Metody probabilistyczne i statystyka (informatyka)

Laboratorium: R i RStudio. Podstawowe funkcje w R.

Jarosław Kotowicz

rok akademicki 2024/2025

Contents

Wprowadzenie

Instalacja oprogramowania

1. R strona domowa www.r-project.org

- wybieramy download
- wybieramy mirror np. serwer niemiecki
- ściągamy plik instalacyjny

2. RStudio (IDE)[1] dla R RStudio (https://posit.co/download/rstudio-desktop)

- należy pobrać system opensource

3. (NIEKONIECZNE) Przeglądarka plików **pdf** natywna dla RStudio (pod Windows) **sumatra** (open-source)

- pobieramy ze strony Sumatra

4. Instalujemy w kolejności R, Sumatra, RStudio

Przygotowanie do pracy

1. W katalogu dokumenty tworzymy folder (nazwa zgodnie z zasadą **Inf_xxxxx**, gdzie *xxxxx* to numer albumu).
2. Uruchamiamy *RStudio*
3. Zachowujemy porządek. Tworzymy nowy projekt

- *Files -> New Project... -> Existing Directory -> Browse...* i wybieramy folder, który utworzyliśmy oraz zaznaczamy *Open in new session*. Uwaga: Zawsze na koniec pracy/zajęć należy zamknąć projekt. *Files -> Close Project* Uwaga: Zawsze na początku pracy/zajęć należy otworzyć projekt. *Files -> Recent Projects* i wybrać swój (*będzie miał on taką nazwę, jak nasz katalog*).

4. Dostosowujemy *RStudio* do następujących wymagań:

- *Tools -> Project Options ->*
  Sweave -> PDF Generaion wybór **knitr** Uwaga: Jeżeli chcemy korzystać z TeX'a (LaTeX'a), to musimy mieć zainstalowany system/program np. MikTeX.
- *Tools -> Global Options ->*
  Appearance wybór wyglądu *Editor theme* np. **ambiance** (*ciemny motyw*).

---

4. W domu instalujemy konieczne biblioteki **R** nazywane **package**

- dolne prawe okienko - okno *Packages -> Install* wpisujemy **tidyverse**

5. Otwieramy pliki

- **R Script**: *Files -> New File -> R Script*
- **R Notebook**: *Files -> New File -> R Notebook*
- **R Markdown**: *Files -> New File -> R Markdown* i wybieramy rodzaj dokumentu np. *Document* lub *Presentation*. Uwaga: Wszystkie pliki podpisujemy zgodnie ze schematem *Inf_xxxxx*.

---

6. Będziemy zawsze pracować albo w pliku skryptowym *R* albo w *R Notebook*.
7. Informacje dla pliku skryptowego

- wywołanie aktualnej linii w pliku skryptowym *CTRL + ENTER* lub zakładka *Run* w górnym prawym rogu okienka pliku skryptowego,
- podczytanie biblioteki *library(nazwa biblioteki)* lub *require(nazwa biblkioteki)*,
- odwołanie się do funkcji z danej biblioteki/pakietu *function*, gdy bibliotekę została podczytana lub *package::function* np. *stats::filter*, gdy nie została,
- znak komentarza hashtag tzn. #,
- komentowanie zaznaczonego fragmentu kodu *CTRL + SHIFT + c*.

---

8. Import danych (INFORMCJE DO PÓŹNIEJSZEGO WYKORZYSTANIA)

- korzystamy z danych dostępnych na stronie Jareda P. Landera (https://jaredlander.com/data)

[1] IDE - integrated development environment.

- plik *csv* (tekstowe) import: *File -> Import Dataset -> From Text (readr)* (w polu *File/URL* kopiujemy adres *https://jaredlander.com/data/* i dopisujemy nazwę pliku np. *acs_ny.csv*) -> *Update*
- wykorzystujemy opcje *Import Options* aby wybrać np. znak rozdzielający dane średnik to *semicolon* itd.
- możemy zmienić też typy kolumn wybierając dla kolumny z menu rozwijalnego wartość Uwaga: W przypadku typu *factor* (zmienna czynnikowa) należy wskazać wszystkie poziomy (wartości) (**BARDZO NIEWYGODNY SPOSÓB**).
- kopiujemy z okienka *Code Preview* do pliku skryptowego polecenie poczytując wcześniej bibliotekę *readr*, ja tutaj podczytuję *tidyverse*.
- braki danych **NA** (skrót od *not available*), nieliczby **NaN** (*not a number*)
- typ *numeric* w R identyczny jak *double* w C++ i co więcej używane też jest równoważnie *numeric* i *double*

Podstawowe obiekty w R.

