Analiza Danych
Podstawy Statystyczne
(ADPS)

Laboratorium 4

Przykład 1 – analiza wariancji

- Wygeneruj 5 prób losowych z tego samego rozkładu normalnego o licznościach 25:

```
n = 25; mi = 20; sigma = 1
y1 = rnorm(n, mi, sigma)
y2 = rnorm(n, mi, sigma)
y3 = rnorm(n, mi, sigma)
y4 = rnorm(n, mi, sigma)
y5 = rnorm(n, mi, sigma)
```

- Scal dane w ramkę i wyświetl wykresy pudełkowe:

```
y = cbind.data.frame(y1, y2, y3, y4, y5)
boxplot(y)
```

Przykład 1 – analiza wariancji

- Przygotuj dane na potrzeby funkcji aov():

```
dane_anova = stack(y)
names(dane_anova) <- c('dane', 'proba')
```

- Sprawdź hipotezę o równości wariancji w próbach:

```
bartlett.test(dane~proba, data = dane_anova)
```

- Przeprowadź analizę wariancji przy założeniu normalności rozkładów:

```
aov_res = aov(dane~proba, data = dane_anova)
summary(aov_res)
```

Przykład 1 – analiza wariancji

- Przeprowadź analizę wariancji bez założenia normalności rozkładów korzystając z testu Kruskala-Wallisa:

  ```
  kruskal.test(dane~proba, dane_anova)
  ```

Przykład 1 – analiza wariancji

- Korzystając z metody Tukeya sprawdź, czy dla którejś z prób jej wartość średnia odbiega od wartości średnich w pozostałych próbach:

```
Tukey_res = TukeyHSD(aov_res)
print(Tukey_res)
plot(Tukey_res)
```

Przykład 1 – analiza wariancji

- Powtórz testy dla danych, w których nie ma równości wartości średnich w poszczególnych próbach:

```
n = 25; mi = 20; sigma = 1; al_2 = 1,2 lub 4
y1 = rnorm(n, mi, sigma)
y2 = rnorm(n, mi, sigma) + al_2
y3 = rnorm(n, mi, sigma)
y4 = rnorm(n, mi, sigma)
y5 = rnorm(n, mi, sigma)
```

Przykład 1 – analiza wariancji

- Wygeneruj dane, gdy próby mają różne długości:

```
n = 25; mi = 20; sigma = 1; al_4 = -2
y1 = rnorm(n, mi, sigma)
y2 = rnorm(n+1, mi, sigma)
y3 = rnorm(n+2, mi, sigma)
y4 = rnorm(n+3, mi, sigma) + al_4
y5 = rnorm(n+4, mi, sigma)
dane_anova = data.frame( dane = c(y1, y2, y3, y4, y5),
    proba = rep( c("y1", "y2", "y3", "y4", "y5"), times =
    c(length(y1), length(y2), length(y3), length(y4), length(y5))) )
```

- Powtórz wcześniej przeprowadzane testy dot. analizy wariancji dla powyższych danych.

Przykład 2 – regresja liniowa

- Wygeneruj dane zgodnie z poniższymi komendami:

```
n = 100; mi = 0; sigma = 2
x = rnorm(n, mi, sigma)
e = rnorm(n, 0, 1)
b0 = 1; b1 = 2
y = b1*x + b0 + e
plot(x, y)
```

Przykład 2 – regresja liniowa

- Wyznacz parametry prostej regresji y = b1*x + b0 i współczynnik determinacji R^2:

```
b1_est = (mean(x*y) - mean(x)*mean(y)) / (mean(x^2) - (mean(x))^2)
b0_est = mean(y) - b1_est*mean(x)
y_est = b1_est*x + b0_est
R2 = 1 - sum((y - y_est)^2)/sum((y - mean(y))^2)
```

- Nanieś na rysunek prostą regresji:

```
arg = c(min(x), max(x))
out = b1_est*arg + b0_est
lines(arg, out, col = 'red')
```

Przykład 2 – regresja liniowa

- To samo wykonaj za pomocą funkcji lm:

```
lm_res = lm(y~x)
summary(lm_res)
arg = c(min(x), max(x))
out = coef(lm_res)[2]*arg + coef(lm_res)[1]
lines(arg, out, col = 'green')
```

Przykład 2 – regresja liniowa

- Wyznacz parametry prostej regresji x = c1*y + c0 i nanieś ją na wykres z danymi:

```
lm_res = lm(x~y)
summary(lm_res)
arg = c(min(y), max(y))
out = coef(lm_res)[2]*arg + coef(lm_res)[1]
lines(out, arg, col = 'blue')
```

Przykład 3 – regresja, przyp. wielowym.

- Wygeneruj dane zgodnie z poniższymi komendami:

```
n = 100
x0 = rep(1, n)
x1 = (1:n)/n
x2 = sin(1:n)
x3 = runif(n, -1, 1)
x = cbind(x0, x1, x2, x3)
b = c(1, -2, 3, -1)
e = rnorm(n, 0, 0.5)
y = x%*%b + e
```

- Narysuj zmienną y:

```
plot(y, type = 'l')
```

Przykład 3 – regresja, przyp. wielowym.

