

## Testy diagnostyczne dla modelu liniowego

<https://cran.r-project.org/web/packages/lmtest/vignettes/lmtest-intro.pdf>

<https://cran.r-project.org/web/packages/strucchange/vignettes/strucchange-intro.pdf>

Estymacja i wnioskowanie z modelu regresji zależy od kilku założeń. Te założenia należy sprawdzić za pomocą diagnostyki regresji. Potencjalne problemy dzielimy na trzy kategorie:

- Błąd - przyjęliśmy, że  $\epsilon \sim N(0, \sigma^2 \mathbf{I})$  lub słownie, że błędy są niezależne, mają równą wariancję i mają rozkład normalny.
- Model - przyjęliśmy, że część strukturalna modelu  $E(\mathbf{Y}) = \mathbf{X}\beta$  jest poprawny.
- Nietypowe obserwacje- czasami tylko kilka obserwacji nie pasuje do modelu. Tych kilka obserwacji może zmienić wybór i dopasowanie modelu.

Techniki diagnostyczne mogą być **graficzne**, które są bardziej elastyczne, ale trudniejsze do ostatecznej interpretacji lub **liczbowe**, które mają węższy zakres, ale nie wymagają intuicji.

Pierwszy model, który wypróbujemy, może okazać się niewystarczający. Diagnostyka regresji często sugeruje ulepszenia, co oznacza, że **budowanie modelu jest często procesem iteracyjnym i interaktywnym**. Diagnostykę powtarza się w kolejnych modelach.

## Testy jednorodności wariancji (homoscedastyczności)

- Test Breuscha-Pagana jednorodności wariancji polega na weryfikacji hipotezy zerowej dotyczącej jednorodności wariancji reszt. Hipoteza alternatywna głosi, że wariancja reszt zależy od zmiennych objaśniających w modelu. Ponieważ średnia reszt  $\hat{\epsilon}$  wynosi 0 zamiast wariancji reszt bada się kwadraty reszt. Po wyznaczeniu reszt  $\hat{\epsilon}$  z modelu liniowego  $\mathbf{Y} = \mathbf{X}\beta + \epsilon$  rozważa się kolejny model liniowy  $\hat{\epsilon}^2 = \mathbf{X}\gamma + \delta$ . Test Breuscha-Pagana sprowadza się do sprawdzenia czy wszystkie wartości  $\gamma$  nie różnią się istotnie od 0.

```
library(openxlsx)
dane<-read.xlsx("C:/Users/User/Documents/Dydaktyka/Modele liniowe/Ćwiczenia/Samochody.xlsx")
model <- lm(Zużycie ~ Pojemność + Moc +Ładowność + Masa + Długość + Szerokość,dane)
require(lmtest)
bptest(model) #studentyzowany test homoscedastyczności Breuscha-Pagan
```

```
studentized Breusch-Pagan test
data: model
BP = 9.2816, df = 6, p-value = 0.1584
```

```
require(car)
ncvTest(model) #Non-constant Variance Score Test Breuscha Pagana
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
```

Chisquare = 2.163788, Df = 1, p = 0.1413

- Test Goldfelda-Quandt'a sprawdzający jednorodności wariancji polega na wykonaniu podziału reszt na dwie grupy i sprawdzeniu jednorodności wariancji w dwóch grupach. Podział na dwie grupy jest zdefiniowany przez argument `point` określający proporcje wielkości grup oraz argument `order.by` określający którą zmienną posłuży do podziału na grupy. Testowana jest więc hipoteza

$$H_0: \sigma^2_{[x[,order.by]<c]} = \sigma^2_{[x[,order.by]>c]}$$

gdzie  $c$  to kwantyl rzędu `point` dla zmiennej `x[,order.by]` a  $\sigma^2_{[x[,order.by]<c]}$  to wariancja reszt dla obserwacji dla których zmienna `x[,order.by]` przyjmuje wartości mniejsze niż  $c$ . Test Goldfelda-Quandt'a jest zaimplementowany w funkcji `gqtest()` w pakiecie `lmtest`.

```
require(lmtest)
gqtest(model,order.by=~Szerokość,point=0.5,data=dane)
```

Goldfeld-Quandt test

data: model

GQ = 4.486, df1 = 5, df2 = 5, p-value = 0.06256

alternative hypothesis: variance increases from segment 1 to 2

**Uwaga** `order.by=NULL` (default) powoduje że reszty są uporządkowane w kolejności występowania – traktowane są jak szereg czasowy

Jeśli użyjemy `attach(dane)`, to w wywołaniu testów `#order.by =~ Szerokość` można zastąpić przez `order.by = Szerokość`

