Podstawy R i RStudio

Tomasz Olczyk

Dlaczego programatycznie w Rmarkdown

Reprodukowalność i literate programming

Wszystkie wyniki (tabele, wykresy itp) powinny byc procedualnie i reprodukowalnie generowane z zapisanego kodu, zapisane w prostym formacie opartym na tekście. Możliwość reprodukowania własnej pacy jest kluczowa w nauce, ale także ze względów czysto pragmatycznych. Osobą, która będzie najczęściej reprodukować Twoje wyniki jesteś Ty za sześć miesięcy.

Celem jest maksymalnie programatyczne tworzenie grafiki. Staramy się nie używać sposobów które nie są programatycznie reprodukowalne.

Zadania:

1. pisanie kodu
2. analiza wyników 3 notatki

Dlatego używamy RMakdown. Jest to czysto tekstowy dokument, któy umożliwia robienie tych wszystkich rzeczy, w sposób reprodukowalny i zapisany w możliwie odporny sposób.

Projekt elementy podstawowe

File > New project

File > New File > Rmarkdown

znak `#` oznacza komentarz wewnętrz segmentu kodu w Rmakdown odnosi się do poziomu nagłówka

skróty klawiszowe Ctrl+Alt+ I - nowy segment kodu w Windowsie Cmd+Alt + I = nowy segment kodu w Mac os

RStudio podstawwowe informacje

- Panele
- Pliki
- Skrypty
- RMarkdown
- Konsola

Instalacja pakietów

Pakiet R jest kolekcją funkcji, danych i dokumentacji, która rozszerza możliwości bazowego R

Poniższy segment jest nazywany code chunk. Pliki R markdown składają się z nagłówka, segmentów tekstu i specjalnie oznaczonych segmentów kodu. Skrypty składają się z kodu i komentarzy poprzedzonych #. W markdown "domyślną" treścią jest tekst "do czytania", kod jest dodatkiem, dlatego znajduje się w specjalnie zaznaczonych segmentach. Raportom w R markdown poświęcimy oddzielne zajęcia.

W skryptach jest na odwrót, dlatego tekst trzeba poprzedzać # żeby program nie uznał go za kod.

Biblioteki instalujmy funkcją install.packages() której argumentem jest nazwa biblioteki w cudzysłowie. Bibliotekę wystarczy zainstalować raz.

```
# kod w tym segmencie instaluje bibliotekę tidyverse

#install.packages("tidyverse")

# hash przed kodem sprawia, że znaki po nim są traktowane jak komentarz, kod nie będzie wykonany
```

Wczytywanie pakietów (bibliotek)

Pakiety wczytujemy funkcją library(). Biblioteki dobrze jest wczytywać na początku analizy. Biblioteka powinna być wczytana przed użyciem funkcji w niej zawartej. Można użyć funkcji bez wczytywania używając składni biblioteka::funkcja

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
#install.packages("gapminder")
library(gapminder)
```

Obiekty i funkcje

W R wszystko jest obiektem.

Nazwy obiektów muszą rozpoczynać się od litery i mogą zawierać tylko litery, cyfry znak _ oraz . .

Wszystko ma nazwę:

zmienne, dane, funkcje. Do nadawania nazw nie wolno używać m.in FALSE, TRUE, Inf, for, else, break, function. Lepiej nie używać we własnych nazwach nazw funkcji, szczególnie z pakiety podstawowego jak c() - konkatentacja, q(), mean(), var(), range() i podobnych

Wszystko jest obiektem:

```r
# przypisuje kolekcję liczb do nazwy  moje liczby
# operator przypisania <- można uzyskać skrótem Alt - na Windowsie, Opt - na mac

moje_liczby  <- c(1, 2, 3, 5, 25) # funkcja c() tworzy wektor czyli uporządkowaną kolekcję elementów je
```

Operator przypisania skróty: -mac os: Opt + - -Win: Alt + -

Po przypisaniu i wykonaniu kodu obiekt pojawi się w przestrzeni roboczej: okno environment

Operator przypisania łączy obiekt z nazwą

Działania wykonuje się za pomocą funkcji:

Kiedy chcemy żeby funkcja wykonała jakieś działanie wywołujemy ją. Funkcje mają nawiasy na końcu swojej nazwy. Funkcje wykonują działania na podstawie wartości przyjmowanych przez ich argumenty

nazwa_funkcji(argument1 = , argument2 = , argument2 =)

Argumenty zawsze oddzielone są przecinkami

```r
# funkcja licząca średnią
mean()
```

```
## Error in mean.default(): argument "x" is missing, with no default
```

Komunikat o błędzie: **Error in mean.default() : argument "x" is missing, with no default** W tym wypadku infomuje o tym, że wartość argumentu x nie została podana Ogólnie jeśli wartość argumentu nie jest liczbą, obiektem istniejącym w przestrzeni roboczej albo wartością logiczną to jest podawana w cudzysłowie (por instalowanie biblioteki versus wczytywanie biblioteki)

```r
# najczęście nie trzeba podawać nazw argumentów, wystarczy podanie ich wartości w odpowiedniej kolejnoś
mean(moje_liczby)
```

```
## [1] 7.2
```

Rezultaty działania funkcji można przypisać do obiektów

```r
podsumowanie <- summary(moje_liczby)
podsumowanie # wyświetla obiekt w konsoli
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0     2.0     3.0     7.2     5.0    25.0
```

Funkcje zawarte są w pakietach (bibliotekach)

Żeby użyć funkcji musimy mieć zainstalowaną i wczytaną bibliotekę (package) z której pochodzi funkcja. Wczytywanie funkcją library można pominąć wskazując bibliotekę i funkcję w ten sposób `nazwa_biblioteki::nazwa_funkcji`

W R działamy na obiektach.Obiektami manipulujemy przekazując informacje o nich do funkcji. Funkcje wykonują operacje na obiektach i zwracają wyniki.

np.

```r
# zwraca tabelę liczności elementów

table(moje_liczby)
```

```
## moje_liczby
##  1  2  3  5 25
##  1  1  1  1  1
```

```r
# zwraca odchylenie standardowe

sd(moje_liczby)
```

```
## [1] 10.05982
```

```r
# mnoży każdy element przez 5

moje_liczby * 5
```

```
## [1]   5  10  15  25 125
```

```r
# dodaje  5 do każdego elementu

moje_liczby + 5
```

```
## [1]  6  7  8 10 30
```

```r
# operacja wektoryzowana, kolejne elementy jednego wektora dodane do kolejnych elementów drugiego wykto
moje_liczby + moje_liczby
```

```
## [1]  2  4  6 10 50
```

Każdy obiekt ma klasę, którą możemy spawdzic funkcją `class()`

```r
class(moje_liczby)
```

```
## [1] "numeric"
```

```r
podsumowanie <- summary(moje_liczby)
class(podsumowanie)
```

```
## [1] "summaryDefault" "table"
```

```r
class(summary)
```

```
## [1] "function"
```

Operacje wykonywane na obiektach mogą skutkować zmianą ich klasy:

```
nowy_wektor <- c(moje_liczby, "jabłko")
class(nowy_wektor)
```

```
## [1] "character"
```

Zbiory danych zaimportowane do też są obiektami. W R jest kilka typów obiektów służących do przechowywania danych ale najbadziej typowym jest ramka danych: data frame. Ramka danych to prostokątna tabela złożona z wierszy (obserwacji) i kolumn (zmiennych). Kolumny w ramce danych mogą być różnej klasy.

Liczby

```
class(1)
```

```
## [1] "numeric"
```

Znaki

```
class("1")
```

```
## [1] "character"
```

Logiczne

```
two <- 2 # strzałka jest operatorem przypisania, przypisuje obiekt do nazwy

Two <- 2 #wielkie litery mają znaczenie
```

Operatory

```
logiczne <- two == Two # podwójny znak = oznacza w R równa się

logiczne
```

```
## [1] TRUE
```

```
nierówna_się <- two != Two

nierówna_się
```

```
## [1] FALSE
```

Obiekt i operator przypisania. Zapisuje obiekt “data”, zawierający dane gapminder

```
data <- (gapminder)
```

Wgląd w dane w ramce

```
str(data)
```

```
## tibble [1,704 x 6] (S3: tbl_df/tbl/data.frame)
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ year     : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num [1:1704] 28.8 30.3 32 34 36.1 ...
##  $ pop      : int [1:1704] 8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 163
##  $ gdpPercap: num [1:1704] 779 821 853 836 740 ...
```

Wyświetla 6 pierwszych wierszy

```
head(data)
```

```
## # A tibble: 6 x 6
##   country     continent  year lifeExp      pop gdpPercap
##   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan Asia       1952    28.8  8425333      779.
## 2 Afghanistan Asia       1957    30.3  9240934      821.
## 3 Afghanistan Asia       1962    32.0 10267083      853.
## 4 Afghanistan Asia       1967    34.0 11537966      836.
## 5 Afghanistan Asia       1972    36.1 13079460      740.
## 6 Afghanistan Asia       1977    38.4 14880372      786.
```

Wyświetla ramkę danych w lewym panelu

```
View(data)
```

Wyświetla ostatnich sześć wierszy

```
tail(data)
```

```
## # A tibble: 6 x 6
##   country  continent  year lifeExp      pop gdpPercap
##   <fct>    <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Zimbabwe Africa     1982    60.4  7636524      789.
## 2 Zimbabwe Africa     1987    62.4  9216418      706.
## 3 Zimbabwe Africa     1992    60.4 10704340      693.
## 4 Zimbabwe Africa     1997    46.8 11404948      792.
## 5 Zimbabwe Africa     2002    40.0 11926563      672.
## 6 Zimbabwe Africa     2007    43.5 12311143      470.
```

Funkcja glimpse z pakietu dplyr. Nazwa funkcji poprzedzona dwukropkiem i nazwą biblioteki informuje/ustala, z której biblioteki ma być zastosowana funkcja o danej nazwie. Przydatne, gdy jest wiele funkcji o tych samych nazwach z różnych bibliotek,

```
dplyr::glimpse(data)
```

```
## Rows: 1,704
## Columns: 6
## $ country   <fct> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan", ~
## $ continent <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, ~
## $ year      <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 1997, ~
## $ lifeExp   <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854, 40.8~
## $ pop       <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372, 12~
## $ gdpPercap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.1134, ~
```

Wektory

Tworzenie wektorów

```
partie <- c("KO", "PiS", "Lewica", "Konfederacja", "TD")


partie
```

```
## [1] "KO"           "PiS"          "Lewica"       "Konfederacja" "TD"
```

Wektor napisów

```
class(partie)
```

```
## [1] "character"
```

Wektor liczb

```
sondaż <- c(29, 35, 10, 10, 10)
```

```
typeof(sondaż)
```

```
## [1] "double"
```

```
# w języku R klasa double reprezentuje liczby zmiennoprzecinkowe o podwójnej precyzji (double precision
```

```
class(sondaż)
```

```
## [1] "numeric"
```

Indeksowanie

```
wektor <- 3:103
```

```
wektor[0]
```

```
## integer(0)
```

```
wektor[1]
```

```
## [1] 3
```

```
LETTERS
```

```
##  [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S"
## [20] "T" "U" "V" "W" "X" "Y" "Z"
```

```
co_drugi <- seq(from = 1, to = length(LETTERS), by  = 2)

LETTERS[co_drugi]
```

```
##  [1] "A" "C" "E" "G" "I" "K" "M" "O" "Q" "S" "U" "W" "Y"
```

Ramki danych

Matryca z dwóch wektorów

```
matryca <- cbind(partie, sondaż)
matryca
```

```
##      partie         sondaż
## [1,] "KO"           "29"
## [2,] "PiS"          "35"
## [3,] "Lewica"       "10"
## [4,] "Konfederacja" "10"
## [5,] "TD"           "10"
```

```
glimpse(matryca)
```

```
##  chr [1:5, 1:2] "KO" "PiS" "Lewica" "Konfederacja" "TD" "29" "35" "10" "10" ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:2] "partie" "sondaż"
```