- Wyznacz parametry modelu korzystając z metody regresji liniowej:

```
lm_res = lm(y ~ 1 + x1 + x2 + x3)
summary(lm_res)
```

- Sprawdź jaki jest wynik wywołania linii:

```
lm_res = lm(y ~ x0 + x1 + x2 + x3 - 1)
summary(lm_res)
```

- Wartości estymat parametrów modelu można obliczyć w następujący sposób:

```
b_est = solve(t(x)%*%x)%*%t(x)%*%y
```

Pliki z danymi ze strony bossa.pl

- Kursy spółek giełdowych:
  http://www.galera.ii.pw.edu.pl/~kjedrze1/Lab4/mstall.zip
- Kursy indeksów giełd zagranicznych:
  http://www.galera.ii.pw.edu.pl/~kjedrze1/Lab4/mstzgr.zip
- Kursy walut:
  http://www.galera.ii.pw.edu.pl/~kjedrze1/Lab4/mstnbp.zip

Wczytanie danych z plików .mst

- Pomocnicza funkcja wczytująca dane z plików .mst:

```
wczytaj_mst = function(plik_zip, plik_mst) {
  unzip(plik_zip, plik_mst)
  dane = read.csv(plik_mst)
  names(dane) = c('ticker', 'date', 'open', 'high', 'low', 'close', 'vol')
  dane$date = as.Date.character(dane$date,format = '%Y%m%d')
  dane }
```

- Przykład użycia:

```
dane = wczytaj_mst('mstall.zip', 'KGHM.mst')
```

Zadanie 1

- Korzystając z metod analizy wariancji, dla wybranej spółki notowanej na GPW zweryfikuj hipotezę o równości wartości średnich procentowych zmian cen zamknięcia
  - w ostatnich sześciu miesiącach (od listopada do kwietnia),
  - w ostatnich trzech miesiącach (od lutego do kwietnia).

*wskazówki:

- obliczenie procentowych zmian cen zamknięcia:

```
dane$close_ch= with(dane, c(NA,100*diff(close))/close)
```

- przykładowy sposób wczytania danych dot. np. stycznia 2017:

```
y1 = with(dane, close_ch[format(date, '%Y-%m') == '2017-01'])
```

Zadanie 2

- Korzystając z regresji liniowej
  - wyznacz zależność indeksu WIG20 od kursów zamknięcia spółek ASSECO, ENERGA, EUROCASH, ORANGE, PKNORLEN, PKOBP, PZU, PEKAO dla danych z roku 2017,
  - oceń istotność poszczególnych zmiennych objaśniających w tak skonstruowanym modelu,
  - przeprowadź analogiczne analizy w przypadku uwzględnienia w modelu mniejszej ilości spółek, np.: PKNORLEN, PKOBP, PZU, PEKAO.

Zadanie 3

- Korzystając z regresji liniowej
  - zbadaj zależności pomiędzy kursami walut USD, EUR, GBP, CHF dla danych z roku 2017, np. zależność kursu CHF od pozostałych ww. walut; wykorzystaj dane z pliku mstnbp.zip,
  - zbadaj zależności pomiędzy indeksami DJIA, NIKKEI, FT-SE100, DAX, WIG20 dla danych z roku 2017, np. zależność indeksu DAX od pozostałych ww. indeksów; wykorzystaj dane z pliku mstzgr.zip*,
  - zbadaj zależności pomiędzy kursami walut USD, EUR, a indeksami giełdowymi DJIA, NIKKEI, FT-SE100, DAX, WIG20.

*uwaga: daty notowań w różnych krajach mogą różnić się, wskazówki dot. stworzenia odpowiedniej ramki z danymi na następnym slajdzie.

Wskazówki do zadania 3

- Stworzenie podramek, np. dla DJIA:

```
dane = wczytaj_mst('mstzgr.zip', 'DJIA.mst')
DJIA_df = subset(dane, format(date, '%Y') == '2017', select =
c('date', 'close'))
names(DJIA_df) = c('date', 'DJIA')
```

- Połączenie danych:

```
ALL_df = merge(DJIA_df, NIKKEI_df, by = 'date')
ALL_df = merge(ALL_df, FT_SE100_df, by = 'date')
itd.
```

Zadanie 4

- W poniższej tabeli przedstawiona jest długość drogi hamowania y w zależności od prędkości pojazdu v.

v **(km/h)**	***y*** **(m)**
33.0	4.7
33.0	4.1
49.1	10.3
65.2	22.3
78.5	33.4
93.0	44.4

  - Dopasuj liniowe zależności opisujące długość drogi y oraz pierwiastek tej długości $\sqrt{y}$ w funkcji prędkości v.
  - Nanieś odpowiednie linie przedstawiające te zależności na rysunek z zarejestrowanymi danymi.
  - Która z liniowych zależności jest bardziej zgodna z danymi?

Zadanie 5

- Plik oldfaithful.txt znajdujący się pod adresem http://www.galera.ii.pw.edu.pl/~kjedrze1/Lab4/oldfaithful.txt zawiera czasy erupcji pewnego gejzeru oraz następujące po nich odstępy czasu do kolejnej erupcji (wszystkie czasy podane są w minutach), które zostały zarejestrowane w ciągu ośmiu kolejnych dni.
  - Na płaszczyźnie xy (x – czas erupcji, y – odstęp) zaznacz punkty odpowiadające danym. Metodą regresji liniowej dopasuj do nich prostą.
  - Oszacuj czas, który upłynie do kolejnej erupcji gejzeru, jeśli poprzednia erupcja trwała 2 min. Oszacuj odchylenie standardowe błędu z jakim został wyznaczony ten czas.
  - Powtórz powyższy punkt przy założeniu, że erupcja trwała 4 min.