

Język R - zadania projektowe

Ola Jaglińska

Zadanie 1

Proszę dla kredytu hipotecznego z ratą stałą udzielonego na kwotę K , przy rocznej stopie oprocentowania r , zaciągniętego na okres n miesięcy wyznaczyć: Przykładowe dane: kwota kredytu = 100 000; roczna stopa oprocentowania = 2%; okres = 36 miesięcy.

1. wysokość oprocentowanie miesięcznego

```
K <- 100000
r <- 0.02
n <- 36
q <- 1 + (r/12)
cat('wysokość oprocentowania miesięcznego wyniesie:', q)

## wysokość oprocentowania miesięcznego wyniesie: 1.001667
```

2. wysokość raty miesięcznej

```
R <- K * q**n * ((q-1)/(q**n-1))
cat('wysokość raty miesięcznej', R)
```

```
## wysokość raty miesięcznej 2864.258
```

3. całkowitą kwotę do spłaty

```
F <- R * n
cat('całkowitą kwotę do spłaty', F)
```

```
## całkowitą kwotę do spłaty 103113.3
```

Zadanie 2

Proszę dla kredytu hipotecznego z ratą malejącą udzielonego na kwotę K , przy rocznej stopie oprocentowania r , zaciągniętego na okres n miesięcy wyznaczyć: Przykładowe dane: kwota kredytu = 100 000; roczna stopa oprocentowania = 2%; okres = 36 miesięcy.

1. wysokość części kapitałowej raty

```
K <- 100000
r <- 0.02
n <- 36
R_0 <- K/n
cat('wysokość części kapitałowej raty:', R_0)
```

```
## wysokość części kapitałowej raty: 2777.778
```

2. wysokość części odsetkowej raty i -tej

```

czesc_odsetkowa_raty_nr <- function(i){
  n = 36
  Ri_1 <- ((K-(i-1)*R_0)*r)/12
  cat('Wysokość części odsetkowej raty nr', i, ':', Ri_1, '\n ')
}
rata_4 <- czesc_odsetkowa_raty_nr(4)

```

```

## Wysokość części odsetkowej raty nr 4 : 152.7778
##

```

```

cat('Wysokość części odsetkowej dla raty 4:', rata_4)

```

```

## Wysokość części odsetkowej dla raty 4:

```

3. wysokość raty i-tej

```

wysokosc_raty_nr <- function(i) {
  Ri_1 <- ((K-(i-1)*R_0)*r)/12
  Ri <- R_0 + Ri_1
  cat('Wysokość raty nr', i, ':', Ri, '\n ')
}
rata_4 <- wysokosc_raty_nr(4)

```

```

## Wysokość raty nr 4 : 2930.556
##

```

```

print(rata_4)

```

```

## NULL

```

4. całkowitą kwotę do spłaty

```

calkowita_kwota <- sum(wysokosc_raty_nr(i[1:36])) print(calkowita_kwota)

```

Jaka była najniższa, średnia i najwyższa wartość raty?

```

najwyzsza_Ri <- max(wysokosc_raty_nr(n)) najnizsza_Ri <- min(wysokosc_raty_nr(n)) srednia <-
mean(wysokosc_raty_nr(n))

```

Zadanie 3

W pliku wig_changes.rds znajduje się wektor uporządkowanych chronologicznie wartości tekstowych + oraz - reprezentujących dni w których • (+) wartość indeksu WIG wzrosła względem wartości z dnia poprzedniego, • (-) wartość indeksu WIG zmalała względem wartości z dnia poprzedniego. Bazując na tym wektorze wyznacz następującą macierz:

```

wig_changes <- readRDS('./wig_changes.rds')
saveRDS(wig_changes, file = './wig_changes.rds')
i3 <- 1:length(wig_changes)
as.numeric(TRUE)

```

```

## [1] 1

```

```

as.numeric(FALSE)

```

```

## [1] 0

```

```

pp <- (sum((wig_changes[i3] == '+' & wig_changes[i3+1] == '+'), na.rm = TRUE))/length(wig_changes)
pm <- (sum((wig_changes[i3] == '+' & wig_changes[i3+1] == '-'), na.rm = TRUE))/length(wig_changes)
mp <- (sum((wig_changes[i3] == '-' & wig_changes[i3+1] == '+'), na.rm = TRUE))/length(wig_changes)

```

```
mm <- (sum((wig_changes[i3] == '-' & wig_changes[i3+1] == '-'), na.rm = TRUE))/length(wig_changes)
kombinacje <- c(pp, pm, mp, mm)
macierz_WIG <- matrix(kombinacje, 2, 2)
print(macierz_WIG)
```

```
##           [,1]      [,2]
## [1,] 0.2846425 0.2384015
## [2,] 0.2382484 0.2385546
```

Podnieś utworzoną macierz do potęgi 3.

```
macierz_WIG3 <- macierz_WIG**3
print(macierz_WIG3)
```

```
##           [,1]      [,2]
## [1,] 0.02306211 0.01354961
## [2,] 0.01352352 0.01357573
```

