

Projekt 1 Markdown

Monika Etrych

8.01.2022

Projekt stworzony w ramach zajęć: Wstęp do Analizy Danych.

Analiza Danych

pytanie 1

Jaki jest rozkład wykształcenia w miejscowościach (we wsiach)?

```
library(dplyr)

##
## Dołączanie pakietu: 'dplyr'

## Następujące obiekty zostały zakryte z 'package:stats':
##
##      filter, lag

## Następujące obiekty zostały zakryte z 'package:base':
##
##      intersect, setdiff, setequal, union

setwd("C:\\Users\\monik\\Desktop\\Wstęp")
budzet = read.csv(file="budzety2005.csv")

wyksztalzenie=budzet %>%
  filter(townsize=="villages")%>%
  count(educ, sort=TRUE) %>%
  rename(number=n) %>%
  mutate(percent=round(number/sum(budzet$townsize=="villages")*100, 2)) %>%
  rename(education=educ)

wyksztalzenie$education[wyksztalzenie$education == "n/a"] <- "no data"
wyksztalzenie

##           education number percent
## 1           primary    4048   36.68
## 2 vocational training    2822   25.57
## 3           high school    2067   18.73
## 4             no data    1350   12.23
## 5           university     483    4.38
## 6         no education     266    2.41
```

1/3 osób mieszkających na wsi ma wykształcenie podstawowe. Połowa osób to osoby z wykształceniem podstawowym lub zawodowym.

pytanie 2

Ile średnio mieszkańców mieszka w mieście(powyżej 100tys), a ile na wsi?

```
mieszkancyWies=budzet %>%
  group_by (hhno) %>%
  filter(townsize=="villages") %>%
  summarize(liczbaMieszkancow = max(persno))

## [1] "Średnia liczba osób mieszkających na wsi"

round(mean(mieszkancyWies$liczbaMieszkancow, na.rm=T))

## [1] 3

mieszkancyMiasto=budzet %>%
  group_by (hhno) %>%
  filter(townsize=="100,000 - 199,999" || townsize== "200,000 -
499,999" || townsize=="500,000 and over") %>%
  summarize(liczbaMieszkancow = max(persno))
mieszkancyMiasto

## # A tibble: 4,513 x 2
##       hhno liczbaMieszkancow
##   <int>         <int>
## 1 100010111             3
## 2 100010121             2
## 3 100010211             2
## 4 100010221             1
## 5 100010321             3
## 6 100010421             2
## 7 100010611             2
## 8 100010711             3
## 9 100010921             3
## 10 100011011            2
## # ... with 4,503 more rows

## [1] "Średnia liczba osób mieszkających w miastach powyżej 100 tys."

round(mean(mieszkancyMiasto$liczbaMieszkancow, na.rm=T))

## [1] 2
```

pytanie 3

W jakim miesiącu średnio są najwyższe wydatki? O ile średnio wzrastają wydatki w grudniu (ile wydajemy na święta Bożego Narodzenia)?

```

miesiac = budzet %>%
  group_by(month_hh) %>%
  summarize(mean(totexpGUS))
miesiac

## # A tibble: 12 x 2
##   month_hh `mean(totexpGUS)`
##   <int>         <dbl>
## 1         1         1904.
## 2         2         1865.
## 3         3         2017.
## 4         4         1900.
## 5         5         2080.
## 6         6         2040.
## 7         7         2088.
## 8         8         2072.
## 9         9         2222.
## 10        10         2220.
## 11        11         2079.
## 12        12         2319.

## [1] "Średnie wydatki w miesiącu"

rok=mean(miesiac$`mean(totexpGUS)` )
rok

## [1] 2067.148

## [1] "Najwyższe wydatki są w grudniu, a potem we wrześniu i październiku
(prawdopodobnie przez wydatki szkolne)"

grudzien = budzet %>%
  filter(month_hh==12)%>%
  select(totexpGUS)

## [1] "Średnia wydatków w grudniu"

gr=round(mean(grudzien$totexpGUS))
gr

## [1] 2319

## [1] "Średnie wydatki w grudniu są wyższe od średnich wydatków w roku o: "
round(gr-rok)

## [1] 252