```
ramka_danych <- data.frame(partie, sondaż)
```

```
glimpse(ramka_danych)
```

```
## Rows: 5
## Columns: 2
## $ partie <chr> "KO", "PiS", "Lewica", "Konfederacja", "TD"
## $ sondaż <dbl> 29, 35, 10, 10, 10
```

Jak połączyć wektory nierównych długości w ramkę danych: link

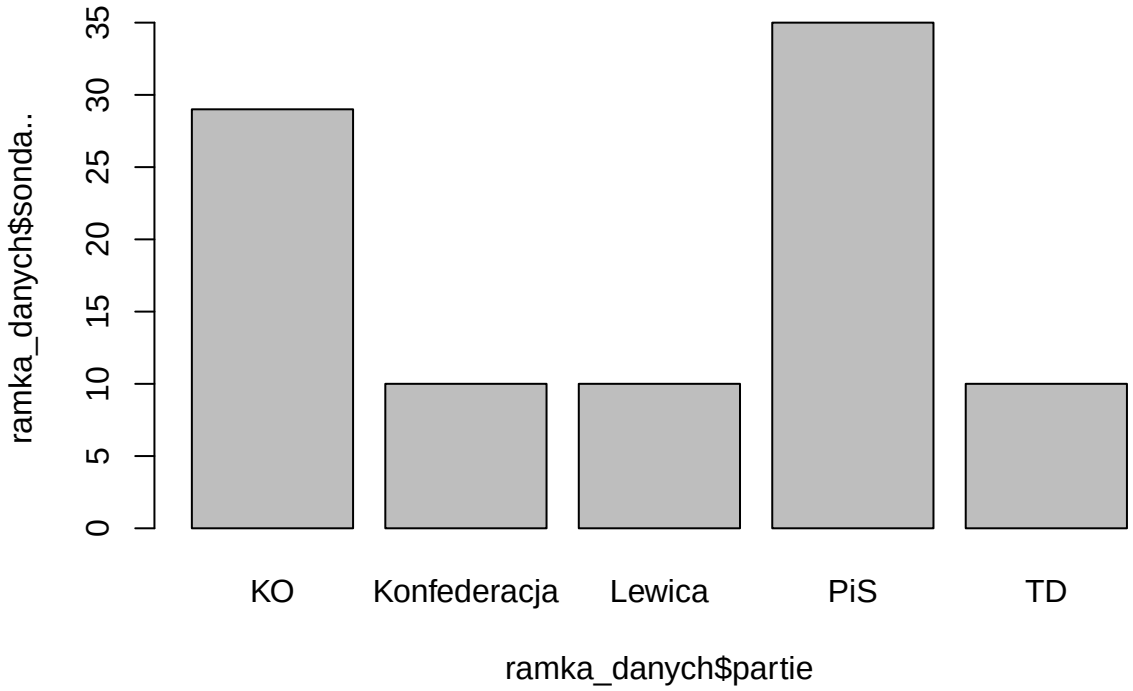
```
str(ramka_danych)
```

```
## 'data.frame':    5 obs. of  2 variables:
##  $ partie: chr  "KO" "PiS" "Lewica" "Konfederacja" ...
##  $ sondaż: num  29 35 10 10 10
```

```
barplot(ramka_danych$sondaż
        ~ ramka_danych$partie)
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'ramka_danych$sondaż' in 'mbcsToSbcs': dot substituted for
## <c5>
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'ramka_danych$sondaż' in 'mbcsToSbcs': dot substituted for
## <bc>
```



Wczytywanie danych

```
?read.delim
```

```
?read.table
```

Funkcje mają zazwyczaj długą listę argumentów, z których większość ma domyślne wartości.

```
read.table(file,
           header = FALSE,
           sep = "",
           quote = "\"'",
           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
           row.names, col.names,
           as.is = !stringsAsFactors,
           na.strings = "NA",
           colClasses = NA,
           nrows = -1,
           skip = 0,
           check.names = TRUE,
           fill = !blank.lines.skip,
           strip.white = FALSE,
           blank.lines.skip = TRUE,
           comment.char = "#",
           allowEscapes = FALSE, flush = FALSE,
           stringsAsFactors = FALSE,
           fileEncoding = "",
           encoding = "unknown",
           text,
           skipNul = FALSE)
```

```
?read.csv()
```

```
read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", ...)

read.csv2(file, header = TRUE, sep = ";", quote = "\"",
          dec = ",", fill = TRUE, comment.char = "", ...)

read.delim(file, header = TRUE, sep = "\t", quote = "\"",
           dec = ".", fill = TRUE, comment.char = "", ...)

read.delim2(file, header = TRUE, sep = "\t", quote = "\"",
            dec = ",", fill = TRUE, comment.char = "", ...)
```

```
?readr::read_delim
```

```
read_delim(
  file,
  delim = NULL,
  quote = "\"",
  escape_backslash = FALSE,
  escape_double = TRUE,
```

```
  col_names = TRUE,
  col_types = NULL,
  col_select = NULL,
  id = NULL,
  locale = default_locale(),
  na = c("", "NA"),
  quoted_na = TRUE,
  comment = "",
  trim_ws = FALSE,
  skip = 0,
  n_max = Inf,
  guess_max = min(1000, n_max),
  name_repair = "unique",
  num_threads = readr_threads(),
  progress = show_progress(),
  show_col_types = should_show_types(),
  skip_empty_rows = TRUE,
  lazy = should_read_lazy()
)

read_csv(
  file,
  col_names = TRUE,
  col_types = NULL,
  col_select = NULL,
  id = NULL,
  locale = default_locale(),
  na = c("", "NA"),
  quoted_na = TRUE,
  quote = "\"",
  comment = "",
  trim_ws = TRUE,
  skip = 0,
  n_max = Inf,
  guess_max = min(1000, n_max),
  name_repair = "unique",
  num_threads = readr_threads(),
  progress = show_progress(),
  show_col_types = should_show_types(),
  skip_empty_rows = TRUE,
  lazy = should_read_lazy()
)

read_csv2(
  file,
  col_names = TRUE,
  col_types = NULL,
  col_select = NULL,
  id = NULL,
  locale = default_locale(),
  na = c("", "NA"),
  quoted_na = TRUE,
  quote = "\"",
```

```
  comment = "",
  trim_ws = TRUE,
  skip = 0,
  n_max = Inf,
  guess_max = min(1000, n_max),
  progress = show_progress(),
  name_repair = "unique",
  num_threads = readr_threads(),
  show_col_types = should_show_types(),
  skip_empty_rows = TRUE,
  lazy = should_read_lazy()
)

read_tsv(
  file,
  col_names = TRUE,
  col_types = NULL,
  col_select = NULL,
  id = NULL,
  locale = default_locale(),
  na = c("", "NA"),
  quoted_na = TRUE,
  quote = "\"",
  comment = "",
  trim_ws = TRUE,
  skip = 0,
  n_max = Inf,
  guess_max = min(1000, n_max),
  progress = show_progress(),
  name_repair = "unique",
  num_threads = readr_threads(),
  show_col_types = should_show_types(),
  skip_empty_rows = TRUE,
  lazy = should_read_lazy()
)
```

```
panteon <- read_csv("https://raw.githubusercontent.com/Tomasz-Olczyk/wizualizacjaR/main/podstawy/panteon
```

```
## Rows: 11341 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): name, countryName, countryCode3, continentName, gender, industry, d...
## dbl (6): LAT, LON, birthyear, L_star, HPI, AverageViews
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Zapisywanie danych

analogicznie funkcjami write*

```
?write.table
```

```
write.table(x, file = "", append = FALSE, quote = TRUE, sep = " ",
            eol = "\n", na = "NA", dec = ".", row.names = TRUE,
            col.names = TRUE, qmethod = c("escape", "double"),
            fileEncoding = "")
```

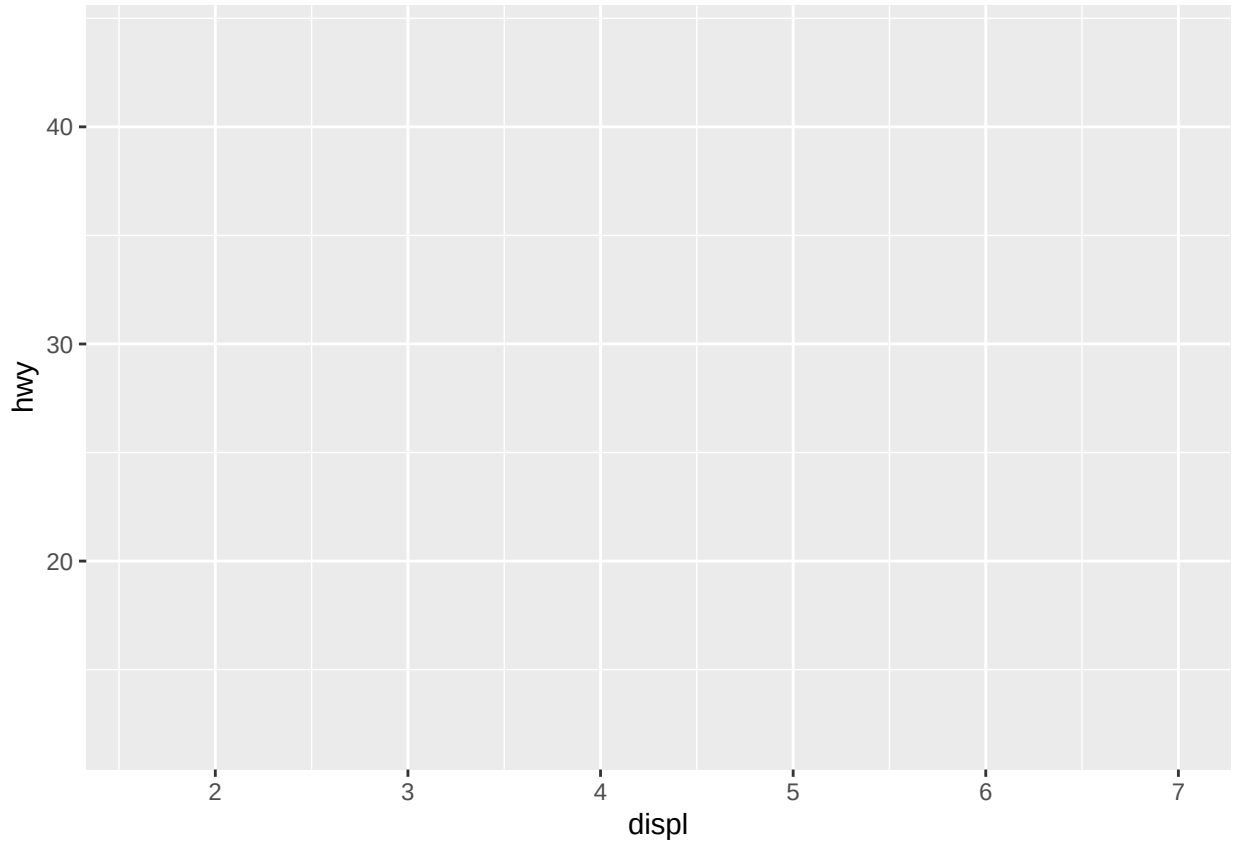
Pozwalajmy sobie na błędy

R zawsze robi to co mu każemy zrobić, ale nie zawsze to, co chcemy żeby zrobił. Inaczej mówiąc R traktuje wszystkie polecenia literalnie.

Będziemy popełniać błędy i będzie ich mnóstwo. Błędy pomagają nauczyć się programowania w R.

Typowe błędy: niedomknięty nawias, niekompletne wyrażanie (+ w konsoli, kursor + Esc) częstybłąd ggplot to + w niewłaściwm miejscy

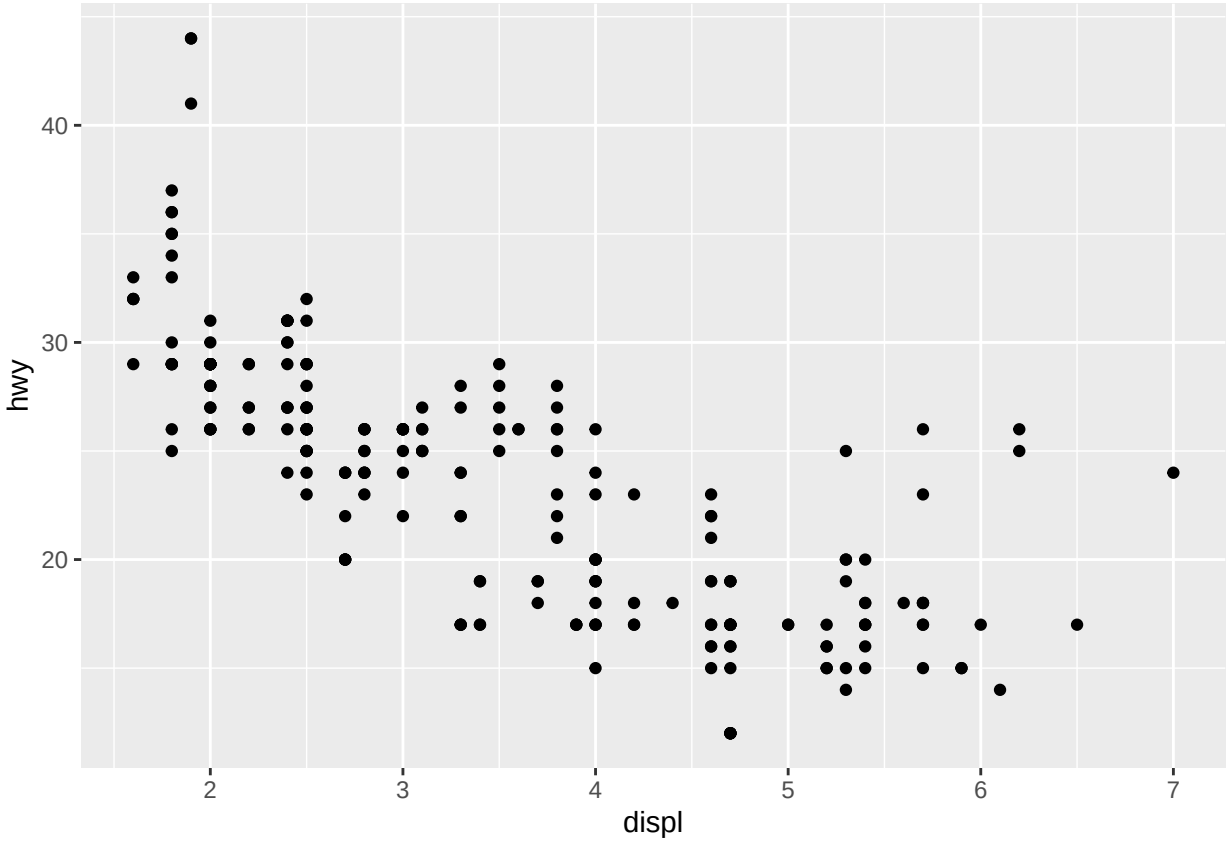
```
ggplot(data = mpg, aes(x = displ, y = hwy))
```




```
+ geom_point()
```

```
## Error:
## ! Cannot use `+` with a single argument.
## i Did you accidentally put `+` on a new line?
```

```
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point()
```



Inne typy plików wejściowych

Praktycznie każde działanie w R można wykonać co najmniej na kilka sposobów za pomocą różnych funkcji różnych pakietów. Dotyczy to zarówno grafiki jak i wczytywania plików.