Pracujemy z wersją 4.3.2 R [1].

biblioteka: **tidyverse** [2].

Czyszczenie środowiska (na każdym laboratorium).

```
rm(list = ls())
```

Przypisania

Przypisanie wartości liczbowej do nazwy (zmiennej)

```
a1 <- 2
a1
## [1] 2

assign('a2', 2)
a2
## [1] 2

(a3 <- 2)
## [1] 2
```

PONIŻSZE PRZYPISANIE JEST NIEPREFEROWANE

```
(a4 = 2)
## [1] 2
```

Wszystko w R jest obiektem. Każdy skalar jest wektorem

```
a4[1]
## [1] 2
```

Przypisanie wartości wektora do nazwy (zmiennej)

```
v1 <- c(2, 3, 4)
v1
## [1] 2 3 4
v1[1]
## [1] 2
```

Typ znakowy *character*

```
napis <- "napis"
napis
## [1] "napis"
```

Wektor, którego elementy są typu znakowego

```
z <- c("napis", "to", "ja")
z
## [1] "napis" "to"    "ja"
```

Łączenie elementów wektora

```
paste(z, collapse = " ")
## [1] "napis to ja"
```

Typ liczbowy *double* (liczby zmiennoprzecinkowe)

```
x <- 1
x
## [1] 1
```

Wektor, którego elementy są typu zmiennoprzecinkowego

```
v2 <- c(1, 25, 7)
v2
## [1]  1 25  7
```

Ostrzeżenie o działaniach na liczbach zmiennoprzecinkowych

```
0.1 * 0.1 == 0.01
## [1] FALSE
```

Typ *integer* (liczby całkowite)

```
z <- 1L
```

```
is.integer(a1)
## [1] FALSE
a4 <- 2L
is.integer(a4)
## [1] TRUE

a5 <- as.integer(a1)
as.integer(2.5)
## [1] 2
```

Typ logiczny *logical*

```
v3 <- c(TRUE, FALSE)
v3
## [1]  TRUE FALSE
```

NIEZALECANE używanie skrótów, o ile nie piszę się programów

```r
v4 <- c(T, F)
v4
## [1]  TRUE FALSE
```

Zmienne czynnikowe (kategorialne)

```r
f1 <- factor(c("zły", "dobry"))
f1
## [1] zły   dobry
## Levels: dobry zły
```

Zmienne czynnikowe (uporządkowane)

```r
f2 <- factor(c("zły", "dobry"), ordered = TRUE)
f2
## [1] zły   dobry
## Levels: dobry < zły
```

Kolejność uporządkowania

```r
f3 <- factor(c("zły", "dobry"), levels = c("zły", "dobry"), ordered = TRUE)
f3
## [1] zły   dobry
## Levels: zły < dobry
```

Listy

```r
l1 <- list(litery = letters, cyfry = 0:9, logiczne = c(TRUE, FALSE))
l1
## $litery
##  [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"
##
## $cyfry
##  [1] 0 1 2 3 4 5 6 7 8 9
##
## $logiczne
## [1]  TRUE FALSE
```

```r
l2 <- list(l1, imiona = c("Adam", "Ewa"), nazwiska = c("Kowalski", "Nowak"))
l2
## [[1]]
## [[1]]$litery
##  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
## [20] "t" "u" "v" "w" "x" "y" "z"
##
## [[1]]$cyfry
##  [1] 0 1 2 3 4 5 6 7 8 9
##
## [[1]]$logiczne
```

```
## [1]  TRUE FALSE
##
##
## $imiona
## [1] "Adam" "Ewa"
##
## $nazwiska
## [1] "Kowalski" "Nowak"
```

