

SPRAWOZDANIE

Zajęcia: Analiza Procesów Ucznia

Prowadzący: prof. dr hab. Vasyl Marstenyuk

Laboratorium: Zajecie 1

Data: 24.02.2020 r.

Temat: „Podstawy języka R”

Wariant 12

Szymon Pęczalski
Informatyka II stopień,
stacjonarne,
1 semestr,
Gr. A

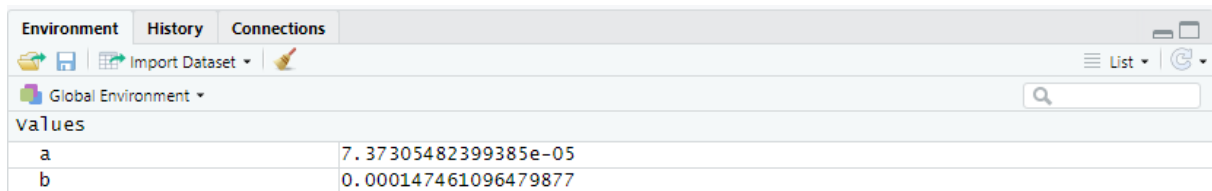
<https://github.com/SzPeczalski/APU>

a) Do zmiennej a podstaw wartość wyrażenia $12/\exp(12)$. Do zmiennej b podstaw podwójną wartość zmiennej a. Wywołaj funkcję sprawdzającą, która z wartości zmiennych jest większa.

```
a <- 12/exp(12)
```

```
b <- a*2
```

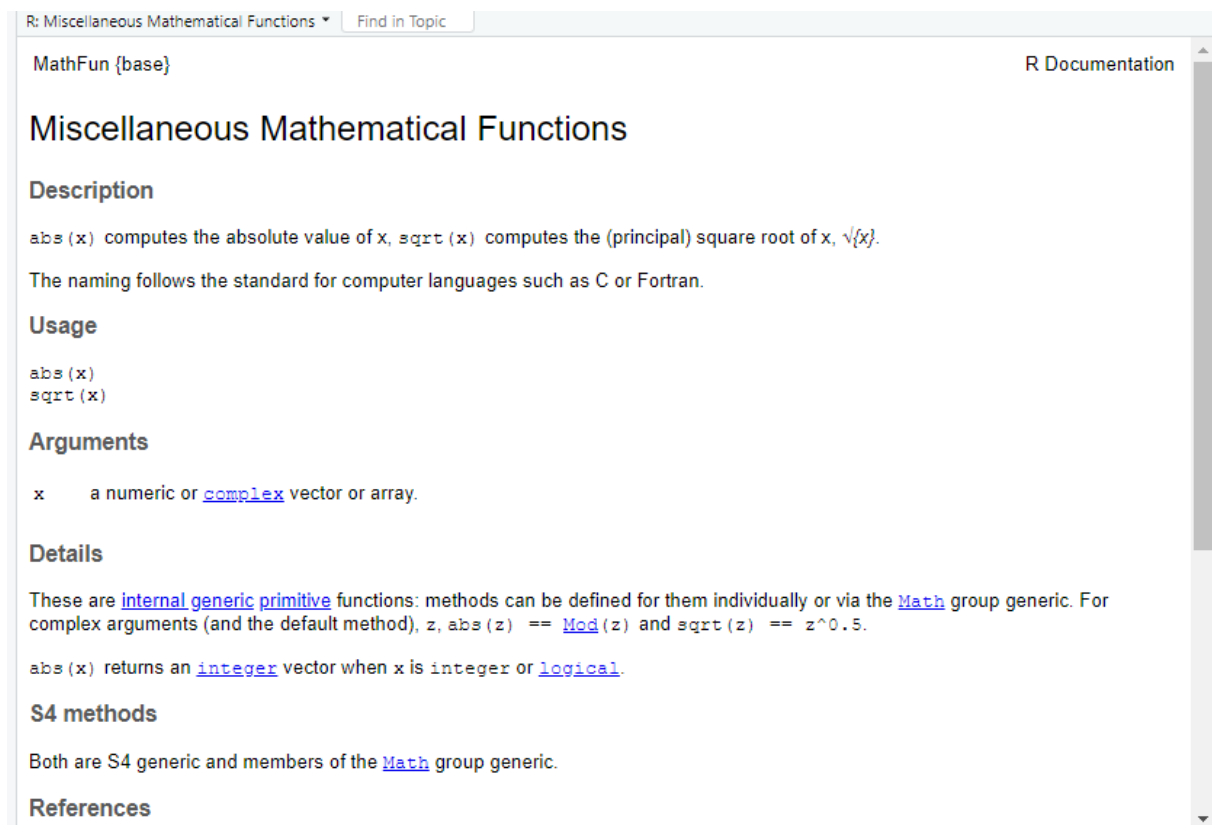
```
max(a,b)
```



Environment		History	Connections
Global Environment			
Values			
a	7.37305482399385e-05		
b	0.000147461096479877		