Do wczytywania plików excela można wykorzystać bibliotekę readxl

```
#install.packages("readxl")
library(readxl)
```

Biblioteka ta ma kilka funkcji wczytujących pliki excelowskie np read_excel która sama ustala czy plik ma rozszerzenie xls czy xlsx ale także funkcje read_xls, i read_xlsx, których możemy użyć jeśli znamy rozszerzenie pliku

```
read_excel(
  path,
  sheet = NULL,
  range = NULL,
  col_names = TRUE,
  col_types = NULL,
  na = "",
```

```
  trim_ws = TRUE,
  skip = 0,
  n_max = Inf,
  guess_max = min(1000, n_max),
  progress = readxl_progress(),
  .name_repair = "unique"
)
```

```
podatki <- read_excel(
  "../podstawy/Chapter14TablesFigures.xlsx", #uwaga musimy być we właściwym katalogu roboczym, żeby ta
  sheet = "TS14.1", #wybieramy arkusz z pliku
  skip = 3 # usuwamy trzy pierwsze wiersze
  )
```

```
## New names:
## * `` -> `...1`
## * `` -> `...6`
## * `` -> `...9`
## * `` -> `...11`
## * `` -> `...14`
```

```
str(podatki)
```

```
## tibble [120 x 15] (S3: tbl_df/tbl/data.frame)
##  $ ...1                                                : chr [1:120] "1900" "1901" "1902" "1903" ...
##  $ U.S.                                                : num [1:120] 0 0 0 0 0 0 0 0 0 0 ...
##  $ U.K.                                                : num [1:120] 0 0 0 0 0 ...
##  $ Germany                                             : num [1:120] 0.03 0.03 0.03 0.03 0.03 0.03 0.03
##  $ France                                              : num [1:120] 0 0 0 0 0 0 0 0 0 0 ...
##  $ ...6                                                : logi [1:120] NA NA NA NA NA NA ...
##  $ U.S.         (top marginal rate on earned income): num [1:120] 0 0 0 0 0 0 0 0 0 0 ...
##  $ U.S.         (top effective rate)                   : num [1:120] 0 0 0 0 0 0 0 0 0 0 ...
##  $ ...9                                                : logi [1:120] NA NA NA NA NA NA ...
##  $ U.K.         (top marginal rate on earned income): num [1:120] 0 0 0 0 0 ...
##  $ ...11                                               : logi [1:120] NA NA NA NA NA NA ...
##  $ France (income tax)                                 : num [1:120] 0 0 0 0 0 0 0 0 0 0 ...
##  $ France (CSG)                                        : num [1:120] 0 0 0 0 0 0 0 0 0 0 ...
##  $ ...14                                               : logi [1:120] NA NA NA NA NA NA ...
##  $ Japan (Saez-Morigushi Table A0)                     : num [1:120] 0.055 0.055 0.055 0.0935 0.2035 ..
```

```
dplyr::glimpse(podatki)
```

```
## Rows: 120
## Columns: 15
## $ ...1                                                <chr> "1900", "1901", "1~
## $ U.S.                                                <dbl> 0.000, 0.000, 0.00~
## $ U.K.                                                <dbl> 0.00000000, 0.0000~
## $ Germany                                             <dbl> 0.03, 0.03, 0.03, ~
## $ France                                              <dbl> 0.00, 0.00, 0.00, ~
## $ ...6                                                <lgl> NA, NA, NA, NA, NA~
## $ `U.S.         (top marginal rate on earned income)` <dbl> 0.000, 0.000, 0.00~
```

```
## $ `U.S.         (top effective rate)`                 <dbl> 0.000, 0.000, 0.00~
## $ ...9                                                <lgl> NA, NA, NA, NA, NA~
## $ `U.K.         (top marginal rate on earned income)` <dbl> 0.00000000, 0.0000~
## $ ...11                                               <lgl> NA, NA, NA, NA, NA~
## $ `France (income tax)`                               <dbl> 0.00, 0.00, 0.00, ~
## $ `France (CSG)`                                      <dbl> 0, 0, 0, 0, 0, 0, ~
## $ ...14                                               <lgl> NA, NA, NA, NA, NA~
## $ `Japan (Saez-Morigushi Table A0)`                   <dbl> 0.0550, 0.0550, 0.~
```

```
podatki$...1 <- as.numeric(podatki$...1)
```

```
## Warning: NAs introduced by coercion
```

```
dplyr::glimpse(podatki)
```

```
## Rows: 120
## Columns: 15
## $ ...1                                                <dbl> 1900, 1901, 1902, ~
## $ U.S.                                                <dbl> 0.000, 0.000, 0.00~
## $ U.K.                                                <dbl> 0.00000000, 0.0000~
## $ Germany                                             <dbl> 0.03, 0.03, 0.03, ~
## $ France                                              <dbl> 0.00, 0.00, 0.00, ~
## $ ...6                                                <lgl> NA, NA, NA, NA, NA~
## $ `U.S.         (top marginal rate on earned income)` <dbl> 0.000, 0.000, 0.00~
## $ `U.S.         (top effective rate)`                 <dbl> 0.000, 0.000, 0.00~
## $ ...9                                                <lgl> NA, NA, NA, NA, NA~
## $ `U.K.         (top marginal rate on earned income)` <dbl> 0.00000000, 0.0000~
## $ ...11                                               <lgl> NA, NA, NA, NA, NA~
## $ `France (income tax)`                               <dbl> 0.00, 0.00, 0.00, ~
## $ `France (CSG)`                                      <dbl> 0, 0, 0, 0, 0, 0, ~
## $ ...14                                               <lgl> NA, NA, NA, NA, NA~
## $ `Japan (Saez-Morigushi Table A0)`                   <dbl> 0.0550, 0.0550, 0.~
```

Dobrą ale nieco wolniejszą biblioteką jest openxlsx

```
#install.packages(openxlsx)
library(openxlsx)
```

```
read.xlsx(
xlsxFile,
sheet,
startRow = 1,
colNames = TRUE,
rowNames = FALSE,
detectDates = FALSE,
skipEmptyRows = TRUE, # pominie puste wiersze
skipEmptyCols = TRUE, # pominie puste kolumny
rows = NULL,
cols = NULL,
check.names = FALSE,
sep.names = ".",
namedRegion = NULL,
```

```
na.strings = "NA",
fillMergedCells = FALSE
)
```

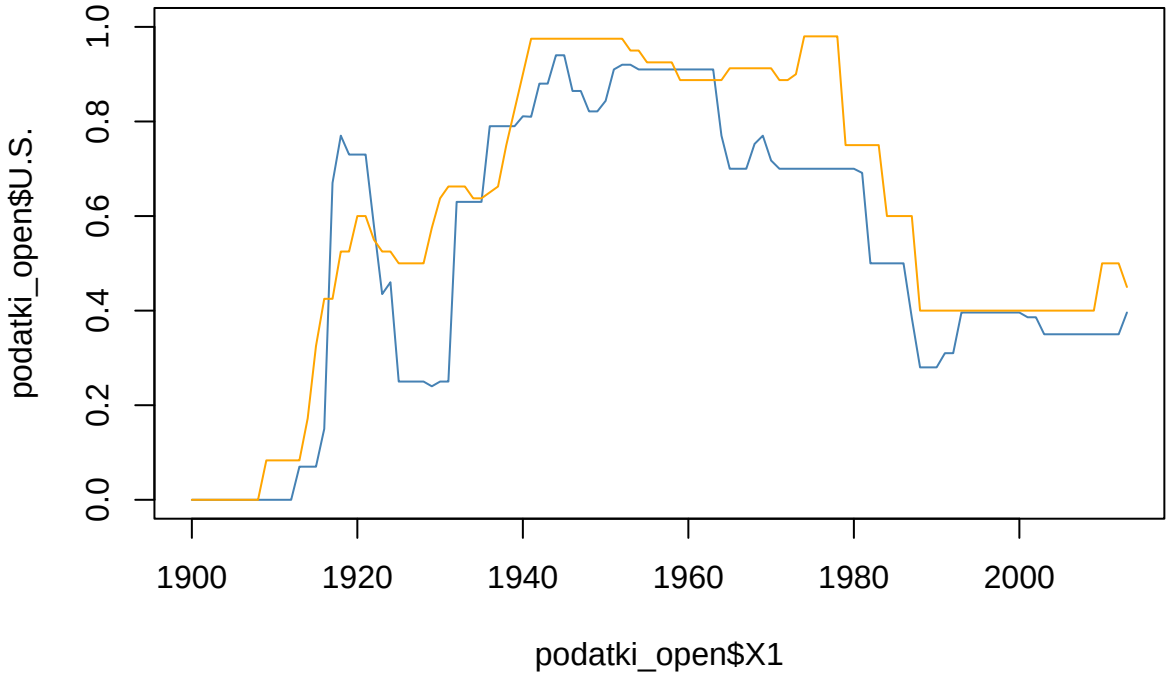
```
podatki_open <- read.xlsx("../podstawy/Chapter14TablesFigures.xlsx",
                          sheet = "TS14.1",
                          startRow = 4)
```

```
plot(y =podatki_open$U.S.,
     x = podatki_open$X1,
     type = "l",
     col = "steelblue",
     ylim = c(0,1))
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): NAs introduced by coercion
```

```
lines(y =podatki_open$U.K.,
     x = podatki_open$X1,
     col = "orange")
```

```
## Warning in xy.coords(x, y): NAs introduced by coercion
```



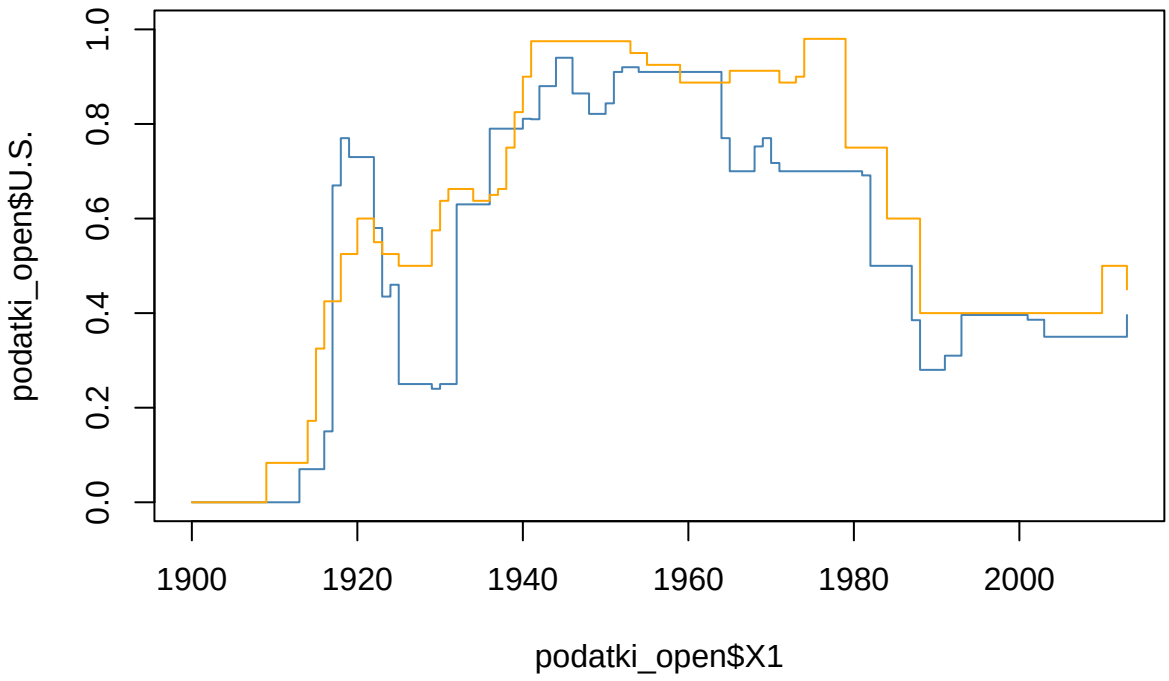
Raczej potrzebujemy schodków bo podatek pozostaje na takim samym poziomie przez jakiś czas

```
plot(y =podatki_open$U.S.,
     x = podatki_open$X1,
     type = "s",
     col = "steelblue",
     ylim = c(0,1))
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): NAs introduced by coercion
```

```
lines(y =podatki_open$U.K.,
     x = podatki_open$X1,
     col = "orange",
     type = "s")
```

```
## Warning in xy.coords(x, y): NAs introduced by coercion
```



