PADR 2019/2020

Praca domowa nr 1 (max. = 15 p.)

Maksymalna ocena: 15 p. (5 zadań po max. 3 p.)

Termin oddania pracy: 05.11.2019 r., godz. 23:59

Do przesłania na adres `A.Geras@mini.pw.edu.pl` ze swojego konta pocztowego `*@*pw.edu.pl`:

- `Nick_Nazwisko_Imie_NrAlbumu_pd1.R` (kody źródłowe funkcji);
- `Nick_Nazwisko_Imie_NrAlbumu_pd1.Rmd` (demonstracja działania oraz szczegółowe testy napisanych funkcji w formie estetycznie sformatowanego raportu w Markdown/`knitr` – przykłady powinny być o wiele bardziej rozbudowane niż te poniżej);

  Uwaga: w raporcie pierwsze polecenie `R` do wykonania to:

  ```
  source("Nick_Nazwisko_Imie_NrAlbumu_pd1.R")
  ```

- `Nick_Nazwisko_Imie_NrAlbumu_pd1.html` (skompilowana wersja powyższego).

Temat wiadomości: `[PADR-1920] Praca domowa nr 1`.

Uwaga:

- We wszystkich funkcjach sprawdź, czy założenia odnośnie przekazanych argumentów są spełnione (funkcja `stopifnot()`);
- Za zadanie wykonane przy użyciu pętli będzie automatycznie przyznawane 0 punktów.

1 Zadanie 1

Dana jest macierz o `n` wierszach i `m` kolumnach. Używając operacji na macierzach (mnożenie macierzy, transpozycja, sumowanie kolummn/wierszy itp.), stwórz funkcję `macierz_korelacji()` obliczającą macierz korelacji `C`, tj. macierz rozmiaru $m \times m$, taką że $c[i, j]$ oznacza liniowy współczynnik korelacji Pearsona pomiędzy `i`-tą i `j`-tą kolumną.

Uwaga: W implementacji funkcji `macierz_korelacji` nie można użyć funkcji `cor`.

Przykład.

```
A <- matrix( c(3,4,5,6,12,5,73,5,3,2,1,4), ncol=3)
A
```

```
##      [,1] [,2] [,3]
## [1,]    3   12    3
## [2,]    4    5    2
## [3,]    5   73    1
## [4,]    6    5    4
```

```
macierz_korelacji(A)
```

```
##           [,1]       [,2]       [,3]
## [1,] 1.0000000  0.1838759  0.2000000
## [2,] 0.1838759  1.0000000 -0.7707141
## [3,] 0.2000000 -0.7707141  1.0000000
```

Poprawność rozwiązania można sprawdzić przy użyciu funkcji `cor`.

```r
cor(A,A)
```

```
##            [,1]       [,2]       [,3]
## [1,] 1.0000000  0.1838759  0.2000000
## [2,] 0.1838759  1.0000000 -0.7707141
## [3,] 0.2000000 -0.7707141  1.0000000
```

2 Zadanie 2

Napisz funkcję `podsumowanie()`, która dla danej na wejściu macierzy rozmiaru $n \times m$ ($n > 3$, $m > 3$) zwraca dwualementową listę zawierającą:

1. wektor zawierający średnie 3 najmniejszych elementów każdej kolumny,
2. macierz rozmiaru $(n-1) \times m$ zawierającą średnie sąsiednich elementów w każdej kolumnie, przy czym sąsiadem elementu `A[i, j]` jest element `A[i+1, j]`.

```r
A <- matrix( c(3,4,5,6,12,5,73,5,3,2,1,4), ncol=3)
A
```

```
##      [,1] [,2] [,3]
## [1,]    3   12    3
## [2,]    4    5    2
## [3,]    5   73    1
## [4,]    6    5    4
```

```r
podsumowanie(A)
```

```
## [[1]]
## [1] 4.000000 7.333333 2.000000
##
## [[2]]
##      [,1] [,2] [,3]
## [2,]  3.5  8.5  2.5
## [3,]  4.5 39.0  1.5
## [4,]  5.5 39.0  2.5
```

3 Zadanie 3

W talii kart liczącej 52 sztuki są cztery kolory: **pik**, **kier**, **karo**, **trefl**. Każdy z kolorów posiada 9 kart numerowanych $\{2, 3, 4, 5, 6, 7, 8, 9, 10\}$, 3 figury **J**, **Q**, **K** oraz Asa **A**. Przeanalizujmy uproszczoną grę w *blackjacka*.