b) Uruchom i poczytaj dokumentację dla funkcji `sqrt()`.

```
?sqrt()
```

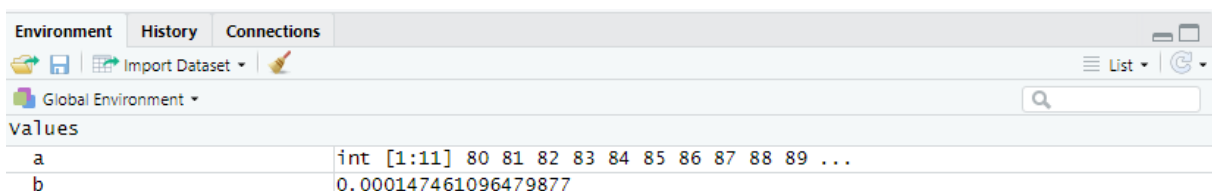


R: Miscellaneous Mathematical Functions	Find in Topic
MathFun {base}	R Documentation
<h2>Miscellaneous Mathematical Functions</h2>	
<h3>Description</h3>	
<code>abs(x)</code> computes the absolute value of <code>x</code> , <code>sqrt(x)</code> computes the (principal) square root of <code>x</code> , \sqrt{x} .	
The naming follows the standard for computer languages such as C or Fortran.	
<h3>Usage</h3>	
<code>abs(x)</code> <code>sqrt(x)</code>	
<h3>Arguments</h3>	
<code>x</code> a numeric or complex vector or array.	
<h3>Details</h3>	
These are internal generic primitive functions: methods can be defined for them individually or via the Math group generic. For complex arguments (and the default method), <code>z, abs(z) == Mod(z)</code> and <code>sqrt(z) == z^0.5</code> .	
<code>abs(x)</code> returns an integer vector when <code>x</code> is integer or logical .	
<h3>S4 methods</h3>	
Both are S4 generic and members of the Math group generic.	
<h3>References</h3>	

c) Stwórz wektor a zawierający liczby od 80 do 90. Policz sumę kwadratów liczb zawartych w wektorze.

```
a <- c(80:90)
```

```
sum(a^2)
```



Environment		History	Connections
Global Environment			
Values			
a	int [1:11] 80 81 82 83 84 85 86 87 88 89 ...		
b	0.000147461096479877		

```
> a <- c(80:90)
> sum(a^2)
[1] 79585
> |
```

d) Wyświetl wszystkie funkcje zawierające frazę max w swojej nazwie

```
??max
```

```
apropos("max", mode = "function")
```

```
> apropos("max", mode = "function")
[1] "cummax"      "max"          "max.col"      "maxEmptyRect" "mem.maxNSize" "mem.maxVSize" "pmax"        "pmax.int"
[9] "promax"      "varimax"      "which.max"
```

e) Ustaw dowolny katalog roboczy. Następnie stwórz zmienną a zawierającą łańcuch znaków "smartfon Motorola". Zapisz zmienną a z obszaru roboczego do pliku w katalogu roboczym. Następnie usuń zmienną a. Sprawdź wartość zmiennej a (powinno jej brakować). Na końcu wczytaj plik ze zmienną a i sprawdź jej wartość.

```
setwd("D:/Users/User1/Documents/R Studio")
```

```
a <- "smartfon Motorola"
```

values	
a	"smartfon Motorola"
b	0.000147461096479877

```
write(a, file='pkt_e.csv')
```

```
rm(a)
```

values	
b	0.000147461096479877

```
a
```

```
a <- read.csv(file = 'pkt_e.csv', sep = ',')
```

```
a
```

