

# Modelowanie Matematyczne - Lab 3

Adam Przemysław Chojecki

28 maja 2024

## Punktacja

- Zadanie 1: 0.5 pkt
- Zadanie 2: 0.5 pkt
- Zadanie 3: 0.5 pkt
- Zadanie 4: 0.5 pkt
- Czystość kodu i zrozumiałe komentarze: 0.5 pkt

## Zadanie 1: Residua i wartości odstające (Dane: C02)

W tym zadaniu będziemy analizować różnicę między różnymi typami residuów, wykrywać wartości odstające i wpływowe oraz interpretować ich znaczenie.

1. Wczytaj zbiór danych C02 używając funkcji `data(C02)`.
2. Dopasuj model liniowy, w którym zmienną zależną będzie `uptake`, a zmienną wyjaśniającą będzie `conc`.
3. Oblicz residua modelu, znormalizowane residua oraz residua studentyzowane używając funkcji `residuals()`, `rstandard()` i `rstudent()`.
4. Na wykresie przedstaw zależność między przewidywanymi wartościami a residuami, znormalizowanymi residuami oraz residuami studentyzowanymi. Czy można wyciągnąć z wykresu jakieś ciekawe obserwacje?
5. Dopasuj model liniowy, w którym zmienną zależną będzie `uptake`, a zmiennymi niezależnymi `conc` i `Type`. Od teraz będziemy badać już tylko ten model.
6. Dla nowego modelu wykonaj punkty 3 oraz 4. Czy teraz widać w wykresie rezyduów jakąś zależność?

7. Zidentyfikuj wartości odstające: znormalizowane residuum  $> 2$ .
8. Zidentyfikuj punkty wpływowe używając kryterium dźwigni (leverages)  $> 2 \frac{p}{n}$  oraz odległości Cooka. Czy metody te wyznaczyły te same punkty?

## Zadanie 2: Współliniowość i VIF (Dane: LifeCycleSavings)

W tym zadaniu będziemy analizować współliniowość zmiennych przy użyciu współczynnika inflacji wariancji (VIF).

1. Wczytaj zbiór danych `LifeCycleSavings` używając funkcji `data(LifeCycleSavings)`.
2. Dodaj do danych przekształcone zmienne `pop15_sq = pop152` oraz `pop75_log = log(pop75)`.
3. Dopasuj model liniowy, w którym zmienną zależną będzie `sr`, a zmiennymi niezależnymi `pop15`, `pop75`, `dpi`, `ddpi`, `pop15_sq`, `pop75_log`.
4. Oblicz współczynniki VIF dla zmiennych niezależnych używając funkcji `vif()` z pakietu `car`.
5. Zidentyfikuj zmienne o współczynniku VIF  $> 10$  i usuń je z modelu. Należy to robić iteracyjnie - po usunięciu jednej ze zmiennych należy policzyć VIF jeszcze raz, gdyż może się on zmienić.
6. Dopasuj nowy model bez zmiennych o wysokim VIF i porównaj wyniki.

## Zadanie 3: Transformacja Box-Cox (Dane: airquality)

W tym zadaniu będziemy stosować transformację Box-Cox do poprawy dopasowania modelu.

1. Wczytaj zbiór danych `airquality` używając funkcji `data(airquality)`.
2. Usuń wiersze z brakującymi wartościami funkcją `na.omit()`.
3. Dopasuj model liniowy, w którym zmienną zależną będzie `Wind`, a zmienną niezależną `Temp`.
4. Zastosuj transformację Box-Cox używając funkcji `boxcox()` z pakietu `MASS`. Czy wykres sugeruje, że liniowa zależność jest poprawna?
5. Dopasuj nowy model na podstawie transformowanej zmiennej zależnej.
6. Porównaj jakość dopasowania obu modeli (przed i po transformacji) używając `summary()` - opisz różnice (np. istotność zmiennej, współczynnik dopasowania  $R^2$ , rezydual).

## Zadanie 4: Wybór cech (Dane: mtcars)

W tym zadaniu będziemy przeprowadzać wybór cech przy użyciu kryteriów AIC i BIC oraz funkcji `step()`.

1. Wczytaj zbiór danych `mtcars` używając funkcji `data(mtcars)`.
2. Dopasuj dwa modele ze zmienną zależną `qsec`:
  - pełny model, w którym bierzemy wszystkie pozostałe zmienne:  
`full_model <- lm(qsec ~ ., data = mtcars)`
  - pusty model, w którym jest tylko intercept:  
`null_model <- lm(qsec ~ 1, data = mtcars)`
3. Użyj funkcji `step()` z kryterium AIC, aby znaleźć najlepszy model używając metody wstecz (`backward`).  
`step_backward_aic <- step(full_model, direction = "backward")`
4. Użyj funkcji `step()` z kryterium AIC, aby znaleźć najlepszy model używając metody do przodu (`forward`).  
`step_forward_aic <- step(null_model,  
scope = list(lower = null_model, upper = full_model),  
direction = "forward")`
5. Porównaj otrzymane modele (metody `backward` i `forward`). Czy funkcja `step()` wybrała te same zmienne do modeli? Jeśli nie, to porównaj je pod względem jakości dopasowania oraz interpretacji.
6. Wykonaj procedurę wyboru wstecz (`backward`) dla kryterium BIC. Porównaj znaleziony model z tym z punktu 3 (w tym celu zmodyfikuj parametr `k` w funkcji `step()`).  
Czy modele są takie same? Czy któryś z nich ma więcej zmiennych?