```

pytanie 4

Jaki jest najczęstszy wiek osoby posiadającej własne mieszkanie?

```
wiek2=budzet %>%
  filter(owntype=="owned outright" || owntype == "owned with mortgage",
         reltohof=="Head") %>%
  count(age) %>%
  rename(liczba=n) %>%
  arrange(age)

wiek2[which.max(wiek2$liczba), ]

##      age liczba
## 41    57     402
```

pytanie 4.5

W jakim wieku są właściele nieruchomości?

```
wiek = budzet %>%
  filter(owntype=="owned outright" || owntype == "owned with mortgage",
         reltohof=="Head") %>%
  select(age)

quantile(wiek$age, probs=c(0.25, 0.5, 0.75))

## 25% 50% 75%
## 35  51  64
```

Najczęstszy wiek osoby posiadającej własne mieszkanie to 57 lat. Osoby posiadające mieszkanie są w różnym wieku, nie dominują osoby starsze.

5 ciekawych zadań

Zadanie 1

Na jednym wykresie liniowym narysować, jak w rozważanych latach zmieniała się oczekiwana długość życia w trzech wybranych przez Ciebie krajach.

```
library(gapminder)
library(tidyr)
library(dplyr)

polska=gapminder %>%
  filter(country=="Poland") %>%
  select(lifeExp, year)
polska

dania=gapminder %>%
  filter(country=="Denmark") %>%
```

```

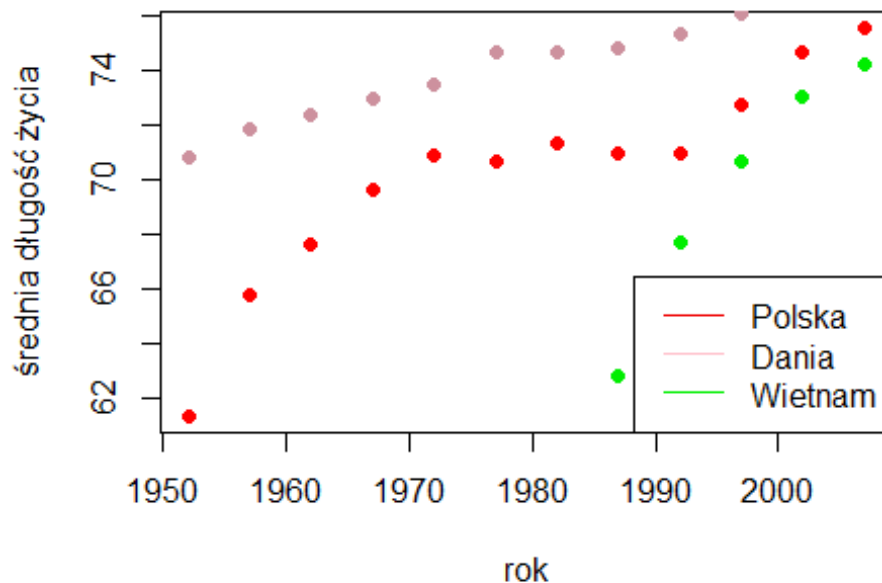
    select(lifeExp, year)
dania

vietnam = gapminder %>%
  filter(country=="Vietnam") %>%
  select(lifeExp, year)
vietnam

plot(polska$year, polska$lifeExp, col="red", pch = 19, xlab="rok",
     ylab="średnia długość życia")
points(dania$year, dania$lifeExp, col="pink3", pch = 19)
points(vietnam$year, vietnam$lifeExp, col="green2", pch = 19)

legend(x="bottomright", legend=c("Polska", "Dania", "Wietnam"), col=c("red2",
"pink", "green2"), lty=1)

```



Oczekiwana wartość życia we wszystkich krajach w ostatnich latach wzrosła - ludzie średnio żyjąc dłużej.