Environment		History	Connections
Global Environment			
Data			
a	0 obs. of 1 variable		
values			
b	0.000147461096479877		

```
> #Zad E
> setwd("D:/Users/User1/Documents/R Studio")
> a <- "smartfon Motorola"
> write(a, file='pkt_e.csv')
> rm(a)
> a
BŁĄD: nie znaleziono obiektu 'a'
> a <- read.csv(file = 'pkt_e.csv', sep = ',')
> a
[1] smartfon.Motorola
<0 wierszy> (lub 'row.names' o zerowej długości)
```

Ten komputer > Dokumenty > R Studio				
Nazwa	Data modyfikacji	Typ	Rozmiar	
pkt_e	25.02.2020 12:45	Plik wartości oddz...	1 KB	

f) Zainstaluj i załaduj pakiet gridExtra, który umożliwia ładną wizualizację danych tabelarycznych. Następnie przy pomocy dokumentacji pakietu znajdź funkcję do wizualizacji danych tabelarycznych. Użyj jej na pierwszych 10 wierszach zbioru danych Titanic.

```
install.packages("gridExtra")
library("gridExtra")
d <- head(Titanic, 10)
g <- tableGrob(d)
grid.arrange(g)
```

Environment

History

Connections

Import Dataset

Global Environment

Data

a

0 obs. of 1 variable

g

List of 20

Values

b

0.000147461096479877

d

num [1:10] 0 0 35 0 0 0 17 0 118 154

Files	Plots	Packages	Help	Viewer
<div>0</div> <div>0</div> <div>35</div> <div>0</div> <div>0</div> <div>0</div> <div>17</div> <div>0</div> <div>118</div> <div>154</div>				

```
> install.packages("gridExtra")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:
https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'D:/Users/User1/Documents/R/win-library/3.6'
(as 'lib' is unspecified)
próbowanie adresu URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/gridExtra_2.3.zip'
Content type 'application/zip' length 1109585 bytes (1.1 MB)
downloaded 1.1 MB
package 'gridExtra' successfully unpacked and MD5 sums checked
The downloaded binary packages are in
d:/Users/User1/AppData/Local/Temp/RtmpMzpru2/downloaded_packages
> library("gridExtra")
> d <- head(Titanic, 10)
> g <- tableGrob(d)
> grid.arrange(g)
```

g) Stwórz wektor zawierający ciąg liczb 400, 392, 384,..., 320.

```
a <- seq(400,320,-8)
```

Environment	History	Connections
<div> <div>Import Dataset</div> <div>List</div> </div>		
Global Environment		
Data		
g	List of 20	
values		
a	num [1:11] 400 392 384 376 368 360 352 344 336 328 ...	
b	0.000147461096479877	
d	num [1:10] 0 0 35 0 0 0 17 0 118 154	

h) Stwórz wektora a z liczbami od 60 do 40 oraz wektor b z liczbami od 40 do 50. Utwórz nowy wektory d będący połączeniem wektora b i a (w takiej kolejności). Wyświetl go.

```
a <- c(60:40)
```

```
b <- c(40:50)
```

```
d <- c(b, a)
```

```
d
```

```
> #Zad H
> a <- c(60:40)
> b <- c(40:50)
> d <- c(b, a)
> d
[1] 40 41 42 43 44 45 46 47 48 49 50 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40
```

i) Stwórz wektor nazwa zawierający nazwy 10 smartfonów Motorola. Potem stwórz wektory wyświetlacz, pamięć_RAM, pamięć_wbudowana, aparat_foto, cena, liczba_opinii zawierające kolejno dane 10 smartfonów. Następnie stwórz ramkę danych smartfony złożoną z wektorów wyświetlacz, pamięć_RAM, pamięć_wbudowana, aparat_foto, cena, liczba_opinii. Wylicz średnią cenę smartfonów.

```
nazwy <- c('Moto G8 Power','Moto G7 Power','Moto E6 Plus','One Action','E5 Play','One Vision','One Zoom','Moto Z','Moto X4','Moto Z3 Play')
```

```
wyswietlacz <- c('6.4"', '6.2"', '6.1"', '6.34"', '5.3"', '6.3"', '6.4"', '5.5"', '5.2"', '6"')
```

```
pamięć_RAM <- c('4 gb','4 gb','4 gb','4 gb','1 gb','4 gb','4 gb','4 gb','3 gb','4 gb')
```

```
pamięć_wbudowana <- c('64 gb','64 gb','64 gb','128 gb','16 gb','128 gb','128 gb','32 gb','32 gb','64 gb')
```

```
aparat_foto <- c('16 Mpix','12 Mpix','13 Mpix','12 Mpix','8 Mpix','48 Mpix','48 Mpix','16 Mpix','12 Mpix','12 Mpix')
```

```
cena <- c(999,698,599,799,299,969,1599,549,649,950)
```

```
liczba_opinii <- c(7,492,19,24,22,168,25,66,7,5)
```

```
smartfony <- data.frame(nazwy,
                        wyswietlacz,pamięć_RAM,pamięć_wbudowana,aparat_foto,cena,liczba_opinii)
mean(smartfony$cena)
```