1. Obserwujemy grę dwóch graczy.

2. Zadaniem gracza jest uzyskać jak najbliżej (ale nie więcej niż) 21 punktów. Wynik powyżej 21 punktów oznacza przegraną. Na przykład, jeśli karty gracza 1 mają wartość 17, a karty gracza 2 mają wartość 22, wygrywa gracz 1.

3. Każda karta w tali ma przypisaną wartość punktową. I tak:

- karty numerowane przyjmują wartość zgodną z miejscem w talii (np. 2 ma wartość 2, 9 ma wartość 9 itd.),
- figury wartości równe 10,
- zaś As przyjmuje wartość 1 lub 11, w zależności co jest *lepsze* dla gracza. Zauważmy, że zbiór "AAA" ma wartość 13.

Na obecnym etapie gry, gracz numer 1 wie, że:

1. na ręce ma **6 trefl** oraz **J pik** (wartość 16),
2. drugi gracz ma dwie karty, jedna z nich to **K trefl**, druga zaś nie jest widoczna.

Gracze mają teraz możliwość dobrania jeszcze jednej karty. Wiedząc, że gracz 2 zdecyduje, że nie dobiera więcej kart, metodą symulacyjną sprawdź, jaką strategię powiniem obrać gracz numer 1 – dobrać czy nie dobierać karty.

Na podstawie 1000 rozgrywek, oszacuj prawdopodobieństwo wygrania gdy gracz 1 dobierze jeszcze jedną kartę oraz prawdopodobieństwo wygrania gdy zdecyduje się nie dobierać dodatkowej karty.

Zakładamy, że prawdopodobieństwa wylosowania poszczególnych kart są takie same.

```
## [1] "Przybliżone prawdopodobieństwo, że gracz 1 wygrywa:"
```

```
## [1] "jeśli nie dobiera karty: 0.3230"
```

```
## [1] "jeśli dobiera kartę: 0.2360"
```

4 Zadanie 4

Dysponujemy zbiorem danych zawierających informację na temat upodobań Szkotów oraz Anglików. Zbiór będziemy reprezentować jako macierz postaci:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & x_{n4} & x_{n5} \end{bmatrix} \in \mathbb{R}^{n \times 5},$$

gdzie i-ty wiersz macierzy $\mathbf{X}$, $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5})$ opisuje wektor cech i-tego uczestnika badania, $i = 1, \ldots, n$. Dokłądniej:

- $x_{i1} \in \{0, 1\}$ opisuje czy i-ty uczestnik oglądał film *Braveheart* (`1` jeśli tak, `0` w przeciwnym przypadku),
- $x_{i2} \in \{0, 1\}$ opisuje czy i-ty uczestnik pije whiskey,
- $x_{i3} \in \{0, 1\}$ opisuje czy i-ty uczestnik lubi piwo,
- $x_{i4} \in \{0, 1\}$ opisuje czy i-ty uczestnik oglądał mecz reprezentacji Anglii w piłkę nożną,
- $x_{i5} \in \{0, 1\}$ opisuje czy i-ty uczestnik czyta plotki na temat rodziny królewskiej.

Na przykład, następujący wektor cech: $(1, 1, 0, 0, 0)$ opisuje osobę, która widziała film *Braveheart*, pije wishkey, nie lubi piwa, nigdy nie oglądała gry reprezentacji Anglii w piłkę nożną i nie czyta plotek na temat brytyjskiej rodziny królewskiej.

Dodatkowo, dysponujemy wektorem $c = c(c_1, c_2, \ldots, c_n)$, który określa narodowość uczestnika badania tj. *Anglik* lub *Szkot*.

Macierz $\mathbf{X}$ i wektor $\mathbf{c}$ będziemy nazywać **zbiorem uczącym**.

Wyobraźmy sobie, że pojawia się teraz wektor opisujący upodobania pewnego nowego człowieka, $\mathbf{z} = (z_1, z_2, z_3, z_4, z_5)$. Wektor ten nie jest wierszem macierzy $\mathbf{X}$ – zatem jego klasa c nie jest znana. Naszym zadaniem będzie jej odgadnięcie (na podstawie informacji ze zbioru uczącego). W tym celu posłużymy się **naiwnym klasyfikatorem Bayesowskim**.