Tablice wielowymiarowe

```
ar1 <- array(rnorm(100), c(10, 5, 2))
ar1
## , , 1
##
##              [,1]        [,2]       [,3]        [,4]       [,5]
##  [1,]  0.13509677 -0.90696393  0.4656015 -0.72988707  0.4333856
##  [2,] -0.05119224 -1.53303009 -1.9960615 -1.02565458  1.0025242
##  [3,] -0.79074109 -0.12312042  0.6221302  0.56120167  1.9562995
##  [4,]  0.61250918 -0.72384254  0.5191677  0.55632667  0.4110655
##  [5,] -0.68573059  0.66397713  0.6431412 -1.09388544 -0.4037064
##  [6,] -0.62977496 -0.84765534  2.0286211 -1.50552575 -0.9317649
##  [7,] -0.60404796 -0.02239486 -0.2617581 -0.37432258 -0.5533862
##  [8,]  1.12943018  0.89952698 -0.6197445 -1.45553550  1.3206827
##  [9,] -1.30437341 -1.44553626 -0.6561104  0.04093708  0.7505326
## [10,] -0.87353560  1.17542769 -0.5230713  0.72547217 -0.3001208
##
## , , 2
##
##              [,1]       [,2]       [,3]       [,4]       [,5]
##  [1,] -0.86480595 -0.2508134  0.7343213 -0.9204918  1.1444800
##  [2,]  0.10934632 -0.9690001  0.1249508 -1.6734932  0.6014900
##  [3,]  0.05488327  1.0784779 -0.8858990 -0.3851834 -0.5403894
##  [4,] -0.22415691 -0.6226460 -0.1513462 -0.8043804 -2.1009250
##  [5,] -0.78183270 -0.6464786  0.4236044  0.1365915 -2.3092643
##  [6,]  0.45358280 -0.1126876 -1.2262668  0.1445106 -0.4993910
##  [7,] -1.76398070 -0.7884912 -0.8413300 -3.6857529 -0.1774876
##  [8,]  0.51724328  0.4673696  1.3892612  0.9650768  0.8990942
##  [9,]  0.05910796  0.6012379 -0.2974048 -0.4210341  1.7885544
## [10,] -0.42737373 -0.8140928  1.9258532  0.9431368  0.4885011
```

Macierze i wybrane operacje na nich

```
m1 <- matrix(1:4, ncol = 2, nrow = 2)
m1
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
m2 <- matrix(1:4, ncol = 2)
```

```
m2
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
m3 <- matrix(1:4, ncol = 2, byrow = TRUE)
m3
##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
```

```
m4 <- matrix(rnorm(100), ncol = 10, nrow = 10)
m4
##              [,1]       [,2]       [,3]        [,4]       [,5]       [,6]
##  [1,]  1.02567581  1.2610999 -0.9645456 -0.14697234  0.1488480 -0.1742230
##  [2,] -1.39794740 -0.4888950 -0.2319919  2.28536288 -1.9900652 -1.2988819
##  [3,] -0.27722498  2.0774252 -0.4439360  0.68557787  0.3846282 -0.8475726
##  [4,]  1.10662077 -0.6325807  0.9611398 -0.97144367  0.6215753  0.7172027
##  [5,] -0.02869781 -0.6153722 -0.2717784  0.37090831 -0.8448948 -2.4294826
##  [6,]  1.17652080 -0.2178309  0.5567940 -0.78042505  0.3388594 -0.7946059
##  [7,]  0.28024038 -1.0977501  0.9544588 -0.78076765  0.4085061  0.2456427
##  [8,]  0.26914415 -1.6869653 -1.2734856 -0.02001303 -0.2182018  1.1573435
##  [9,]  0.31834798 -0.4591093 -0.1621171  0.50673982  0.1139678  0.7072789
## [10,] -0.95025797 -1.0074917 -0.5087157 -2.07862947 -0.9088694  1.2824386
##              [,7]       [,8]       [,9]      [,10]
##  [1,] -0.62346827  1.1959404 -1.0773430 -1.3034006
##  [2,] -0.64386063 -0.2307029 -0.3743702 -0.6246145
##  [3,]  0.48225146 -0.4453748 -0.3059891  0.6871656
##  [4,] -1.35983100  1.1241093 -1.0467165  1.0253160
##  [5,]  0.09531093 -0.4788558  0.1460804  0.3307137
##  [6,]  1.39979103  0.1385598  0.5164438  1.3155267
##  [7,] -1.01212841  1.1557534  1.1415156  1.9817401
##  [8,] -0.23164905  0.8695089  1.1611847  0.2721394
##  [9,] -0.20441813  1.4244346  0.3664593  1.9576783
## [10,] -1.37867638  1.6442721 -0.3191072  0.9006680
```