Environment

History

Connections

Import Dataset

Global Environment

List

Data

g

List of 20

smartfony

10 obs. of 7 variables

values

a

int [1:21] 60 59 58 57 56 55 54 53 52 51 ...

aparat_foto

chr [1:10] "16 Mpix" "12 Mpix" "13 Mpix" "12 Mpix" "8 Mpix" "48 Mpix" ...

b

int [1:11] 40 41 42 43 44 45 46 47 48 49 ...

cena

num [1:10] 999 698 599 799 299 ...

d

int [1:32] 40 41 42 43 44 45 46 47 48 49 ...

liczba_opinii

num [1:10] 7 492 19 24 22 168 25 66 7 5

nazwy

chr [1:10] "Moto G8 Power" "Moto G7 Power" "Moto E6 Plus" "One Action"...

pamię_RAM

chr [1:10] "4 gb" "4 gb" "4 gb" "4 gb" "1 gb" "4 gb" "4 gb" "4 gb" "3 ...

pamię_wbudowana

chr [1:10] "64 gb" "64 gb" "64 gb" "128 gb" "16 gb" "128 gb" "128 gb" ...

wyswietlacz

chr [1:10] "6.4\" "6.2\" "6.1\" "6.34\" "5.3\" "6.3\" "6.4\" "5...

```
> #Zad 1
> nazwy <- c('Moto G8 Power', 'Moto G7 Power', 'Moto E6 Plus', 'One Action', 'E5 Play', 'One Vision', 'One Zoom', 'Moto Z', 'Moto X4', 'Moto Z3 Play')
> wyswietlacz <- c('6.4"', '6.2"', '6.1"', '6.34"', '5.3"', '6.3"', '6.4"', '5.5"', '5.2"', '6"')
> pamie_ram <- c('4 gb', '4 gb', '4 gb', '4 gb', '1 gb', '4 gb', '4 gb', '4 gb', '3 gb', '4 gb')
> pamie_wbudowana <- c('64 gb', '64 gb', '64 gb', '128 gb', '16 gb', '128 gb', '128 gb', '32 gb', '32 gb', '64 gb')
> aparat_foto <- c('16 mpix', '12 mpix', '13 mpix', '12 mpix', '8 mpix', '48 mpix', '48 mpix', '16 mpix', '12 mpix', '12 mpix')
> cena <- c(999, 698, 599, 799, 299, 969, 1599, 549, 649, 950)
> liczba_opinii <- c(7, 492, 19, 24, 22, 168, 25, 66, 7, 5)
> smartfony <- data.frame(nazwy, wyswietlacz, pamie_ram, pamie_wbudowana, aparat_foto, cena, liczba_opinii)
> mean(smartfony$cena)
[1] 811
```

```
newRow <- data.frame(nazwy = 'Moto C+', wyswietlacz = '5"', pamieć_RAM = '1 gb', pamieć_wbudowana
                     = '16 gb', aparat_foto = '8 Mpix', cena = 299, liczba_opinii = 50)
smartfony <- rbind(smartfony, newRow)
mean(smartfony$cena)
```

The screenshot shows the RStudio Environment pane. At the top, there are tabs for 'Environment', 'History', and 'Connections'. Below the tabs, there are icons for saving, loading, and importing datasets. The main area of the pane is titled 'Global Environment' and contains a search bar. Below the search bar, there is a table listing the objects in the environment:

Object Name	Description
g	List of 20
newRow	1 obs. of 7 variables
smartfony	10 obs. of 7 variables

```
> #Zad 3
> newRow <- data.frame(nazwy = 'Moto C+', wyswietlacz = '5"', pamiec_RAM = '1 gb', pamiec_wbudowana = '16 gb', aparat_foto = '8 Mpi
x', cena = 299, liczba_opinii = 50)
> smartfony <- rbind(smartfony, newRow)
> mean(smartfony$cena)
[1] 764.4545
```

k) Korzystając z ramki danych smartfony dodaj nową kolumnę określając ocenę klientów. Wpisz do kolumny odpowiednio oceny w skali od 0 do 5 krok 0.5. Dodana kolumna powinna się automatycznie przekonwertować do cech jakościowych (tzw. factors). Wylicz średnią ceny każdej oceny.

```
smartfony$ocena <- c('5','5','5','5','5','5','5','5','5','5','5')
aggregate(smartfony$cena, list(smartfony$ocena), mean)
```

```
> #Zad k
> smartfony$ocena <- c('5','5','5','5','5','5','5','5','5','5','5')
> aggregate(smartfony$cena, list(smartfony$ocena), mean)
  Group.1      x
1      5 764.4545
```

l) Do ramki danych smartfony dodaj kolejne 4 smartfony. Narysuj na wykresie słupkowym liczebność reprezentantów każdej z ocen klientów (pakiet plotrix).

```
newRow <- data.frame(nazwy = 'Moto Z Play', wyswietlacz = '5.5"', pamieć_RAM = '3 gb',
  pamieć_wbudowana = '32 gb', aparat_foto = '16 Mpix', cena = 1275, liczba_opinii = 70, ocena = '5')
smartfony <- rbind(smartfony, newRow)
```

```
newRow <- data.frame(nazwy = 'Moto Z2 Play', wyswietlacz = '5.5"', pamieć_RAM = '4 gb',
  pamieć_wbudowana = '64 gb', aparat_foto = '12 Mpix', cena = 1020, liczba_opinii = 9, ocena = '5')
smartfony <- rbind(smartfony, newRow)
```

```
newRow <- data.frame(nazwy = 'Moto G6 Play', wyswietlacz = '5.7"', pamieć_RAM = '3 gb',
  pamieć_wbudowana = '32 gb', aparat_foto = '13 Mpix', cena = 499, liczba_opinii = 581, ocena = '5')
smartfony <- rbind(smartfony, newRow)
```

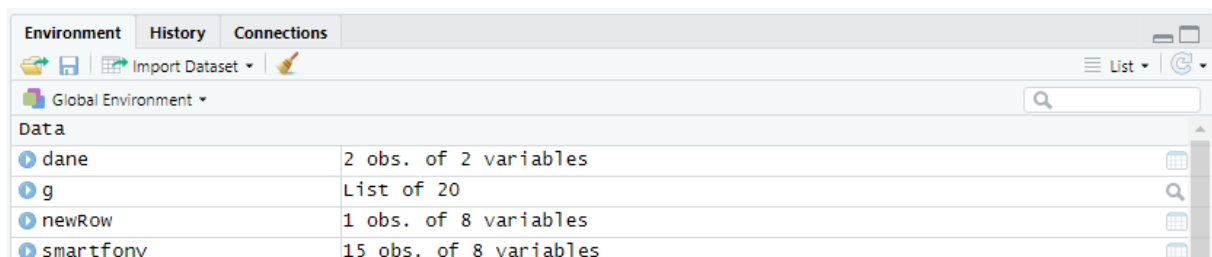
```
newRow <- data.frame(nazwy = 'Moto E6 Plus', wyswietlacz = '6.1"', pamieć_RAM = '4 gb',
  pamieć_wbudowana = '64 gb', aparat_foto = '13 Mpix', cena = 599, liczba_opinii = 26, ocena = '4.5')
smartfony <- rbind(smartfony, newRow)
```

```
install.packages("plotrix")
```

```
library(plotrix)
```