Indeksowanie C.D.

```
pięć_kolumn <- panteon[,1:5]
```

```
sześć_wierszy <- panteon[1:6,]
```

```
sześć_wierszy_pięciu_kolumn <- panteon[1:6, 1:5]

sześć_wierszy_pięciu_kolumn
```

```
## # A tibble: 6 x 5
##   name              countryName    countryCode3   LAT     LON
##   <chr>             <chr>          <chr>        <dbl>   <dbl>
## 1 Abraham Lincoln   UNITED STATES  USA           37.6  -85.7
## 2 Aristotle         Greece         GRC           40.3   23.5
## 3 Ayn Rand          Russia         RUS           60.0   30.3
## 4 Andre Agassi      UNITED STATES  USA           36.1 -115.
## 5 Aldous Huxley     UNITED KINGDOM GBR           51.2   -0.61
## 6 Andrei Tarkovsky  Russia         RUS           NA     NA
```

Indeksowanie wektora

```
panteon$countryName[1:6]
```

```
## [1] "UNITED STATES"  "Greece"         "Russia"         "UNITED STATES"
## [5] "UNITED KINGDOM" "Russia"
```

```
piąta_kolumna <- panteon[,5]
```

Czwarty element z piątej kolumny

```
piąta_kolumna[4,]
```

```
## # A tibble: 1 x 1
##     LON
##   <dbl>
## 1 -115.
```

Zakres

```
range(panteon$HPI)
```

```
## [1]  9.879447 31.993795
```

Średnia

```
mean(panteon$HPI)
```

```
## [1] 22.30824
```

summary

```
summary(panteon$domain)
```

```
##    Length     Class      Mode
##     11341 character character
```

```
class(panteon$domain)
```

```
## [1] "character"
```

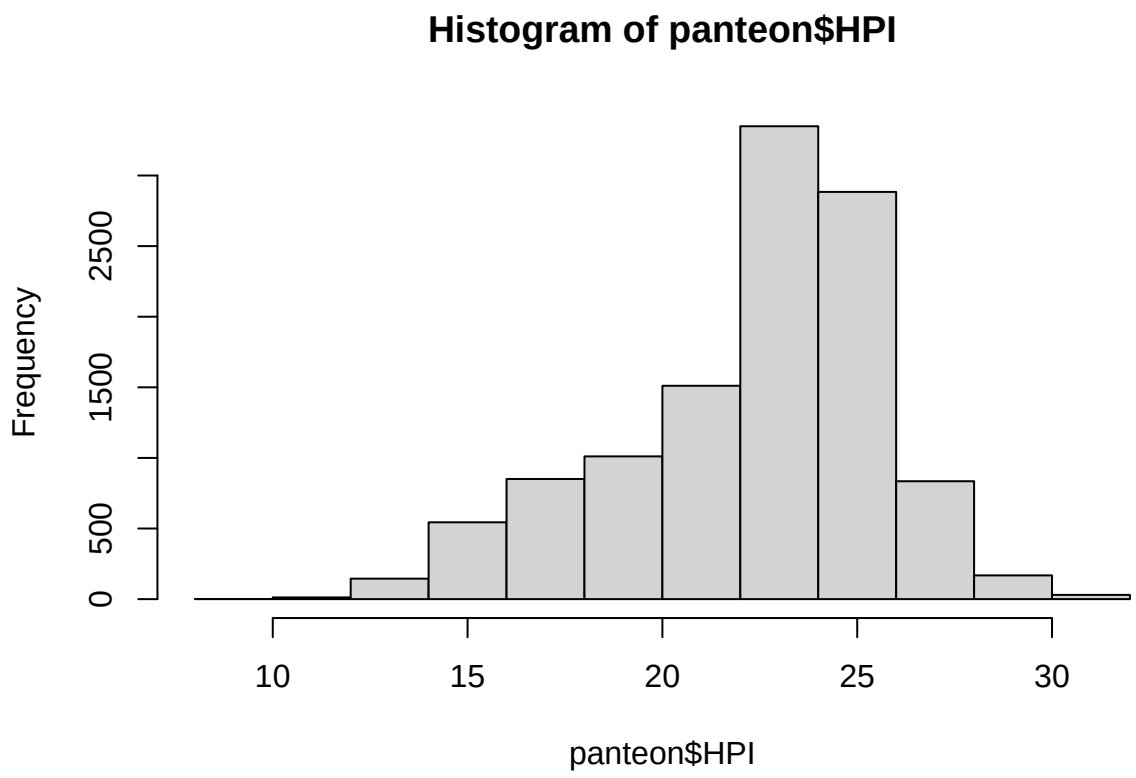
```
table(panteon$gender)
```

```
##
## Female   Male
##   1495   9846
```

Przykłady wykresów w bazowym R

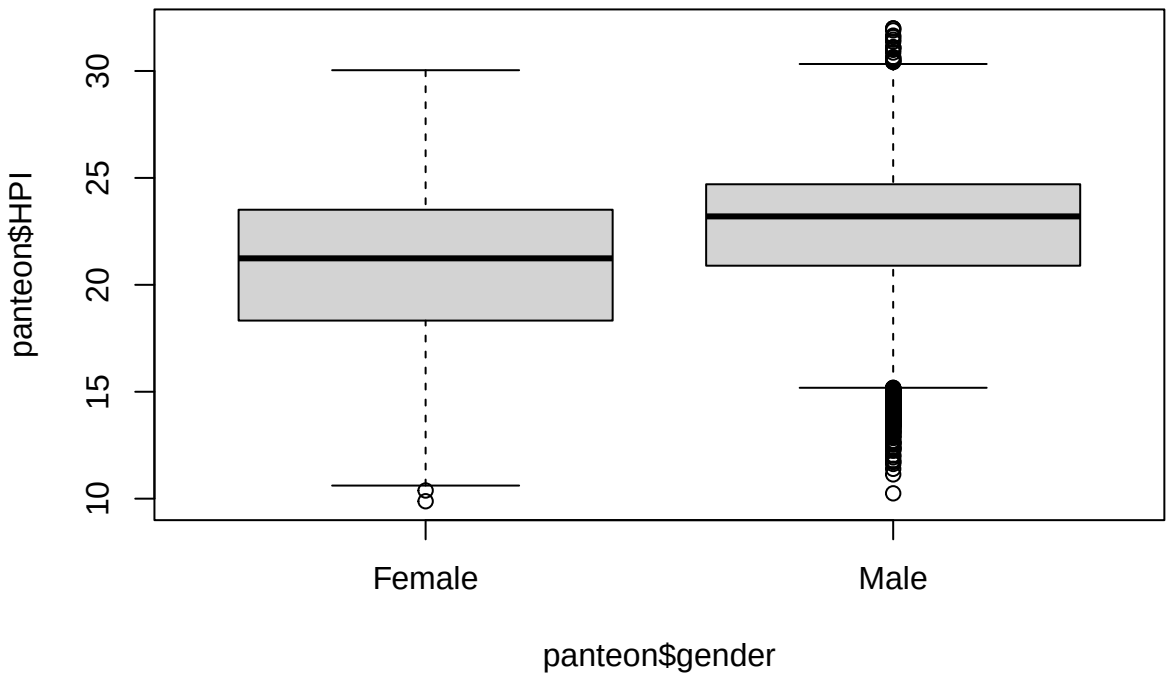
histogram

```
hist(panteon$HPI)
```



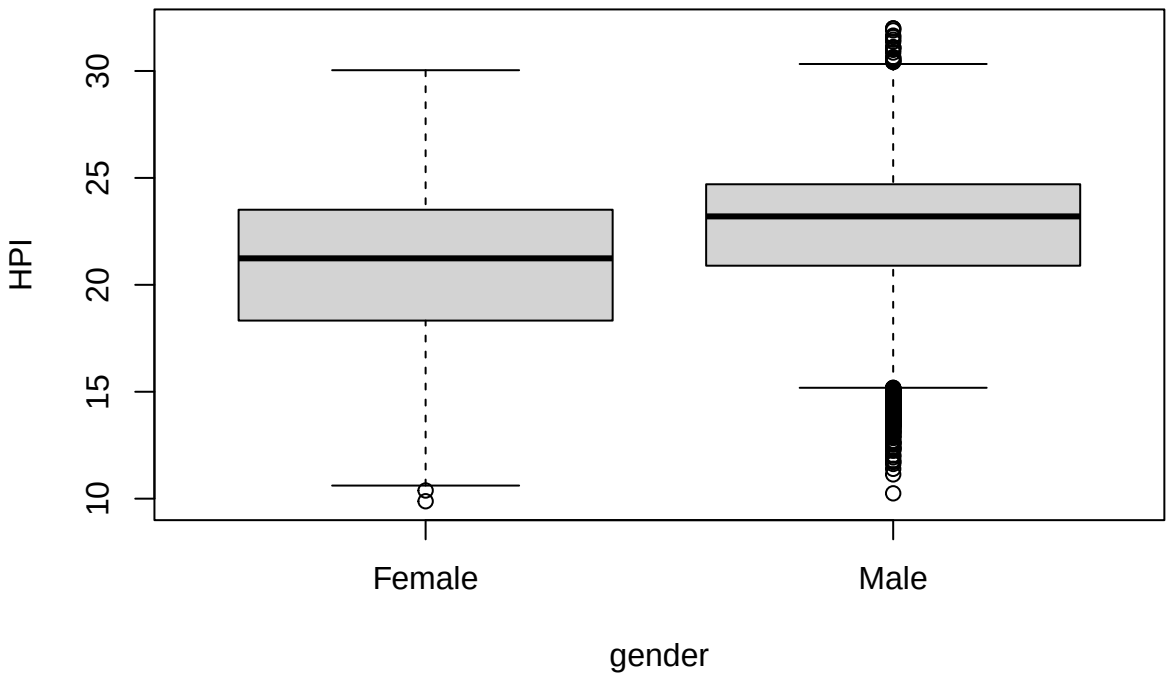
wykres pudełkowy

```
boxplot(panteon$HPI ~ panteon$gender)
```



wykres pudełkowy według kategorii

```
boxplot(HPI~gender, data = panteon)
```



Zadanie: boxplot HPI według kontynentów

Wykres słupkowy.

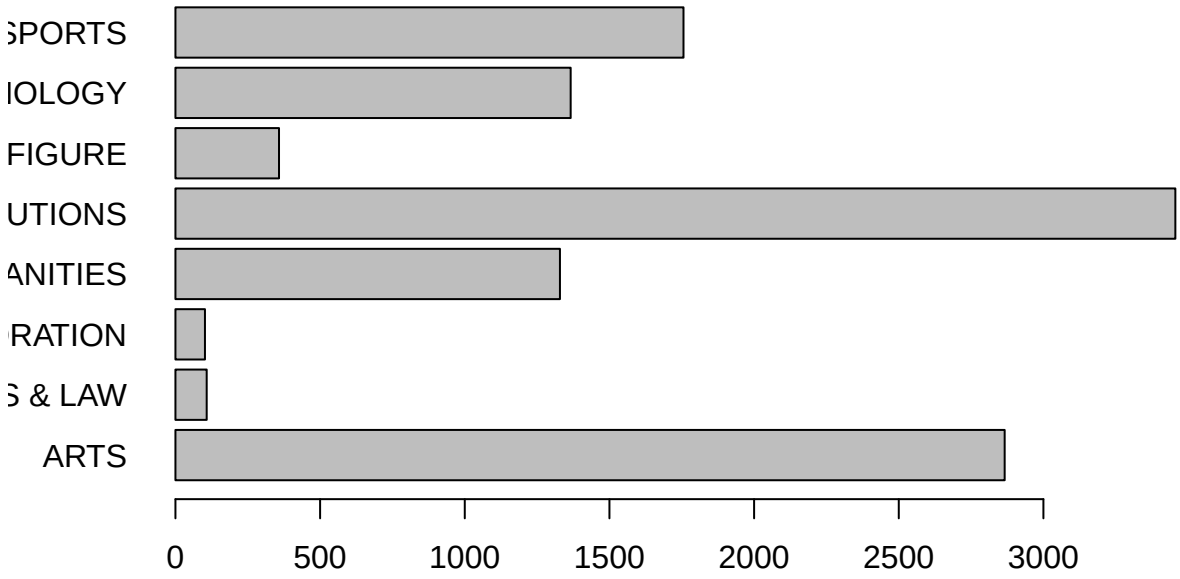
1. zliczenie funkcją table

```
(domeny <- table(panteon$domain))
```

```
##
##                 ARTS       BUSINESS & LAW          EXPLORATION
##                 2866                  108                  102
##           HUMANITIES         INSTITUTIONS       PUBLIC FIGURE
##                 1329                 3456                  358
## SCIENCE & TECHNOLOGY               SPORTS
##                 1366                 1756
```