Zadanie 4a W ramach pewnego ubezpieczenia N klientów płaci składkę wysokości K w zamian za możliwość uzyskania kwoty F jeżeli nastąpi zdarzenie. Zbuduj symulację tego ubezpieczenia na okres T miesięcy zgodnie z poniższym algorytmem. 1. Przyjmij $\alpha = 1$. 2. Wyznacz rezerwę na wypłaty: $S_t = \alpha$, $\alpha = 1$, $\alpha - 1 + \alpha$, $\alpha > 1$. 3. Wyznacz liczbę wypłat: $\alpha = \# : 2 - 1 \ 0.9999$, $\alpha = 2$. 4. Wypłać odszkodowania: $\alpha = \alpha -$. 5. Sprawdź płynność: $\alpha = 0$. 1. Jeżeli spełnione, to zmodyfikuj liczbę ubezpieczonych do $N = N + n - o - a$, gdzie n to losowa liczba z przedziału od 0 do 100 nowych klientów, natomiast o to losowa liczba z przedziału od 0 do 90 klientów rezygnujących. 2. Jeżeli nie spełnione, to firma zbankrutowała. Zatrzymaj algorytm przed czasem. 6. Przyjmij $\alpha = \alpha + 1$. 7. Jeżeli α , to przejdź do 2, w przeciwnym przypadku KONIEC.

Zadanie 5

Plik age.rds zawiera dane dotyczące wieku klientów pewnego banku. Przeanalizuj te dane pod kątem odpowiedzi na następujące pytania.

```
age <- readRDS('./age.rds')
saveRDS(age, file = './age.rds')
```

1. Jaki wiek ma najmłodszy i najstarszy klient?

```
najmlodszy <- min(age, na.rm = TRUE)
najstarszy <- max(age, na.rm = TRUE)
cat(najmlodszy, '\n', najstarszy, sep = '')
```

```
## 16
## 86
```

2. Jaki jest przeciętny wiek klientów banku?

```
sredni_wiek = mean(age, na.rm = TRUE)
round(sredni_wiek, 3)
```

```
## [1] 44.555
```

3. Jak bardzo zróżnicowani są klienci banku pod względem wieku?

```
roznica_wieku <- najstarszy - najmlodszy
cat('Różnica wieku wynosi:', roznica_wieku)
```

```
## Różnica wieku wynosi: 70
```

4. Ilu klientów banku jest niepełnoletnich? Jaki to procent całości?

```
niepelnoletni <- sum(as.numeric(age < 18), na.rm = TRUE)
print(niepelnoletni)
```

```
## [1] 33
```

5. Ilu klientów banku jest w wieku 30-50 lat? Jaki to procent całości?

```
wiek_30_50 <- sum(as.numeric(age >= 30 & age <= 50 ), na.rm = TRUE)
procent_30_50 <- (wiek_30_50*100/length(age))
print(procent_30_50)
```

```
## [1] 65.36
```

6. Ilu klientów nie podało swojego wieku? Jaki to procent całości?

```
brak_wieku <- sum(is.na(age))/length(age)*100
cat(brak_wieku, '%', sep = '')
```

```
## 0.53%
```

7. Ile klientów bank posiada w segmentach wiekowych [16,17], [18,24], [25,34], [35,44], [45,64], [65,Inf]? Jaki to procent całości?

```
przedzialy <- cut(age, c(16,17, 18,24, 25,34, 35,44, 45,64, 65, Inf))
segmenty_wiekowe <- table(przedzialy)
segmenty_procent <- ((sum(segmenty_wiekowe[c(1,3,5,7,9,11)]))/length(age))*100
cat('Procent całości:', ' ', segmenty_procent, '%', sep = '')
```

```
## Procent całości: 91.72%
```

Zadanie 6

Wykonanie poniższego kodu spowoduje skonstruowanie prostego modelu linowego zapisanego w postaci listy w obiekcie model. Wykonaj ten kod, a następnie: 1. przyjrzyj się strukturze obiektu model

```
c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14) -> ctl
c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69) -> trt
gl(2, 10, 20, labels = c("Ctl","Trt")) -> group
c(ctl, trt) -> weight
summary(lm(weight ~ group)) -> model
print(model)
```

```
##
## Call:
## lm(formula = weight ~ group)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.0710 -0.4938  0.0685  0.2462  1.3690
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.0320     0.2202  22.850 9.55e-15 ***
## groupTrt       -0.3710     0.3114  -1.191   0.249
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.6964 on 18 degrees of freedom
## Multiple R-squared:  0.07308,    Adjusted R-squared:  0.02158
## F-statistic: 1.419 on 1 and 18 DF,  p-value: 0.249
```

2. znajdź i wyświetl współczynniki modelu (coefficients),

```
print(model$coefficients)
```

```
##              Estimate Std. Error  t value    Pr(>|t|)
## (Intercept)    5.032   0.2202177  22.85012 9.547128e-15
## groupTrt      -0.371   0.3114349  -1.19126 2.490232e-01
```