Zadanie 2

Wyfiltrować dane z roku 1987 dla krajów europejskich. Dodać zmienną system, która będzie przyjmowała trzy wartości: RWPG dla krajów RWPG, UE dla krajów Unii Europejskiej oraz inne dla pozostałych krajów. Następnie wyznaczyć podstawowe statystyki opisowe (średnia, mediana, odchylenie standardowe, Q1 i Q3). Czy można wyciągnąć z w/w statystyk jakieś ciekawe wnioski?

```

library(gapminder)
library(tidyr)
library(dplyr)

rwp=c("Bulgaria", "Cuba", "Czechoslovakia", "East Germany", "Hungary",
      "Mongolia", "Poland", "Romania", "Soviet Union", "Vietnam")
ue=c("Austria", "Belgium", "Bulgaria", "Croatia", "Cyprus", "Czech Republic",
     "Denmark", "Estonia", "Finland", "France", "Germany", "Greece", "Hungary",
     "Ireland", "Italy", "Latvia", "Lithuania", "Luxembourg", "Malta",
     "Netherlands", "Poland", "Portugal", "Romania", "Slovakia", "Slovenia",
     "Spain", "Sweden")

kraje=gapminder %>%
  filter(year=="1987",
         continent=="Europe") %>%
  mutate(system=ifelse(country %in% rwp, "rwp",
                       ifelse(country %in% ue, "ue",
                              "inne")))

RWPG = kraje %>%
  filter(system=="rwp") %>%
  select(-year, -country, -system)
RWPG

## # A tibble: 4 x 4
##   continent lifeExp      pop gdpPercap
##   <fct>      <dbl>    <int>    <dbl>
## 1 Europe      71.3  8971958      8240.
## 2 Europe      69.6 10612740     12986.
## 3 Europe      71.0 37740710      9082.
## 4 Europe      69.5 22686371      9696.

UE = kraje %>%
  filter(system=="ue") %>%
  select(-year, -country, -system)
UE

## # A tibble: 16 x 4
##   continent lifeExp      pop gdpPercap
##   <fct>      <dbl>    <int>    <dbl>
## 1 Europe      74.9  7578903     23688.
## 2 Europe      75.4  9870200     22526.
## 3 Europe      71.5  4484310     13823.
## 4 Europe      71.6 10311597     16310.
## 5 Europe      74.8  5127024     25116.
## 6 Europe      74.8  4931729     21141.
## 7 Europe      76.3 55630100     22066.
## 8 Europe      74.8 77718298     24639.
## 9 Europe      76.7  9974490     16121.

```

```
## 10 Europe      74.4 3539900    13873.
## 11 Europe      76.4 56729703   19207.
## 12 Europe      76.8 14665278   23651.
## 13 Europe      74.1 9915289    13039.
## 14 Europe      72.2 1945870    18679.
## 15 Europe      76.9 38880702   15765.
## 16 Europe      77.2 8421403    23587.
```

```
inne = kraje %>%
  filter(system=="inne")%>%
  select(-year, -country, -system)
inne
```

```
## # A tibble: 10 x 4
##   continent lifeExp      pop gdpPercap
##   <fct>      <dbl>    <int>    <dbl>
## 1 Europe      72    3075321    3739.
## 2 Europe     71.1  4338977    4314.
## 3 Europe     77.2   244676   26923.
## 4 Europe     74.9   569473   11733.
## 5 Europe     75.9  4186147   31541.
## 6 Europe     71.2  9230783   15871.
## 7 Europe     71.1  5199318   12037.
## 8 Europe     77.4  6649942   30282.
## 9 Europe     63.1 52881328    5089.
## 10 Europe    75.0 56981620   21665.
```

```
summary(RWPG)
```

```
##      continent      lifeExp      pop      gdpPercap
## Africa :0 Min. :69.53 Min. : 8971958 Min. : 8240
## Americas:0 1st Qu.:69.57 1st Qu.:10202544 1st Qu.: 8872
## Asia :0 Median :70.28 Median :16649556 Median : 9389
## Europe :4 Mean :70.36 Mean :20002945 Mean :10001
## Oceania :0 3rd Qu.:71.07 3rd Qu.:26449956 3rd Qu.:10519
## Max. :71.34 Max. :37740710 Max. :12986
```

```
summary(UE)
```

```
##      continent      lifeExp      pop      gdpPercap
## Africa : 0 Min. :71.52 Min. : 1945870 Min. :13039
## Americas: 0 1st Qu.:74.28 1st Qu.: 5078200 1st Qu.:16032
## Asia : 0 Median :74.89 Median : 9892744 Median :20174
## Europe :16 Mean :74.93 Mean :19982800 Mean :19577
## Oceania : 0 3rd Qu.:76.48 3rd Qu.:20719134 3rd Qu.:23603
## Max. :77.19 Max. :77718298 Max. :25116
```

```
summary(inne)
```

```
##      continent      lifeExp      pop      gdpPercap
## Africa : 0 Min. :63.11 Min. : 244676 Min. : 3739
## Americas: 0 1st Qu.:71.16 1st Qu.: 3353028 1st Qu.: 6750
```

```
## Asia : 0 Median :73.43 Median : 4769148 Median :13954
## Europe :10 Mean :72.89 Mean :14335758 Mean :16319
## Oceania : 0 3rd Qu.:75.67 3rd Qu.: 8585573 3rd Qu.:25609
## Max. :77.41 Max. :56981620 Max. :31541
```