Wybrane operacje na elementach macierzy

```
m5 <- abs(m4)
m5
##             [,1]      [,2]      [,3]       [,4]      [,5]      [,6]       [,7]
##  [1,] 1.02567581 1.2610999 0.9645456 0.14697234 0.1488480 0.1742230 0.62346827
##  [2,] 1.39794740 0.4888950 0.2319919 2.28536288 1.9900652 1.2988819 0.64386063
##  [3,] 0.27722498 2.0774252 0.4439360 0.68557787 0.3846282 0.8475726 0.48225146
##  [4,] 1.10662077 0.6325807 0.9611398 0.97144367 0.6215753 0.7172027 1.35983100
##  [5,] 0.02869781 0.6153722 0.2717784 0.37090831 0.8448948 2.4294826 0.09531093
##  [6,] 1.17652080 0.2178309 0.5567940 0.78042505 0.3388594 0.7946059 1.39979103
##  [7,] 0.28024038 1.0977501 0.9544588 0.78076765 0.4085061 0.2456427 1.01212841
##  [8,] 0.26914415 1.6869653 1.2734856 0.02001303 0.2182018 1.1573435 0.23164905
##  [9,] 0.31834798 0.4591093 0.1621171 0.50673982 0.1139678 0.7072789 0.20441813
## [10,] 0.95025797 1.0074917 0.5087157 2.07862947 0.9088694 1.2824386 1.37867638
##            [,8]      [,9]     [,10]
##  [1,] 1.1959404 1.0773430 1.3034006
##  [2,] 0.2307029 0.3743702 0.6246145
```

```
##  [3,] 0.4453748 0.3059891 0.6871656
##  [4,] 1.1241093 1.0467165 1.0253160
##  [5,] 0.4788558 0.1460804 0.3307137
##  [6,] 0.1385598 0.5164438 1.3155267
##  [7,] 1.1557534 1.1415156 1.9817401
##  [8,] 0.8695089 1.1611847 0.2721394
##  [9,] 1.4244346 0.3664593 1.9576783
## [10,] 1.6442721 0.3191072 0.9006680
```

```
m6 <- m4^2
m6
##               [,1]       [,2]       [,3]         [,4]       [,5]       [,6]
##  [1,] 1.0520108669 1.59037289 0.93034812 0.0216008697 0.02215573 0.03035366
##  [2,] 1.9542569203 0.23901833 0.05382025 5.2228834789 3.96035937 1.68709426
##  [3,] 0.0768536875 4.31569537 0.19707916 0.4700170193 0.14793882 0.71837937
##  [4,] 1.2246095363 0.40015830 0.92378972 0.9437028076 0.38635587 0.51437971
##  [5,] 0.0008235644 0.37868296 0.07386351 0.1375729756 0.71384726 5.90238552
##  [6,] 1.3842011869 0.04745029 0.31001958 0.6090632616 0.11482571 0.63139861
##  [7,] 0.0785346718 1.20505533 0.91099151 0.6095981240 0.16687723 0.06034035
##  [8,] 0.0724385744 2.84585184 1.62176545 0.0004005215 0.04761201 1.33944392
##  [9,] 0.1013454348 0.21078137 0.02628196 0.2567852401 0.01298866 0.50024347
## [10,] 0.9029902170 1.01503955 0.25879166 4.3207004841 0.82604353 1.64464879
##               [,7]       [,8]      [,9]      [,10]
##  [1,] 0.388712681 1.43027345 1.1606680 1.69885302
##  [2,] 0.414556508 0.05322383 0.1401530 0.39014330
##  [3,] 0.232566475 0.19835870 0.0936293 0.47219656
##  [4,] 1.849140361 1.26362173 1.0956155 1.05127295
##  [5,] 0.009084174 0.22930292 0.0213395 0.10937154
##  [6,] 1.959414917 0.01919880 0.2667142 1.73061040
##  [7,] 1.024403910 1.33576603 1.3030579 3.92729400
##  [8,] 0.053661280 0.75604572 1.3483499 0.07405987
##  [9,] 0.041786771 2.02901400 0.1342924 3.83250417
## [10,] 1.900748573 2.70363062 0.1018294 0.81120281
```

```
m7 <- m4 * m4
m7
##               [,1]       [,2]       [,3]         [,4]       [,5]       [,6]
##  [1,] 1.0520108669 1.59037289 0.93034812 0.0216008697 0.02215573 0.03035366
##  [2,] 1.9542569203 0.23901833 0.05382025 5.2228834789 3.96035937 1.68709426
##  [3,] 0.0768536875 4.31569537 0.19707916 0.4700170193 0.14793882 0.71837937
##  [4,] 1.2246095363 0.40015830 0.92378972 0.9437028076 0.38635587 0.51437971
##  [5,] 0.0008235644 0.37868296 0.07386351 0.1375729756 0.71384726 5.90238552
##  [6,] 1.3842011869 0.04745029 0.31001958 0.6090632616 0.11482571 0.63139861
##  [7,] 0.0785346718 1.20505533 0.91099151 0.6095981240 0.16687723 0.06034035
##  [8,] 0.0724385744 2.84585184 1.62176545 0.0004005215 0.04761201 1.33944392
##  [9,] 0.1013454348 0.21078137 0.02628196 0.2567852401 0.01298866 0.50024347
## [10,] 0.9029902170 1.01503955 0.25879166 4.3207004841 0.82604353 1.64464879
##               [,7]       [,8]      [,9]      [,10]
##  [1,] 0.388712681 1.43027345 1.1606680 1.69885302
##  [2,] 0.414556508 0.05322383 0.1401530 0.39014330
##  [3,] 0.232566475 0.19835870 0.0936293 0.47219656
##  [4,] 1.849140361 1.26362173 1.0956155 1.05127295
##  [5,] 0.009084174 0.22930292 0.0213395 0.10937154
##  [6,] 1.959414917 0.01919880 0.2667142 1.73061040
```

```
##  [7,] 1.024403910 1.33576603 1.3030579 3.92729400
##  [8,] 0.053661280 0.75604572 1.3483499 0.07405987
##  [9,] 0.041786771 2.02901400 0.1342924 3.83250417
## [10,] 1.900748573 2.70363062 0.1018294 0.81120281
```