2. wykres słupkowy

```
barplot(domeny,
        horiz = TRUE,
        las =1)
```



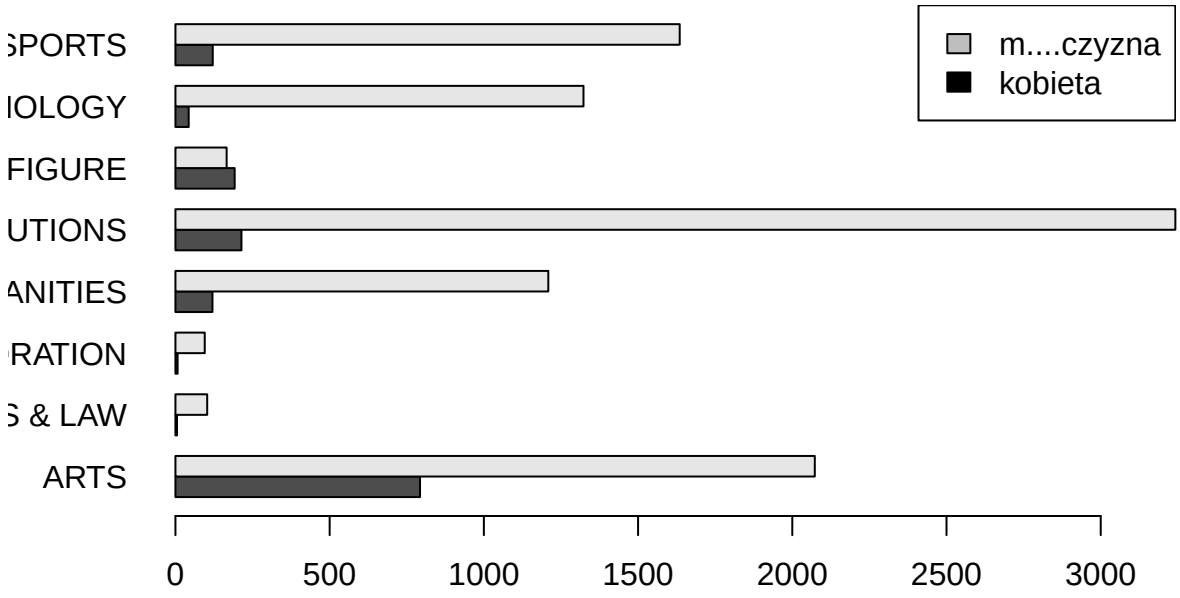

```
?barplot
```

elemnty dodatkowe

```
domeny_gender <- table(panteon$gender,
                       panteon$domain)

barplot(domeny_gender,
        las = 1, # orientacja etykiet osi
        beside = TRUE, #słupki obok siebie
        horiz = TRUE)

legend("topright",
       c("mężczyzna", "kobieta"),
       fill = c("grey", "black"))
```

```
## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <c4>

## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <99>

## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
```

```
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <c5>

## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <bc>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <c4>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <c5>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <bc>
```

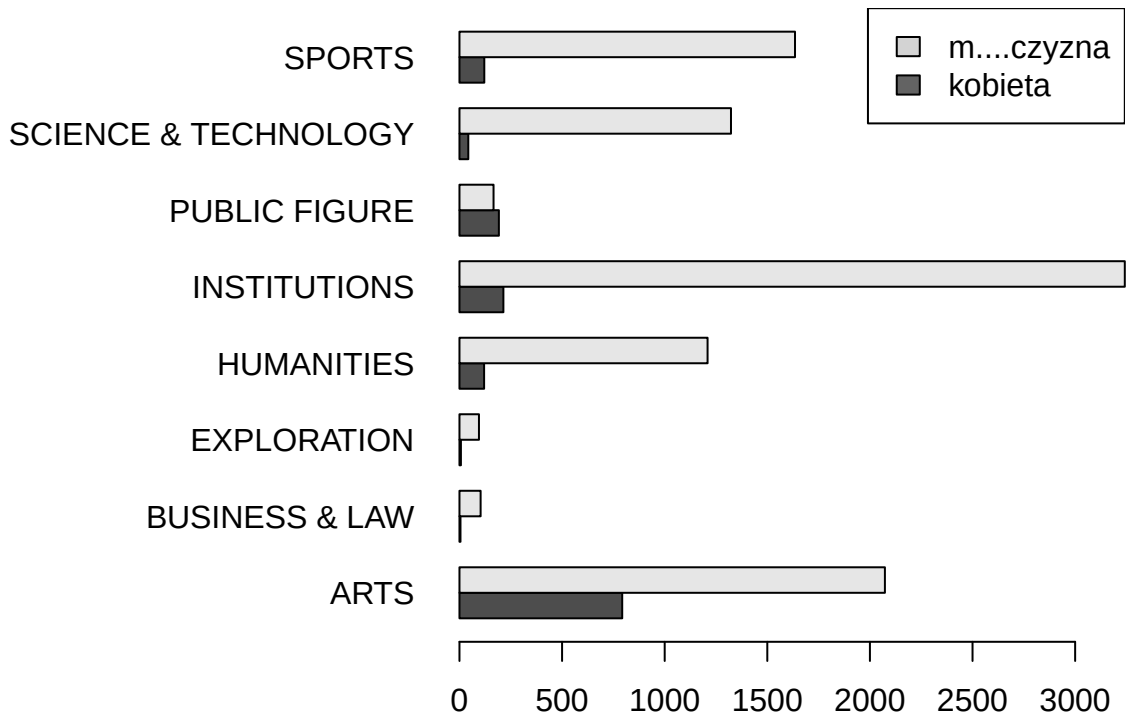


marginesy

```
par(mar=c(3,12,3,3)) # marginesy w kolejności par(mar = c(bottom, left, top, right))
#czyli dół, lewy, góra, prawy
```

```r
barplot(domeny_gender,
        las = 1,
        beside = TRUE,
        horiz = TRUE)

legend("topright",
       c("mężczyzna", "kobieta"),
       fill = c("lightgrey", "gray39"))
```

```
## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <c4>

## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <99>

## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <c5>

## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <bc>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <c4>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <c5>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <bc>
```



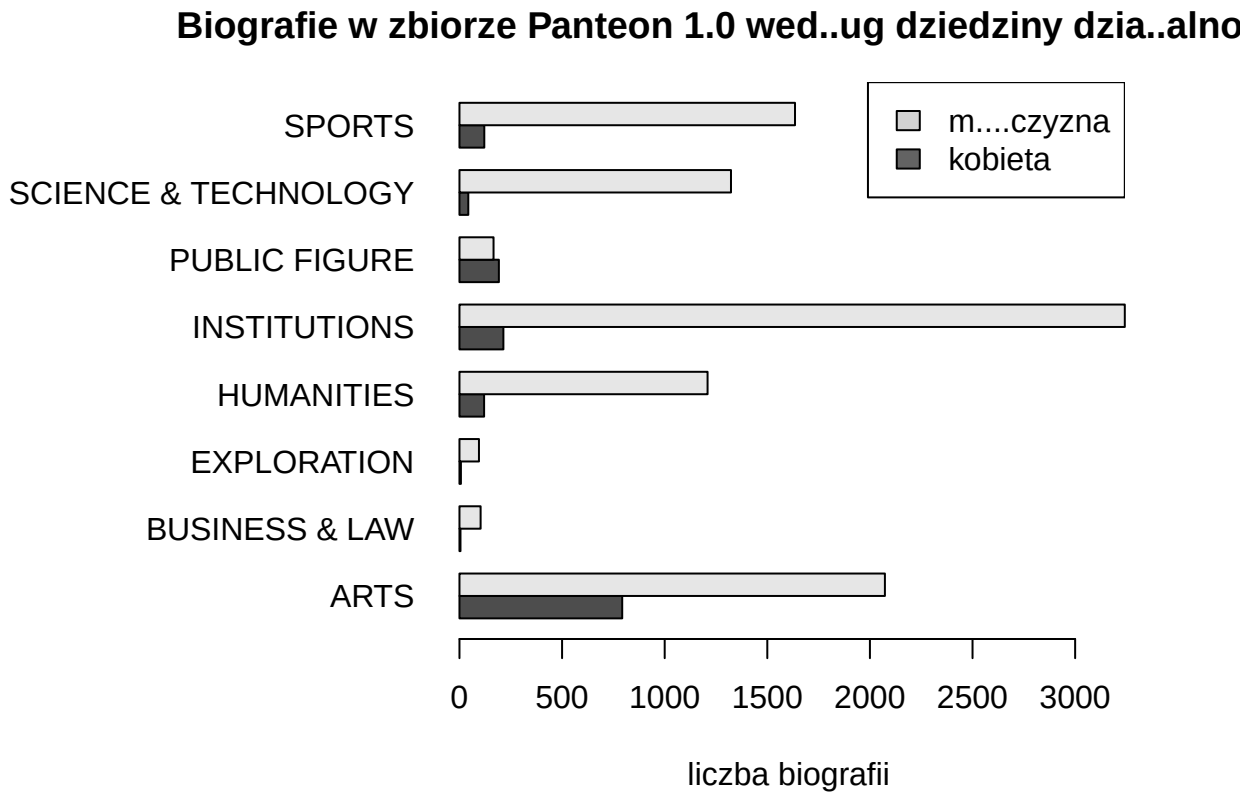

```
par(mar=c(5,12,3,3)) # marginesy w kolejności par(mar = c(bottom, left, top, right))
#czyli dół, lewy, góra, prawy

barplot(domeny_gender,
        las = 1,
        beside = TRUE,
        horiz = TRUE,
        xlab = "liczba biografii",
        main = "Biografie w zbiorze Panteon 1.0 według dziedziny działalności i płci")
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'Biografie w zbiorze Panteon 1.0 według dziedziny
## działalności i płci' in 'mbcsToSbcs': dot substituted for <c5>

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'Biografie w zbiorze Panteon 1.0 według dziedziny
## działalności i płci' in 'mbcsToSbcs': dot substituted for <82>

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'Biografie w zbiorze Panteon 1.0 według dziedziny
## działalności i płci' in 'mbcsToSbcs': dot substituted for <c5>

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'Biografie w zbiorze Panteon 1.0 według dziedziny
## działalności i płci' in 'mbcsToSbcs': dot substituted for <82>
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'Biografie w zbiorze Panteon 1.0 według dziedziny
## działalności i płci' in 'mbcsToSbcs': dot substituted for <c5>

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'Biografie w zbiorze Panteon 1.0 według dziedziny
## działalności i płci' in 'mbcsToSbcs': dot substituted for <9b>

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'Biografie w zbiorze Panteon 1.0 według dziedziny
## działalności i płci' in 'mbcsToSbcs': dot substituted for <c5>

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'Biografie w zbiorze Panteon 1.0 według dziedziny
## działalności i płci' in 'mbcsToSbcs': dot substituted for <82>
```

```
legend("topright",
       c("mężczyzna", "kobieta"),
       fill = c("lightgrey", "gray39"))
```

```
## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <c4>

## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <99>

## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <c5>

## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <bc>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <c4>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <c5>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <bc>
```