Zadanie 3

Wśród krajów o PKB niższym od 5000\$ (rozważyć dane za rok 1977), wyznaczyć po trzy kraje z Afryki, Europy i Azji, w których spodziewana długość życia jest największa.

```
library(gapminder)
library(tidyr)
library(dplyr)

gapminder %>%
  filter(gdpPercap<5000,
         year==1977,
         continent=="Asia" | continent=="Europe" | continent=="Africa") %>%
  group_by(continent) %>%
  top_n(3, lifeExp) %>%
  arrange(continent, desc(lifeExp)) %>%
  select(-year, -pop, -gdpPercap)

## # A tibble: 9 x 3
## # Groups:   continent [3]
##   country          continent lifeExp
##   <fct>           <fct>      <dbl>
## 1 Reunion         Africa        67.1
## 2 Mauritius       Africa        64.9
## 3 Tunisia         Africa        59.8
## 4 Korea, Dem. Rep. Asia         67.2
## 5 Sri Lanka       Asia         65.9
## 6 Malaysia        Asia         65.3
## 7 Bosnia and Herzegovina Europe        69.9
## 8 Albania         Europe        68.9
## 9 Turkey          Europe        59.5
```

Krajem w którym najdłużej żyje się w Afryce jest Reunion, w Europie: Bośnia i Hercegowina, a w Azji: Korea Północna

Zadanie 4

Stworzyć funkcję, która na podstawie długości boków trójkąta odpowiada na pytanie o to, czy trójkąt jest ostrokątny, prostokątny lub rozwartokątny.

```
trojkat <- function(a, b, c)
{
  #ustawienie najdluzszego boku
  boki=c(a, b, c)
  c=max(boki)
```



```

if(a!=b & b!=c & a!=c | a==b | a==c | b==c)
{
  c=max(boki)
  boki = boki[! boki %in% c(c)] #usuwa max bok z listy boków
}

a=boki[1]
b=boki[2]

#wyznacznice kata
cosinus=(c^2-a^2-b^2)/(-2*a*b)
alfa= 180 * acos(cosinus) / pi #zamiana na stopnie
alfa
if(alfa<90)
{
  print("trójkąt ostrokątny")
}
else if(alfa==90)
{
  print("trójkąt prostokątny")
}
else
{
  print("trójkąt rozwartokątny")
}
}
trojkat(15,10,11)

## [1] "trójkąt rozwartokątny"

trojkat(3,4,5)

## [1] "trójkąt prostokątny"

trojkat(5,5,7)

## [1] "trójkąt ostrokątny"

```

Zadanie 5

Stworzyć funkcję, która na podstawie długości boków trójkąta, wyznacza jego pole. Zadbaj o to, by funkcja wyświetliła komunikat, jeśli podane długości boków nie tworzą trójkąta.

```

trojkat2 = function(a,b,c)
{
  #ustawienie najdluzszego boku
  boki=c(a, b, c)
  if(a!=b & b!=c & a!=c | a==b & a<c | a==c & a<b | b==c & b<a) #kazda inna/
dwa male takie same i jeden wiekszy
  {
    c=max(boki)

```

```

    boki = boki[! boki %in% c(c)] #usuwa najdłuższy bok z listy boków
  }
  else if (a==b & a>c | a==c & a>b | b==c & b>a) #dwa takie same duże boki i
jeden krótszy
  {
    print("Podane boki nie tworzą trójkąta.")
    return()
  }
  a=boki[1]
  b=boki[2]

  if(a+b <= c)
  {
    print("Podane boki nie tworzą trójkąta.")
    return()
  }
  else
  {
    p=(a+b+c)/2
    pole=sqrt(p*(p-a)*(p-b)*(p-c))
    pole
  }
}
trojkat2(5,4,3)
## [1] 6
trojkat2(4,4,4)
## [1] 6.928203
trojkat2(1,2,3)
## [1] "Podane boki nie tworzą trójkąta."
## NULL
trojkat2(5,5,3)
## [1] "Podane boki nie tworzą trójkąta."
## NULL

```