- Test Harrisona-McCabe'a jest podobny do Goldfelda-Quandt'a, sprawdza tą samą hipotezę. Również badana jest zależność wariancji reszt dla obserwacji dla których zmienna wskazana argumentem `order.by` przyjmuje wartości mniejsze niż kwantyl rzędu `point`. Statystyką testową jest iloraz sumy kwadratów tych reszt do sumy kwadratów wszystkich reszt.

$$T_{HMC} = \frac{\sum_{i: x[i, order.by] < c} \hat{\epsilon}_i^2}{\sum_{i=1}^n \hat{\epsilon}_i^2}$$

Test jest zaimplementowany w funkcji `hmcetest()` w pakiecie `lmtest`. Opcja `plot=TRUE` wygeneruje rysunek przedstawiający wartość statystyki  $T_{HMC}$  w zależności od punktu odcięcia

```
hmcetest(model,order.by=~ Szerokość,point=0.5,data=dane,plot=TRUE)
```

Harrison-McCabe test

data: model

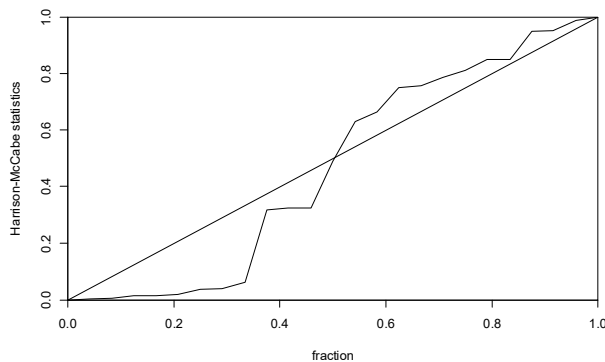
HMC = 0.48848, p-value = 0.473

```
hmcetest(model,order.by =~ Szerokość,point=0.3,data=dane,plot=TRUE)
```

Harrison-McCabe test

data: model

HMC = 0.037393, p-value = 0.001



## Testy niezależności reszt

- Test Durбина-Watsona niezależności reszt umożliwia weryfikację hipotezy o niezależności reszt jako alternatywy przez sprawdzenie, czy autokorelacja rzędu 1 reszt nie jest istotna. Wykonywany jest test na istotność współczynnika w modelu AR(1) dla reszt

```
dwtest(model,order.by=NULL,data=dane)
```

Durbin-Watson test

data: model

DW = 1.3891, p-value = 0.0347

alternative hypothesis: true autocorrelation is greater than 0

Uwaga `order.by=NULL` (default) powoduje że reszty są uporządkowane w kolejności występowania – traktowane są jak szereg czasowy

```
dwtest(model,order.by=~Szerokość,data=dane)
```

Durbin-Watson test

data: model

DW = 1.9786, p-value = 0.3439

alternative hypothesis: true autocorrelation is greater than 0

- Test Breuscha-Godfreya niezależności reszt podobnie jak test Durбина Watsona ma statystykę testową opartą na współczynnikach autokorelacji przy czym w przypadku tego testu uwzględniane są współczynniki do rzędu `order` w modelu AR(`order`).

```
bgtest(model,order=3,data=dane)
```

Breusch-Godfrey test for serial correlation of order up to 3

data: model

LM test = 5.1493, df = 3, p-value = 0.1612

## Testy liniowości modelu

- Test Harveya-Colliera liniowości modelu rekurencyjnie wyznaczonych reszt (*recursive residuals* -metodologia CUSUM) używany jest do zweryfikowania hipotezy, czy zależność między zmiennymi objaśniającymi a zmienną objaśnianą ma charakter liniowy. Wyznaczane reszty rekurencyjne są nieskorelowane w odróżnieniu od zwykłych reszt. Jeśli zależność w modelu nie jest liniowa wartość średnia wyznaczonych rekurencyjnie reszt nie jest równa 0. Statystyką testową jest właśnie ta średnia.

```
harvtest(model,order.by=NULL,data=dane)
```

Harvey-Collier test

data: model

HC = 1.7223, df = 16, p-value = 0.1043

- Test Rainbow liniowości modelu rekurencyjnie opiera się na prostej intuicji, że jeżeli zależność w prawdziwym modelu jest liniowa to w modelu wyznaczonym na podzbiorze obserwacji powinniśmy zaobserwować podobne oceny współczynników modelu jak na całym zbiorze danych. Domyślnie porównywane są oceny współczynników modelu wyznaczone na całym zbiorze danych z ocenami wyznaczonymi na połowie obserwacji o najmniejszych wartościach cechy `order.by`.

```
raintest(model,order.by=NULL,data=dane)
```