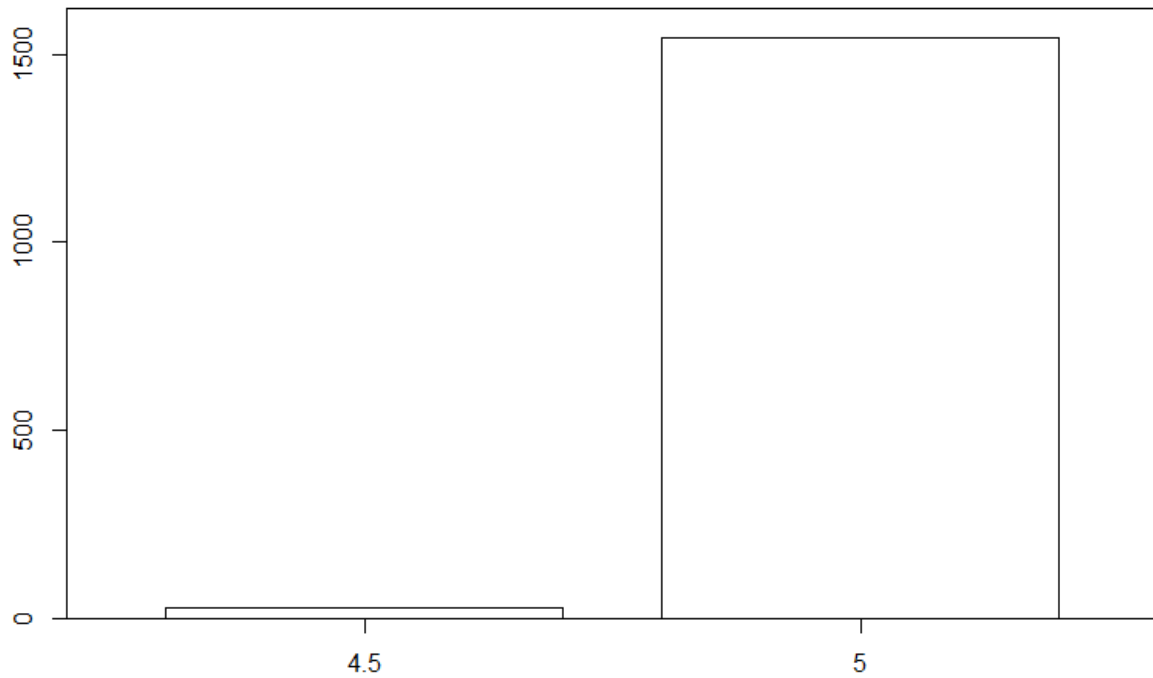
```
dane <- aggregate(smartfony$liczba_opinii, list(smartfony$ocena), sum)
```

```
barp(dane[,2], names.arg = dane[,1], main = 'Liczebność reprezentantów kazdej z ocen')
```



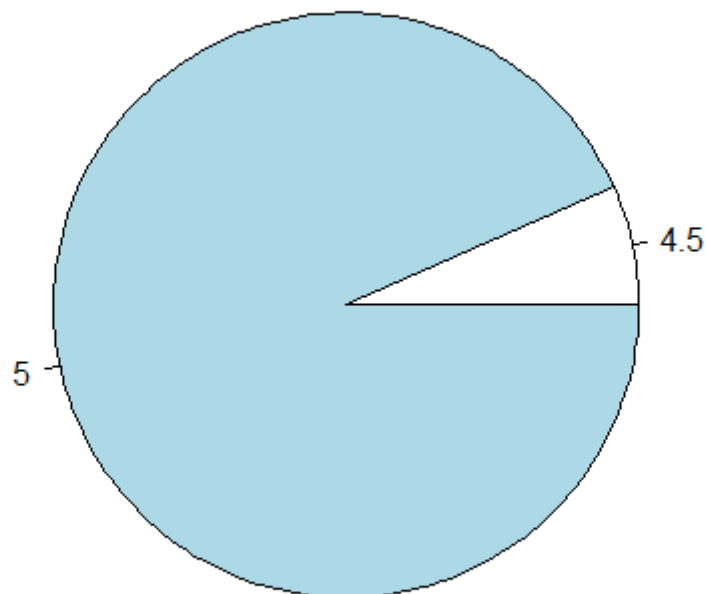
Object	Type	Observations / Variables
dane	data.frame	2 obs. of 2 variables
g	list	List of 20
newRow	data.frame	1 obs. of 8 variables
smartfony	data.frame	15 obs. of 8 variables

Liczebność reprezentantów kazdej z ocen



m) Wykorzystując ramkę danych smartfony pokaż procentowy udział każdej oceny przy pomocy wykresu kołowego oraz wachlarzowego (pakiet plotrix).