Zauważmy, że do obliczenia prawdopodobieństwa, że i-ty uczestnik jest Szkotem lub Anglikiem, możemy użyć reguły Bayesa:

$$\mathbb{P}(c = \text{Szkot}|\mathbf{x}_i) = \frac{\mathbb{P}(\mathbf{x}_i|c = \text{Szkot})\mathbb{P}(c = \text{Szkot})}{\mathbb{P}(\mathbf{x}_i)}$$

$$\mathbb{P}(c = \text{Anglik}|\mathbf{x}_i) = \frac{\mathbb{P}(\mathbf{x}_i|c = \text{Anglik})\mathbb{P}(c = \text{Anglik})}{\mathbb{P}(\mathbf{x}_i)}$$

Mianownik w powyższych równaniach możemy pominąć[1] uzyskując w ten sposób następujące równania:

$$\mathbb{P}(c = \text{Szkot}|\mathbf{x}_i) \propto \mathbb{P}(\mathbf{x}_i|c = \text{Szkot})\mathbb{P}(c = \text{Szkot})$$
$$\mathbb{P}(c = \text{Anglik}|\mathbf{x}_i) \propto \mathbb{P}(\mathbf{x}_i|c = \text{Anglik})\mathbb{P}(c = \text{Anglik})$$

1. Prawdopodobieństwa (a priori) $\mathbb{P}(c = \text{Szkot})$ oraz $\mathbb{P}(c = \text{Anglik})$ możemy weystymować na podstawie zbioru uczącego w następujący sposób:

   $$\mathbb{P}(c = \text{Anglik}) = \frac{n_{\text{Anglik}}}{n},$$

   gdzie n_{Anglik} oznacza liczbę Anglików w zbiorze uczącym, zaś n liczbę wszystkich osób w zbiorze uczącym. W przypadku wyznaczenia $\mathbb{P}(c = \text{Szkot})$ postępujemy analogicznie.
2. Aby obliczyć $\mathbb{P}(\mathbf{x}_i|c = \text{Szkot})$ oraz $\mathbb{P}(\mathbf{x}_i|c = \text{Anglik})$ zakładamy warunkową niezależność:

   $$\mathbb{P}(\mathbf{x}_i|c = \text{Szkot}) = \mathbb{P}(x_{i1}|c = \text{Szkot})\mathbb{P}(x_{i2}|c = \text{Szkot})\ldots\mathbb{P}(x_{i5}|c = \text{Szkot})$$

3. Zauważmy, że w zbiorze uczącym niektóre z prawdopodobieństw warunkowych mogą być równe 0, co może powodować błędną klasyfikację. Aby uniknąć tego problemu zastosujemy korektę Laplace'a:

   $$\mathbb{P}(x_{\cdot l} = 1|c = c') = \frac{|\{j : x_{jl} = 1 \text{ gdzie } c = c'\}| + 1}{|\{j : x_{jl} = 1 \text{ gdzie } c = c'\}| + |\{j : x_{jl} = 0 \text{ gdzie } c = c'\}| + 2}.$$
4. Klasyfikacja (predykcja klasy) w przypadku tego klasyfikatora odbywa się według następującej zasady:

$$M(\mathbf{z}) = \arg\max_{c'} \left\{ \log \mathbb{P}(c = c') + \sum_{i=1}^{5} \log \mathbb{P}(z_i|c = c') \right\}$$

Napisz funkcję `naiwy_klasyfikator(X, c, z)`, będzie przypisywać klasę nowym wektorom przy pomocy naiwnego klasyfikatora bayesowskiego.

Funkcja powinna zwrócić listę dwuelementową z ustawionym atrybutem `names`:

- pierwszy element (`prob`) powinien zawierać wartość wyestymowanych prawdopodobieństw,
- drugi element (`group`) powinien zawierać wyznaczoną klasę czyli narodowość uczestnika.

