pmatrix} \sum x_i^2 - x_1 \sum x_i & n x_1 - \sum_{i} x_i \\ \sum x_i^2 - x_2 \sum x_i & n x_2 - \sum x_i \\ \vdots & \vdots \\ \sum x_i^2 - x_n \sum x_i & n x_n - \sum x_i \end{pmatrix} \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \end{pmatrix} =$$

$$= \frac{1}{n\sum x_i^2 - \left(\sum x_i\right)^2} \begin{pmatrix} \sum x_i^2 - x_1 \sum x_i + x_1(n x_1 - \sum x_i) & \dots & \sum x_i^2 - x_1 \sum x_i + x_n(n x_1 - \sum x_i) \\ \sum x_i^2 - x_2 \sum x_i + x_1(n x_2 - \sum x_i) & \dots & \sum x_i^2 - x_2 \sum x_i + x_n(n x_2 - \sum x_i) \\ \vdots & & \vdots \\ \sum x_i^2 - x_n \sum x_i + x_1(n x_n - \sum x_i) & \dots & \sum x_i^2 - x_n \sum x_i + x_n(n x_n - \sum x_i) \end{pmatrix}$$