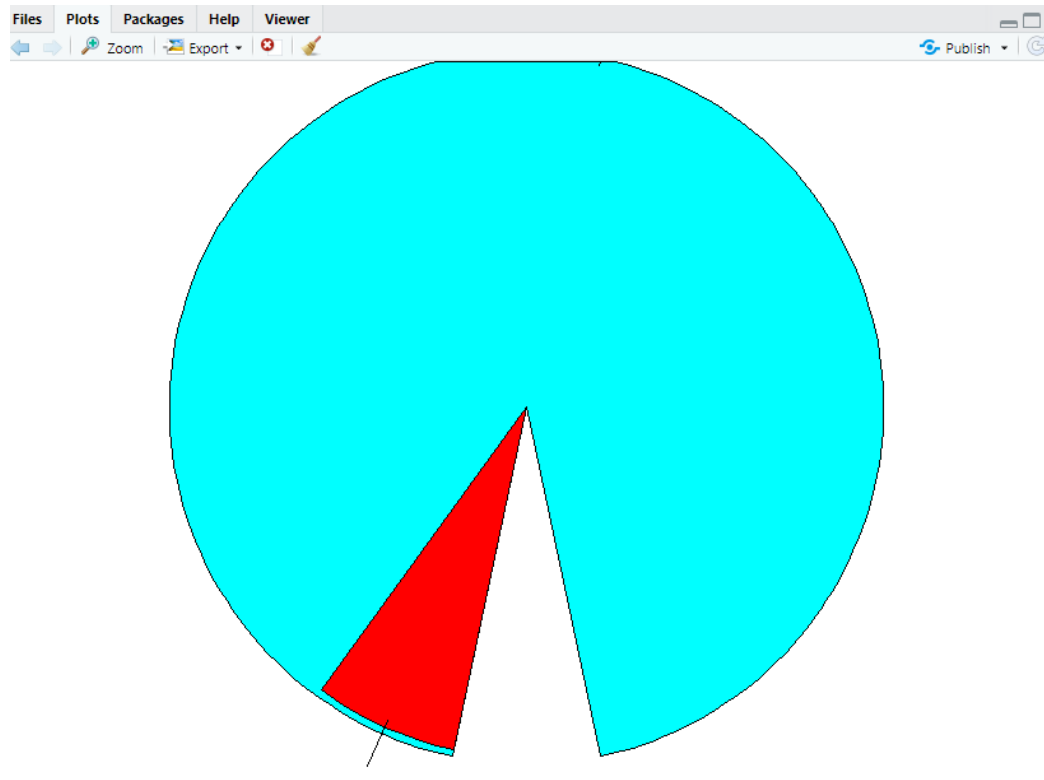
```
#####WYKRES KOŁOWY#####  
percentage <- table(smartfony$ocena)/length(smartfony$ocena)  
pie(percentage)
```



#####WYKRES WACHLARZOWY#####

```
percentage <- table(smartfony$ocena)/length(smartfony$ocena)
```

```
fan.plot(percentage, labels = names(percentage), main = "Procentowy udział oceny")
```



n) Do ramki danych smartfony dodaj nową kolumnę status_opinii z wartościami: "nie ma", "mniej 50 opinii", "50-100 opinii", "więcej 100 opinii" w zależności od liczby opinii. Zamień dodaną kolumnę na cechy jakościowe. Następnie przy pomocy wykresu kołowego wyrysuj procentowy udział smartfonów o konkretnym statusie opinii.

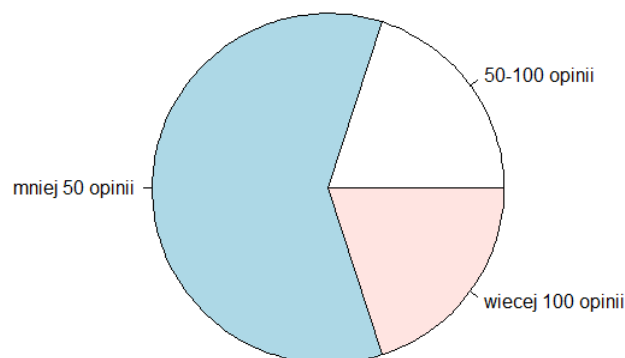
```
n_column <- ifelse(smartfony$liczba_opinii>100,'wiecej 100 opinii',
ifelse(smartfony$liczba_opinii>=50, '50-100 opinii', ifelse(smartfony$liczba_opinii>0, 'mniej
50 opinii', 'nie ma')))
```

```
smartfony['status_opinii'] <- factor(n_column)
```