Zbiór treningowy:

```
X <- structure(c(0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1,
1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0,
0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0,
0, 0, 1, 0, 0, 0), .Dim = c(13L, 5L))
X
```

```
##      [,1] [,2] [,3] [,4] [,5]
##  [1,]    0    0    1    1    0
##  [2,]    0    1    1    1    1
##  [3,]    1    0    0    1    1
##  [4,]    0    1    1    1    1
##  [5,]    1    1    1    1    1
##  [6,]    0    0    1    1    1
##  [7,]    1    1    1    0    0
```

[1] Dlaczego?

```
##  [8,]    1    1    0    0    0
##  [9,]    0    1    1    0    0
## [10,]    1    1    0    0    1
## [11,]    1    0    0    1    0
## [12,]    0    1    0    0    0
## [13,]    1    1    0    0    0
```

```r
c <- rep(c('Anglik', 'Szkot'), c(6, 7))
c
```

```
##  [1] "Anglik" "Anglik" "Anglik" "Anglik" "Anglik" "Anglik" "Szkot"
##  [8] "Szkot"  "Szkot"  "Szkot"  "Szkot"  "Szkot"  "Szkot"
```

```r
z <- c(1, 1, 0, 1, 0)

naiwny_bayes(X, c, z)
```

```
## $prob
## $prob$apriori
## c
##    Anglik     Szkot
## 0.4615385 0.5384615
##
## $prob$Anglik
## x_1 = 1 x_2 = 1 x_3 = 1 x_4 = 1 x_5 = 1
##   0.375   0.500   0.750   0.875   0.750
##
## $prob$Szkot
##    x1 = 1    x2 = 1    x3 = 1    x4 = 1    x5 = 1
## 0.6666667 0.7777778 0.3333333 0.2222222 0.2222222
##
##
## $group
## [1] "Szkot"
```

```r
naiwny_bayes(X, c, c(0, 0, 1, 0, 1))$group
```

```
## [1] "Anglik"
```

5 Zadanie 5

Pairs bootstrap jest metodą obliczania przedziału ufności dla parametrów w modelu regresji liniowej prostej. Zadanie regresji liniowej polega na opisaniu liniowej zależności zmiennej objaśnianej y od zmiennej objaśniającej x, tj., wyznaczeniu współczynników prostej

$$y = \alpha x + \beta$$

w taki sposób by minimalizowały one sumę kwadratów błędów

$$E(\alpha, \beta; \mathtt{x}, \mathtt{y}) = \sum_{i=1}^{n} (\alpha + \beta x_i - y_i)^2 .$$

W tymcelu możemy wykorzystać funkcję `lm()`, tj.:

```r
model = lm(y~x)
model$coefficients # wektor wyestymowanych współczynników prostej
```

Napisz funkcję `pairs_bootstrap(x, y, M)` obliczającą 95-procentowy, dwustronny przedział ufności dla współczynnika nachylenia w modelu regresji liniowej jednowymiarowej. Argumenty funkcji to:

- wektory zawierające wartości zmiennej objaśnianej y oraz zmiennej objaśniającej x,
- a także liczba powtórzeń eksperymentu M.

Funkcja powinna wykonać M razy następujące czynności:

1. Próbkowanie (ang. *resampling*) danych wejściowych: losujemy pary (x_i, y_i) ze zwracaniem, przy czym $x_i \in X, y_i \in Y, i = 1, .., n$, gdzie n to długość wektora X) **Uwaga! Nie gubimy zależności pomiędzy zmiennymi objaśnianą i objaśniającą.** (zob. funkcja `sample()`)
2. Obliczenie współczynnika nachylenia dla spróbkowanych danych otrzymanych w punkcie powyżej (w tym celu możesz użyć funkcji `lm()`).

W celu wyznaczenia przedziału ufności należy policzyć odpowiednie kwartyle otrzymanego w ten sposób wektora współczynników nachylenia (zob. funkcję `quantile()`).

Dane:

```
x <- 1:15
y <- 3*1:15+rnorm(15,0,2)+2
```

Bootstapowy przedział ufności:

```
model <- lm(y~x)
przedzial <- pairs_bootstrap(x, y, 1000)
print(paste0("(", round(przedzial[1],2),", ",round(przedzial[2],2), ")"))
```

```
## [1] "(2.78, 3.31)"
```

```
plot(x,y, pch=16, las=1)
curve(przedzial[1]*x+2, col="red", add=TRUE)
curve(przedzial[2]*x+2, col="red", add=TRUE)
```