3. Znajdź i wyświetl wartości resztowe modelu (residuals),

```
print(model$residuals)
```

```
##      1      2      3      4      5      6      7      8      9     10     11
## -0.862  0.548  0.148  1.078 -0.532 -0.422  0.138 -0.502  0.298  0.108  0.149
##      12     13     14     15     16     17     18     19     20
## -0.491 -0.251 -1.071  1.209 -0.831  1.369  0.229 -0.341  0.029
```

4. znajdź i wyświetl wartość dopasowanego R2 (adj.r.squared).

```
print(model$adj.r.squared)
```

```
## [1] 0.02158191
```

Zadanie 7

Załaduj plik `ugly_diamonds.csv` do R w postaci poprawnej ramki danych, tzn. ramki danych, która spełnia poniższą specyfikację: 'data.frame': 10 obs. of 10 variables: \$ carat : num 0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 \$ cut : chr "Ideal" "Premium" "Good" "Premium" ... \$ color : chr "E" "E" "E" "I" ... \$ clarity: chr "SI2" "SI1" "VS1" "VS2" ... \$ depth : num 61.5 59.8 56.9 62.4 NA 62.8 62.3 61.9 65.1 59.4 \$ table : int 55 61 65 58 58 57 57 55 61 61 \$ price : int 326 326 327 334 335 336 336 337 337 338 \$ x : num 3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 \$ y : num 3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 \$ z : num 2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39

```
read.table(
  './ugly_diamonds.csv',
  header = TRUE, sep = '%', quote = '"', dec = ',',
  stringsAsFactors = FALSE
) -> ugly_diamonds
str(ugly_diamonds)
```

```
## 'data.frame':   10 obs. of  10 variables:
## $ carat   : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23
## $ cut     : chr  "Ideal" "Premium" "Good" "Premium" ...
## $ color   : chr  "E" "E" "E" "I" ...
## $ clarity: chr  "SI2" "SI1" "VS1" "VS2" ...
## $ depth   : chr  "61,5" "59,8" "56,9" "62,4" ...
## $ table   : int   55 61 65 58 58 57 57 55 61 61
## $ price   : int   326 326 327 334 335 336 336 337 337 338
## $ x       : num   3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4
## $ y       : num   3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05
## $ z       : num   2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39
```

Zadanie 9 Plik `albums.csv` zawiera następujące dane dotyczące albumów muzycznych: • `artist_id` – identyfikator artysty, • `album_title` – tytuł albumu, • `genre` – gatunek muzyczny, • `year_of_pub` – rok

publikacji, • num_of_tracks – liczba piosnek na płycie, • num_of_sales – liczba sprzedanych płyt, • rolling_stone_critic – ocena Rolling Stone Magazine, • mtv_critic – ocena MTV, • music_maniac_critic – ocena Music Maniac. Bazując na zdobytej dotychczas wiedzy przeprowadź prostą analizę tej próbki

```
albums <- read.csv('./albums.csv', stringsAsFactors = FALSE)
max_sales <- max(albums$num_of_sales)
max_sales_name <- albums$album_title[albums$num_of_sales == max_sales]
cat('Największa sprzedaż płyt:', max_sales, 'Nazwa albumu:', max_sales_name, '\n')
```

Największa sprzedaż płyt: 999994 Nazwa albumu: Decent Waterbuck

```
min_sales <- min(albums$num_of_sales)
min_sales_name <- albums$album_title[albums$num_of_sales == min_sales]
cat('Najniższa sprzedaż płyt:', min_sales, 'Nazwa albumu:', min_sales_name)
```

Najniższa sprzedaż płyt: 1009 Nazwa albumu: Nervous Aggressive Trick Arabic The Shock Sin

```
max_rolling <- max(albums$rolling_stone_critic)
max_rolling_percent <- (sum(as.numeric(albums$rolling_stone_critic == max_rolling)))/length(albums$roll
cat('Największa ocena Rolling Stone Magazine:', max_rolling, '\nProcent całości:', max_rolling_percent,
```

Największa ocena Rolling Stone Magazine: 5

Procent całości: 10.059 %

```
min_rolling <- min(albums$rolling_stone_critic)
min_rolling_percent <- (sum(as.numeric(albums$rolling_stone_critic == min_rolling)))/length(albums$roll
cat('Najniższa ocena Rolling Stone Magazine:', min_rolling, '\nProcent całości:', min_rolling_percent,
```

Najniższa ocena Rolling Stone Magazine: 0.5

Procent całości: 10.044 %

```
genre_percent <- function(genre) {
  result <- (sum(as.numeric(albums$genre == genre)))/length(albums$genre)*100
  return(cat('Procentowa zawartość dla gatunku:', genre, result, "% \n"))
}
genre_percent("Rock")
```

Procentowa zawartość dla gatunku: Rock 3.804 %

```
genre_percent("Rap")
```

Procentowa zawartość dla gatunku: Rap 5.788 %

```
genre_percent("Jazz")
```

Procentowa zawartość dla gatunku: Jazz 1.975 %

```
genre_percent("Punk")
```

Procentowa zawartość dla gatunku: Punk 3.787 %

```
genre_percent("Disco")
```

Procentowa zawartość dla gatunku: Disco 0 %