#####WYKRES KOŁOWY#####

```
percentage <-table(smartfony$status_opinii)/length(smartfony$status_opinii)
```

```
pie(percentage)
```



	nazwy	wyswietlacz	pamie_RAM	pamie_wbudowana	aparat_foto	cena	liczba_opinii	ocena	status_opinii
1	Moto G8 Power	6.4"	4 gb	64 gb	16 Mpix	999	7	5	mniej 50 opinii
2	Moto G7 Power	6.2"	4 gb	64 gb	12 Mpix	698	492	5	wiecej 100 opinii
3	Moto E6 Plus	6.1"	4 gb	64 gb	13 Mpix	599	19	5	mniej 50 opinii
4	One Action	6.34"	4 gb	128 gb	12 Mpix	799	24	5	mniej 50 opinii
5	E5 Play	5.3"	1 gb	16 gb	8 Mpix	299	22	5	mniej 50 opinii
6	One Vision	6.3"	4 gb	128 gb	48 Mpix	969	168	5	wiecej 100 opinii
7	One Zoom	6.4"	4 gb	128 gb	48 Mpix	1599	25	5	mniej 50 opinii
8	Moto Z	5.5"	4 gb	32 gb	16 Mpix	549	66	5	50-100 opinii
9	Moto X4	5.2"	3 gb	32 gb	12 Mpix	649	7	5	mniej 50 opinii
10	Moto Z3 Play	6"	4 gb	64 gb	12 Mpix	950	5	5	mniej 50 opinii
11	Moto C+	5"	1 gb	16 gb	8 Mpix	299	50	5	50-100 opinii
12	Moto Z Play	5.5"	3 gb	32 gb	16 Mpix	1275	70	5	50-100 opinii
13	Moto Z2 Play	5.5"	4 gb	64 gb	12 Mpix	1020	9	5	mniej 50 opinii
14	Moto G6 Play	5.7"	3 gb	32 gb	13 Mpix	499	581	5	wiecej 100 opinii
15	Moto E6 Plus	6.1"	4 gb	64 gb	13 Mpix	599	26	4.5	mniej 50 opinii

o) Wykorzystując ramkę danych smartfony stwórz zdanie o każdym z smartfonów postaci: nazwa + " ma ocenę klientów " + ocena_klientów + " bo ma liczbę opinii" + liczba_opinii. Plus oznacza konkatencję łańcuchów i wartości.

```
for (i in 1:length(smartfony$nazwy)){
  print(paste(smartfony$nazwy[i], 'ma ocenę klientów', smartfony$ocena[i], 'bo ma
  liczbę opinii', smartfony$liczba_opinii[i]))
}

> #Zad 0
> for (i in 1:length(smartfony$nazwy)){
+   print(paste(smartfony$nazwy[i], 'ma ocenę klientów', smartfony$ocena[i], 'bo ma liczbę opinii', smartfony$liczba_opinii[i]))
+ }
[1] "Moto G8 Power ma ocenę klientów 5 bo ma liczbę opinii 7"
[1] "Moto G7 Power ma ocenę klientów 5 bo ma liczbę opinii 492"
[1] "Moto E6 Plus ma ocenę klientów 5 bo ma liczbę opinii 19"
[1] "One Action ma ocenę klientów 5 bo ma liczbę opinii 24"
[1] "E5 Play ma ocenę klientów 5 bo ma liczbę opinii 22"
[1] "One Vision ma ocenę klientów 5 bo ma liczbę opinii 168"
[1] "One Zoom ma ocenę klientów 5 bo ma liczbę opinii 25"
[1] "Moto Z ma ocenę klientów 5 bo ma liczbę opinii 66"
[1] "Moto X4 ma ocenę klientów 5 bo ma liczbę opinii 7"
[1] "Moto Z3 Play ma ocenę klientów 5 bo ma liczbę opinii 5"
[1] "Moto C+ ma ocenę klientów 5 bo ma liczbę opinii 50"
[1] "Moto Z Play ma ocenę klientów 5 bo ma liczbę opinii 70"
[1] "Moto Z2 Play ma ocenę klientów 5 bo ma liczbę opinii 9"
[1] "Moto G6 Play ma ocenę klientów 5 bo ma liczbę opinii 581"
[1] "Moto E6 Plus ma ocenę klientów 4.5 bo ma liczbę opinii 26"
```

p) Zachować ramkę danych w pliku .csv. Załadować ramkę danych z pliku .csv

```
write.csv(smartfony, 'smartfony.csv')
dane <- read.csv('smartfony.csv')
```