Rainbow test

data: model

Rain = 1.4754, df1 = 12, df2 = 5, p-value = 0.3514

```
raintest(model,order.by=~Szerokość,data=dane)
```

Rainbow test

data: model

Rain = 0.52083, df1 = 12, df2 = 5, p-value = 0.8355

- Test Ramseya RESET (*regression specification error*) liniowości modelu polega na weryfikacji liniowości modelu przez dodanie do modelu dodatkowych regresorów i sprawdzeniu czy parametry modelu dla tych regresorów są istotnie różne od zera. Rozszerzony model jest więc postaci  $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}$  gdzie  $\mathbf{Z}$  jest generowane jako kolejne potęgi zmiennych  $\mathbf{X}$ . Parametrem `power` można określić które potęgi mają być rozważane (domyślnie są to potęgi druga i trzecia. Jeśli zmiennych jest dużo, to zamiast dodawać potęgi zmiennej  $\mathbf{X}$  można dodawać potęgi składowych głównych wyznaczonych przez (argument `type` określa czego potęgi mają być dodawane.

```
resettest(model,data=dane, type="regressor")
```

RESET test  
 data: model  
 RESET = 0.77846, df1 = 12, df2 = 5, p-value = 0.6673

```
resettest(model,data=dane, type="princomp")
```

RESET test  
 data: model  
 RESET = 1.9667, df1 = 2, df2 = 15, p-value = 0.1744

## Testy normalności

```
library(nortest)
shapiro.test(model$residuals) #test Shapiro-Wilka
sf.test(model$residuals) #test Shapiro-Francia
lillie.test(model$residuals) #test Kolmogorowa –Lillieforsa
cvm.test(model$residuals) #test Cramera –von Misses
ad.test(model$residuals) #test Andersona-Darlinga
```

```
library(tseries) #biblioteka z testem Jarque-Bery
jarque.bera.test(model$residuals)
```

```
library(moments) #biblioteka z testem D’Agostino
agostino.test(model$residuals)
```

Testy diagnostyczne modelu liniowego w jednej funkcji

```
require(gvlma)
gvlma.lm(model,alphalevel=0.05,timeseq=1:24)
```

Call:

```
lm(formula = Zużycie ~ Pojemność + Moc + Ładowność + Masa + Długość +
  Szerokość, data = dane)
```

Coefficients:

| (Intercept) | Pojemność | Moc        | Ładowność | Masa      | Długość    | Szerokość  |
|-------------|-----------|------------|-----------|-----------|------------|------------|
| 11.0700588  | 0.0019816 | -0.0148460 | 0.0013945 | 0.0071142 | -0.0002526 | -0.0074952 |

ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS  
 USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:  
 Level of Significance = 0.05

Call:

```
gvlma.lm(lmobj = model, alphalevel = 0.05, timeseq = 1:24)
```

|                    | Value   | p-value | Decision                |
|--------------------|---------|---------|-------------------------|
| Global Stat        | 8.49175 | 0.07514 | Assumptions acceptable. |
| Skewness           | 2.79000 | 0.09485 | Assumptions acceptable. |
| Kurtosis           | 0.06975 | 0.79171 | Assumptions acceptable. |
| Link Function      | 3.53244 | 0.06018 | Assumptions acceptable. |
| Heteroscedasticity | 2.09956 | 0.14734 | Assumptions acceptable. |

W pakiecie `mvnormtest` są testy wielowymiarowej normalności np. `mshapiro.test`

Graficzny test wielowymiarowej normalności oparty na kwadracie odległości Mahalanobisa

Jeżeli  $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ , to kwadrat odległości Mahalanobisa pomiędzy  $\mathbf{X}$  i  $\boldsymbol{\mu}$  ma rozkład  $\chi_p^2$ .

```
require(graphics)
ma <- cbind(1:6, 1:3)
(S <- var(ma))
mahalanobis(c(0, 0), 1:2, S)

x <- matrix(rnorm(100*3), ncol = 3)
stopifnot(mahalanobis(x, 0, diag(ncol(x))) == rowSums(x*x))
##- Here, D^2 = usual squared Euclidean distances
Sx <- cov(x)
D2 <- mahalanobis(x, colMeans(x), Sx)
plot(density(D2, bw = 0.5), main="Squared Mahalanobis distances, n=100, p=3") ; rug(D2)
qqplot(qchisq(ppoints(100), df = 3), D2,
       main = expression("Q-Q plot of Mahalanobis" * ~D^2 *
                          " vs. quantiles of" * ~chi[3]^2))
abline(0, 1, col = 'gray')
```