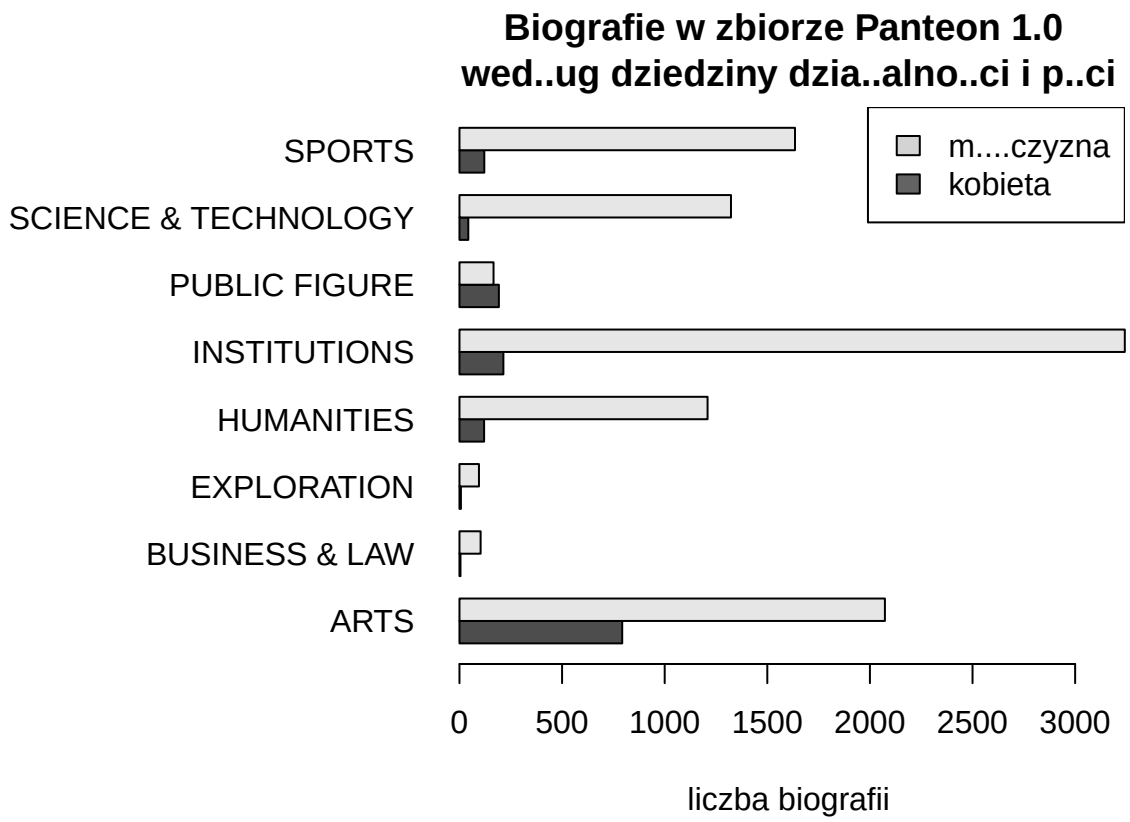

```
par(mar=c(5,12,3,3)) # marginesy w kolejności par(mar = c(bottom, left, top, right))
#czyli dół, lewy, góra, prawy

barplot(domeny_gender,
        las = 1,
        beside = TRUE,
        horiz = TRUE,
        xlab = "liczba biografii",
        main = "Biografie w zbiorze Panteon 1.0 \nwedług dziedziny działalności i płci")
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'według dziedziny działalności i płci' in 'mbcsToSbcs':
## dot substituted for <c5>

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'według dziedziny działalności i płci' in 'mbcsToSbcs':
## dot substituted for <82>

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'według dziedziny działalności i płci' in 'mbcsToSbcs':
## dot substituted for <c5>

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'według dziedziny działalności i płci' in 'mbcsToSbcs':
## dot substituted for <82>
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'według dziedziny działalności i płci' in 'mbcsToSbcs':
## dot substituted for <c5>

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'według dziedziny działalności i płci' in 'mbcsToSbcs':
## dot substituted for <9b>

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'według dziedziny działalności i płci' in 'mbcsToSbcs':
## dot substituted for <c5>

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'według dziedziny działalności i płci' in 'mbcsToSbcs':
## dot substituted for <82>
```

```
legend("topright",
       c("mężczyzna", "kobieta"),
       fill = c("lightgrey", "gray39"))
```

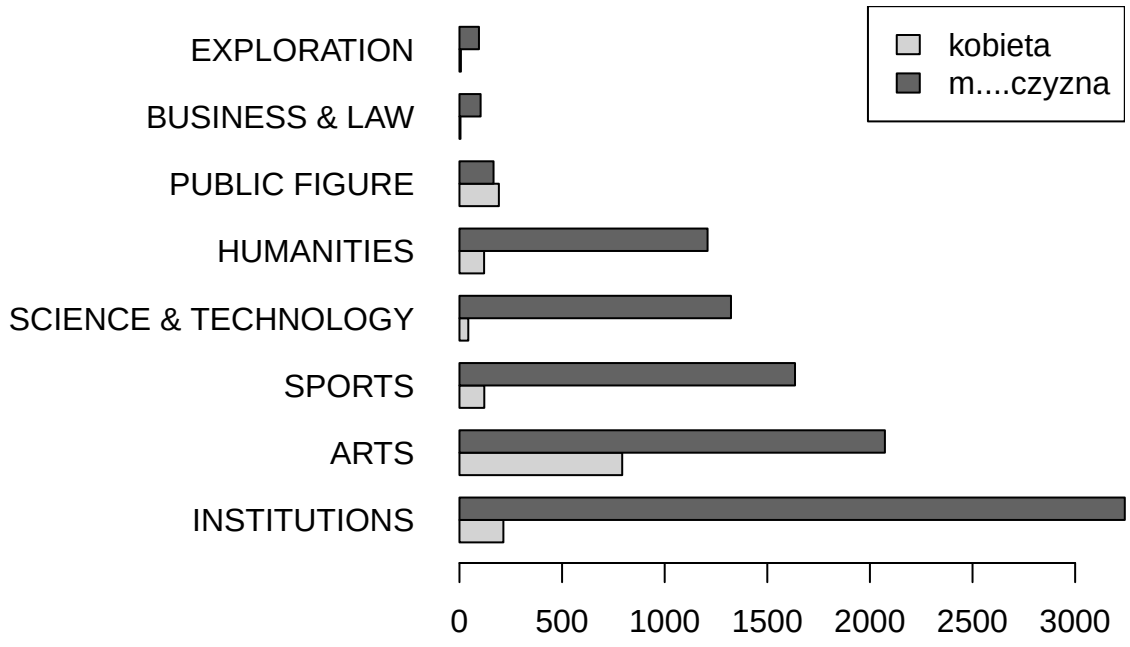
```
## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <c4>

## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <99>

## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <c5>

## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <bc>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <c4>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <c5>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <bc>
```




```
par(mar=c(5,12,3,3))

#  Sortowanie według sum w kolumnach
sorted_indices <- order(colSums(domeny_gender), decreasing = TRUE)

# Zastosowanie sortowania do kolumn
domeny_gender_sorted <- domeny_gender[, sorted_indices]

# Tworzenie wykresu z posortowanymi danymi
barplot(
  domeny_gender_sorted,
  las = 1,
  beside = TRUE,
  horiz = TRUE,
  col = c("lightgrey", "gray39")
)

# Dodanie legendy
legend(
  "topright",
  c("kobieta", "mężczyzna"),
  fill = c("lightgrey", "gray39")
)
```

```
## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
```

```
## <c4>

## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <99>

## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <c5>

## Warning in (function (s, units = "user", cex = NULL, font = NULL, vfont =
## NULL, : conversion failure on 'mężczyzna' in 'mbcsToSbcs': dot substituted for
## <bc>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <c4>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <c5>

## Warning in text.default(x, y, ...): conversion failure on 'mężczyzna' in
## 'mbcsToSbcs': dot substituted for <bc>
```



Zadanie barplot według płci i kontynentów

```
dev.off()
```

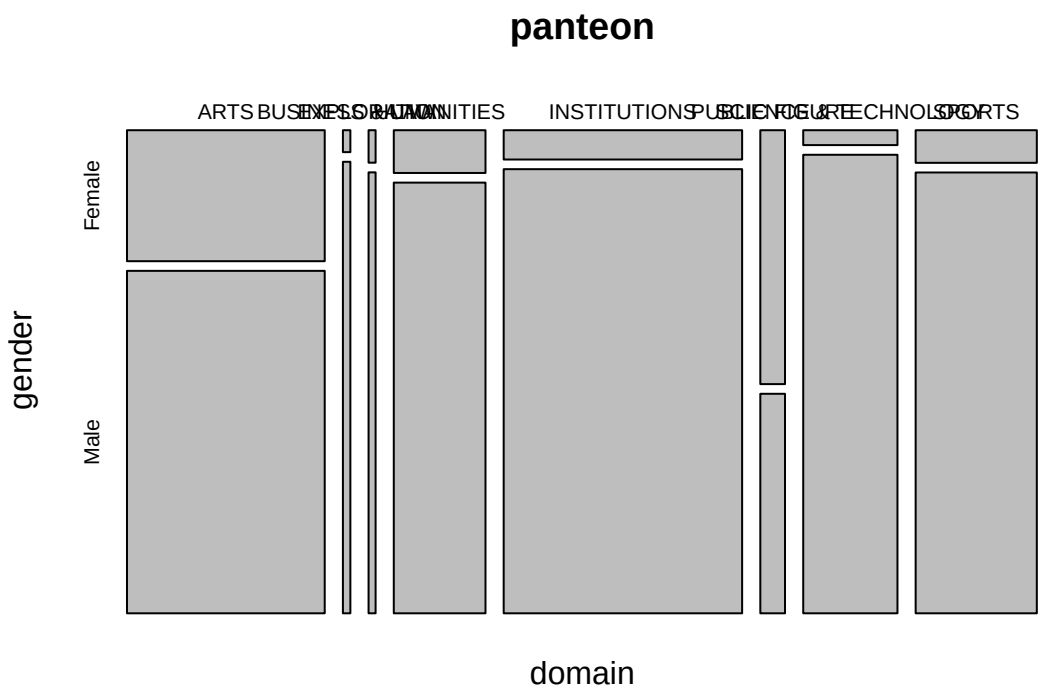
```
## null device
##           1
```

```
?dev.off()
```

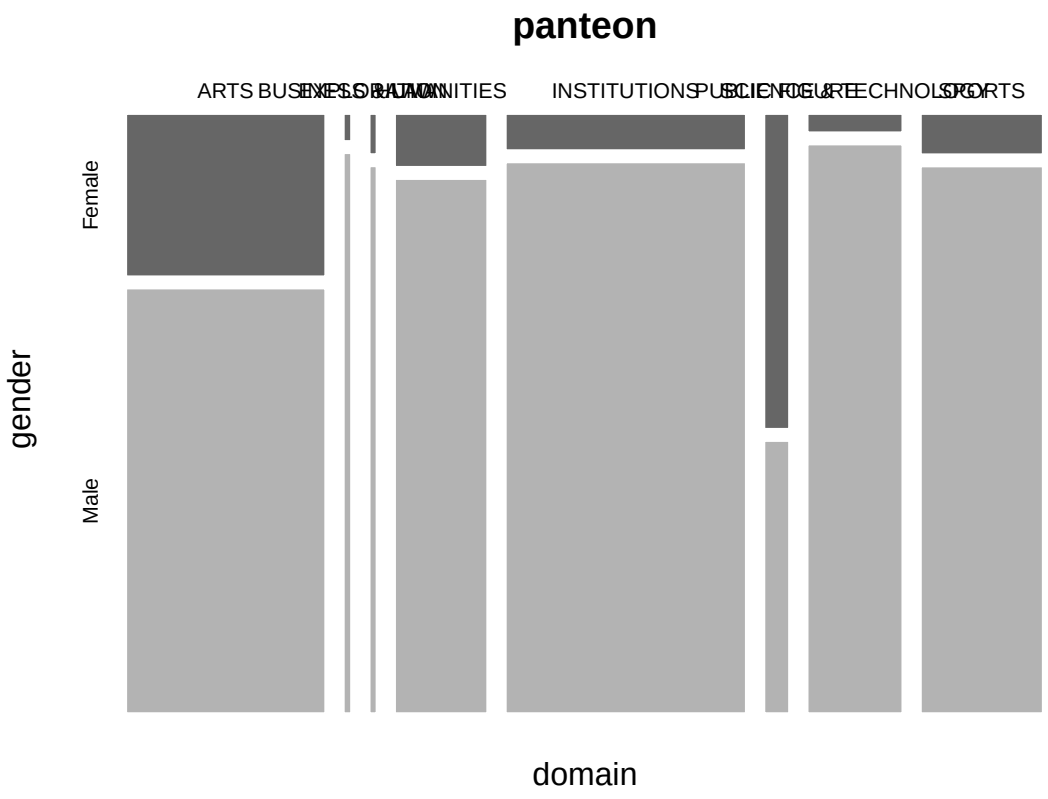
Wykres mozaikowy

```
?mosaicplot
```

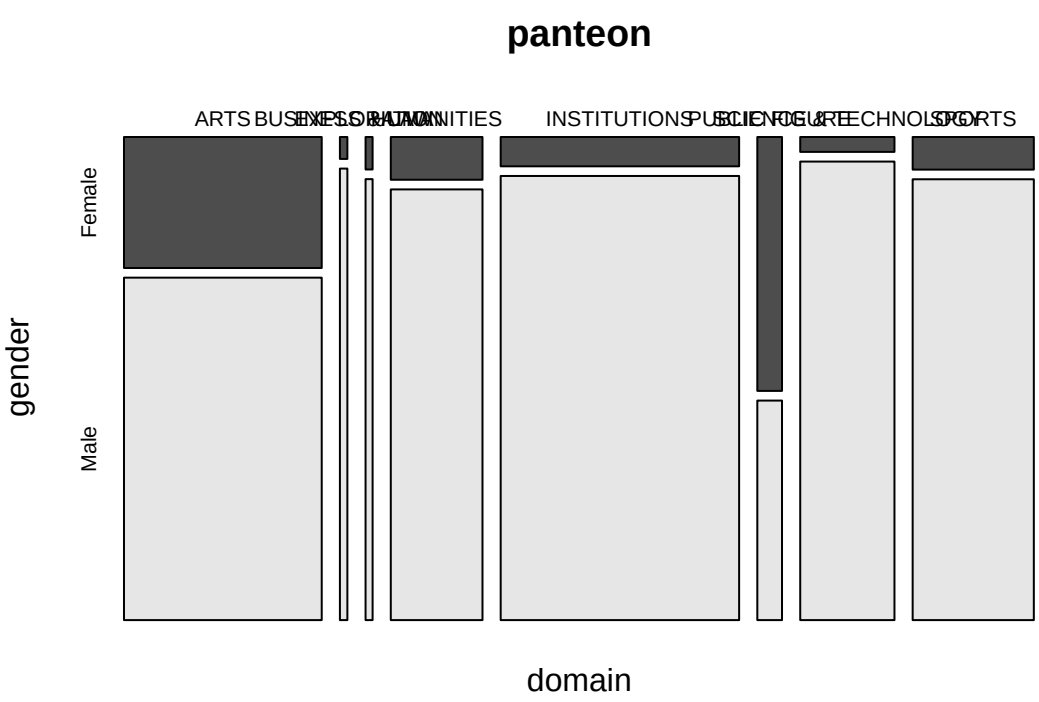
```
mosaicplot(~domain+gender, data = panteon)
```




```
par(mar=c(3,3,3,3))

mosaicplot(~domain+gender,
           data = panteon,
           border = "white",
           col = c("grey40", "grey70"))
```




```
mosaicplot(~domain + gender,
           data = panteon,
           color = TRUE)
```



kolejność

```
(domeny_ko <- as.data.frame(table(panteon$domain)))
```

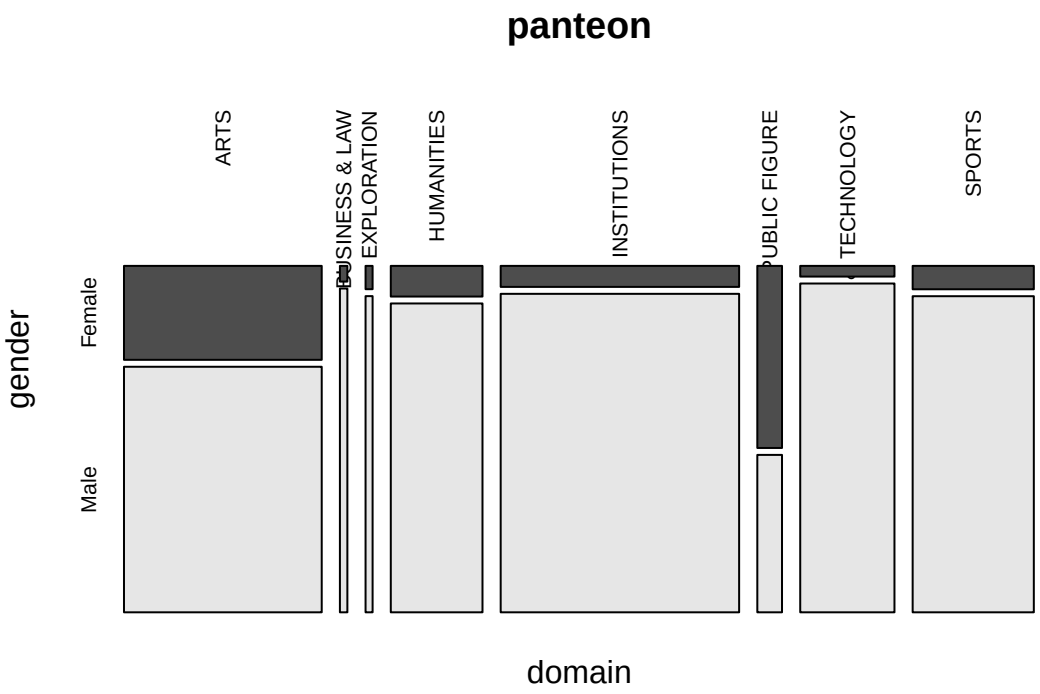
```
##                   Var1 Freq
## 1                 ARTS 2866
## 2       BUSINESS & LAW  108
## 3          EXPLORATION  102
## 4           HUMANITIES 1329
## 5         INSTITUTIONS 3456
## 6        PUBLIC FIGURE  358
## 7 SCIENCE & TECHNOLOGY 1366
## 8               SPORTS 1756
```

Zmienne typu factor i ustalanie ich porządku funkcją faktor

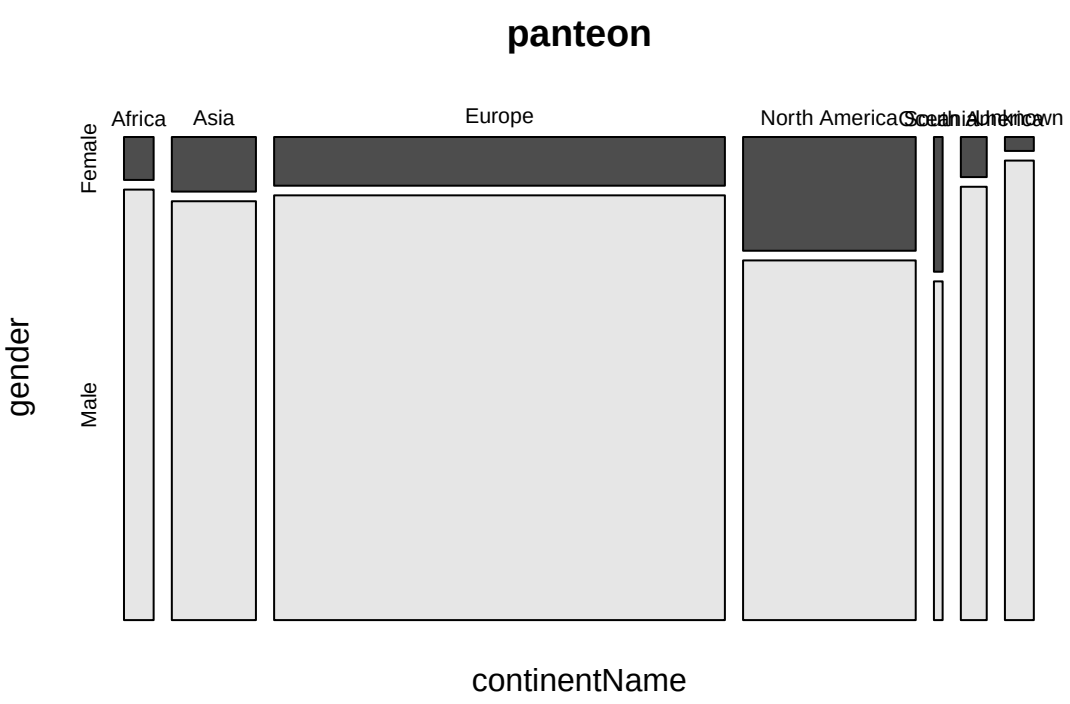
```
panteon$domain <- factor(panteon$domain, levels = c("INSTITUTIONS",
                                                    "ARTS",
                                                    "SPORTS",
                                                    "SCIENCE & TECHNOLOGY",
                                                    "HUMANITIES",
                                                    "PUBLIC FIGURE",
                                                    "BUSINESS & LAW",
                                                    "EXPLORATION"))
```

```
panteon$domain <- factor(panteon$domain, levels = domeny_ko$Var1)
```

```
mosaicplot(~domain + gender,
           data = panteon,
           color = TRUE,
           las = 3)
```

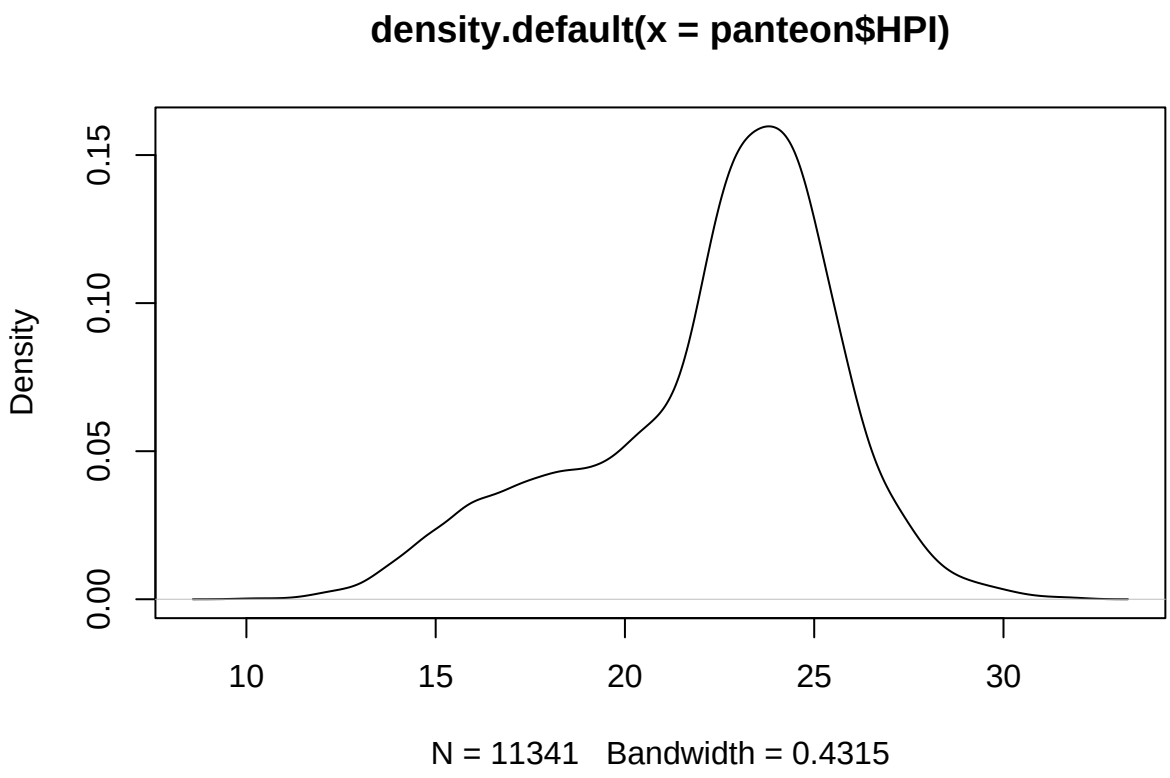



```
mosaicplot(~continentName + gender,
           data = panteon,
           color = TRUE)
```



wykres gęstości

```
plot(density(panteon$HPI))
```

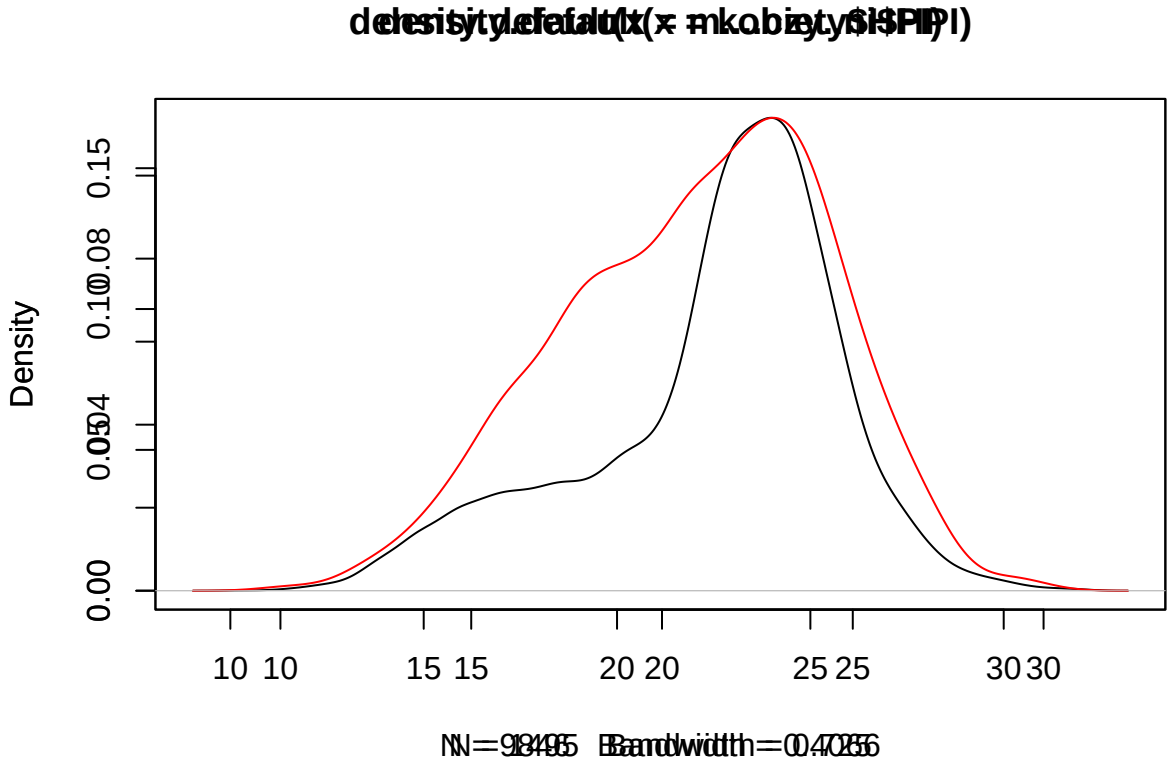



```
mężczyźni <- filter(panteon, gender == "Male")
kobiety <- filter(panteon, gender == "Female")
```

```
plot(density(mężczyźni$HPI))
```

```
## Warning in title(...): conversion failure on 'density.default(x =
## mężczyźni$HPI)' in 'mbcsToSbcs': dot substituted for <c4>

## Warning in title(...): conversion failure on 'density.default(x =
## mężczyźni$HPI)' in 'mbcsToSbcs': dot substituted for <99>

## Warning in title(...): conversion failure on 'density.default(x =
## mężczyźni$HPI)' in 'mbcsToSbcs': dot substituted for <c5>

## Warning in title(...): conversion failure on 'density.default(x =
## mężczyźni$HPI)' in 'mbcsToSbcs': dot substituted for <bc>

## Warning in title(...): conversion failure on 'density.default(x =
## mężczyźni$HPI)' in 'mbcsToSbcs': dot substituted for <c5>

## Warning in title(...): conversion failure on 'density.default(x =
## mężczyźni$HPI)' in 'mbcsToSbcs': dot substituted for <ba>
```

```
par(new = TRUE)
plot(density(kobiety$HPI),
     col = "red")
```



Usuniemy nakładające się elementy

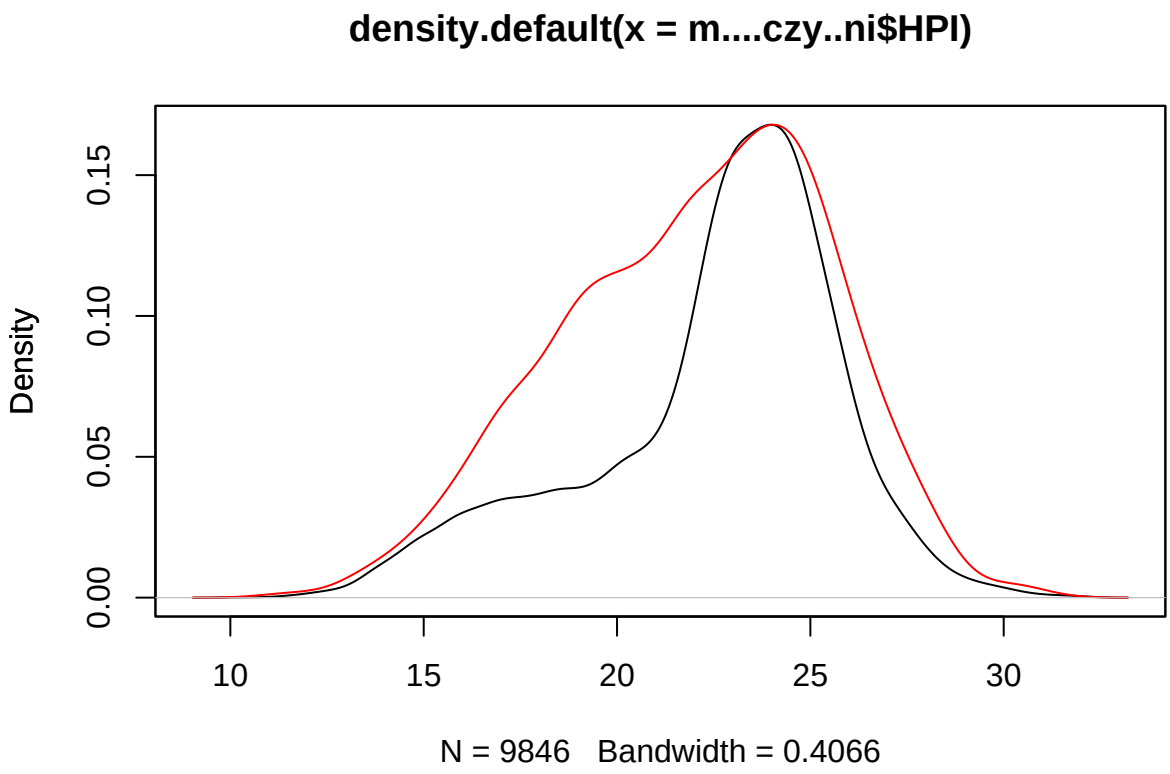
```
?par()
```

```
plot(density(mężczyźni$HPI))
```

```
## Warning in title(...): conversion failure on 'density.default(x =
## mężczyźni$HPI)' in 'mbcsToSbcs': dot substituted for <c4>

## Warning in title(...): conversion failure on 'density.default(x =
## mężczyźni$HPI)' in 'mbcsToSbcs': dot substituted for <99>

## Warning in title(...): conversion failure on 'density.default(x =
## mężczyźni$HPI)' in 'mbcsToSbcs': dot substituted for <c5>

## Warning in title(...): conversion failure on 'density.default(x =
## mężczyźni$HPI)' in 'mbcsToSbcs': dot substituted for <bc>

## Warning in title(...): conversion failure on 'density.default(x =
## mężczyźni$HPI)' in 'mbcsToSbcs': dot substituted for <c5>
```

```
## Warning in title(...): conversion failure on 'density.default(x =
## mężczyźni$HPI)' in 'mbcsToSbcs': dot substituted for <ba>
```

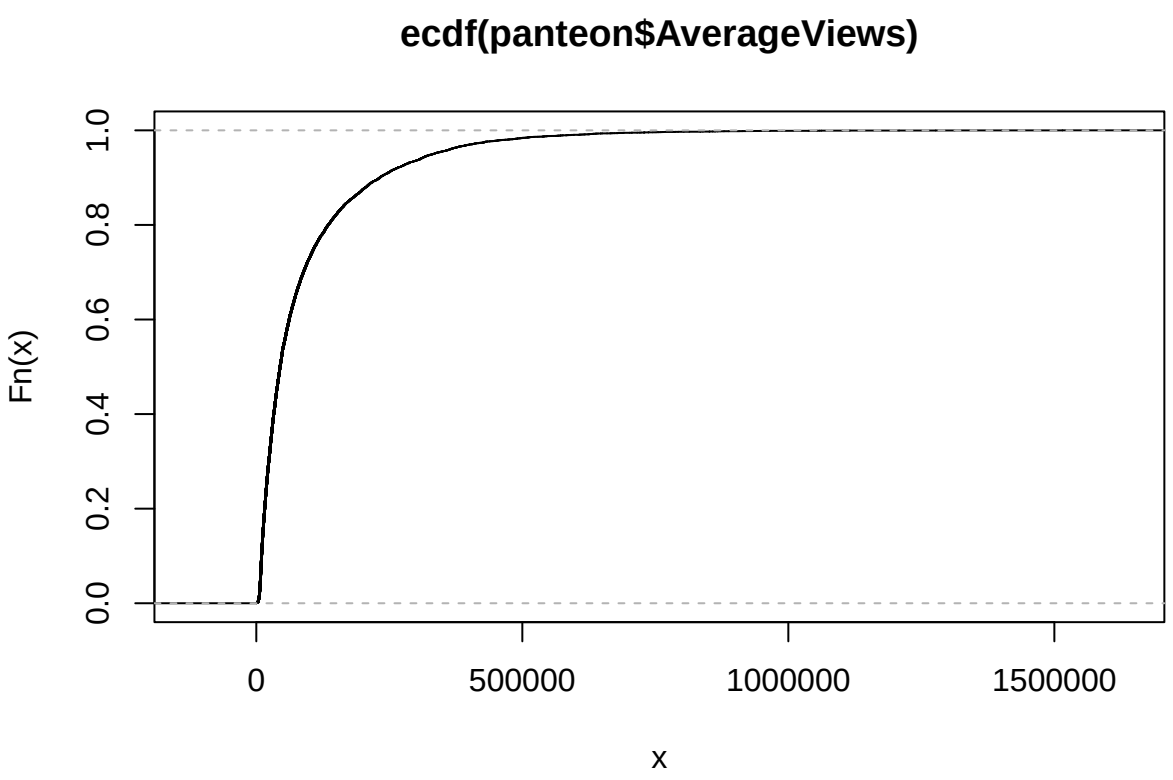
```
par(new = TRUE)
plot(density(kobiety$HPI),
     col = "red",
     xlab = "",
     main = "",
     xaxt = "n",
     yaxt = "n")
```



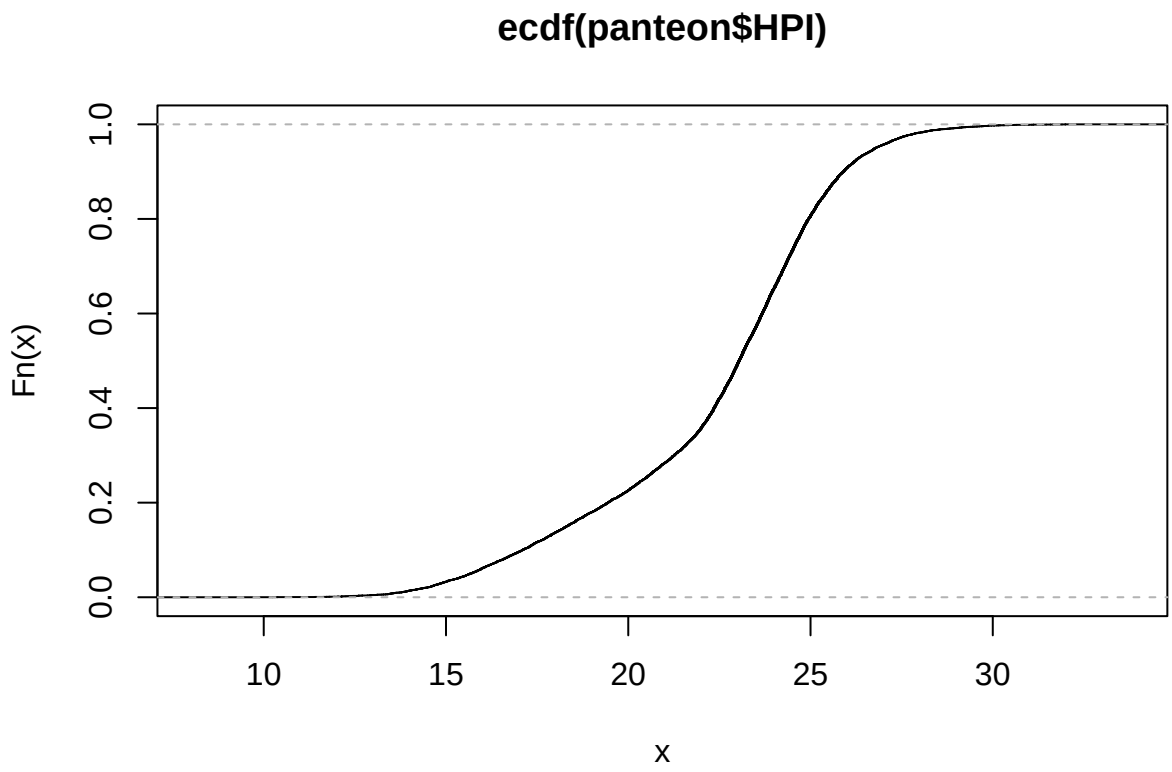
ecdf - dystrybuanta empiryczna

Jaki procent obserwacji przyjmuje wartość mniejszą niż x

```
plot(ecdf(panteon$AverageViews))
```




```
plot(ecdf(panteon$HPI))
```



Pomoc

```
?help()
```

```
help(ggplot)
```

Katalog roboczy

Zasadniczo lepiej operować używając rproj w RStudio niż katalogu roboczego.

Informacja o katalogu roboczym:

```
getwd()
```

```
## [1] "/Users/golemxiv/Documents/dydaktyka/wizualizacjaR/podstawy"
```

Ustawianie katalogu roboczego

```
setwd() # argumentem jest ścieżka dostępu do katalogu roboczego w cudzysłowie
```

```
## Error in setwd(): argument "dir" is missing, with no default
```

Skrypty

Skróty przydatne w skryptach:

ctrl enter - uruchamia zaznaczony fragment kodu ctrl shift enter - uruchamia cały skrypt

Praca domowa:

- tworzymy nowy projekt
- umieszczamy w nim plik z danymi
- tworzymy w nim plik r markdown który wczytuje potrzebne biblioteki, wczytuje dane i rysuje na ich podstawie wykres
- kompresujemy folder projektu
- wysyłamy na t.olczyk@uw.edu.pl
- notujemy wszelkie problemy, pytania i trudności jakie pojawią się w tym procesie

Co po zajęciach

Plan minimum:

Ściągawka Rstudio

Plan dla ambitnych

Rozdział 1-3 Long, J. D. (2020). Język R: Receptury: analiza danych, statystyka i przetwarzanie grafiki, (K. Sawka, Tłum.). Helion SA.

online po angielsku

Co przed następnymi zajęciami?:

Plan minimum:

Ściągawka ggplot

Plan dla ambitnych:

rozdział 1 z Wickham, H., & Grolemund, G. (2020). Język R: Kompletny zestaw narzędzi dla analityków danych (J. Zatorska, Tłum.). Wydawnictwo Helion.

online po angielsku