Wybrane operacje na macierzach

```
m1 %*% m1
##      [,1] [,2]
## [1,]    7   15
## [2,]   10   22
```

```
colSums(m7)
##  [1]  6.848065 12.248106  5.306751 12.592325  6.399004 13.028668  7.874076
##  [8] 10.018436  5.665649 14.097509
rowSums(m7)
##  [1]  8.325349 14.115509  6.922714  9.652646  7.576274  7.072897 10.621919
##  [8]  8.159629  7.146023 14.485626
colMeans(m7)
##  [1] 0.6848065 1.2248106 0.5306751 1.2592325 0.6399004 1.3028668 0.7874076
##  [8] 1.0018436 0.5665649 1.4097509
rowMeans(m7)
##  [1] 0.8325349 1.4115509 0.6922714 0.9652646 0.7576274 0.7072897 1.0621919
##  [8] 0.8159629 0.7146023 1.4485626
```

Ramki danych

```
df1 <- data.frame(lzp = rnorm(10),
                  lc = sample(1:100, 10),
                  wl = sample(c(TRUE, FALSE), 10, replace = TRUE),
                  zf = sample(factor(c("K", "M")), 10, replace = TRUE))
df1
##              lzp lc    wl zf
## 1  -1.2864243505 89  TRUE  M
## 2   0.0856667692 88 FALSE  M
## 3  -0.9662987150 46 FALSE  K
## 4  -0.7160019574 80 FALSE  M
## 5  -0.3098307975 81  TRUE  K
## 6   0.0015849022 76 FALSE  M
## 7  -0.6919053070 38 FALSE  M
## 8  -0.7230857017 52 FALSE  K
## 9   0.0003839343 26 FALSE  K
## 10 -0.0051861011 69  TRUE  K
```

```
df2 <- data.frame(x1 = rep('a', 4), numbers = 1:4, logic = sample(c(TRUE, FALSE), 4, replace = TRUE))
df2
##   x1 numbers logic
## 1  a       1  TRUE
## 2  a       2 FALSE
## 3  a       3 FALSE
## 4  a       4 FALSE
```

Wektory (działania na nich, porównywanie, konstrukcja)

Działania na wektorach

```
v5 <- c(1:2)
v5 + v5
## [1] 2 4
```

```
v3 / 2
## [1] 0.5 0.0
```

```
2 * TRUE
## [1] 2
2 * FALSE
## [1] 0
3 == TRUE
## [1] FALSE
```

Wektory logiczne & porównywanie

```
v6 <- c(TRUE, FALSE, FALSE)
v7 <- c(TRUE, FALSE, TRUE)
v6 == v7
## [1]  TRUE  TRUE FALSE
v6 != v7
## [1] FALSE FALSE  TRUE
v6 <= v7
## [1] TRUE TRUE TRUE
v6 < v7
## [1] FALSE FALSE  TRUE
v6 & v7
## [1]  TRUE FALSE FALSE
v6 | v7
## [1]  TRUE FALSE  TRUE

all(v6)
## [1] FALSE
any(v6)
## [1] TRUE

sum(v6)
## [1] 1
```

Konstrukcje specjalne wektorów liczbowych

Wektory liczbowe konstruowane przez funkcję *seq*

```
v8 <- seq(from = 1, to = 4, by = .5)
v9 <- seq(1, 4, .5)
v8 == v9
## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

Wektory liczbowe konstruowane przez funkcję **:**

```r
1:4
## [1] 1 2 3 4
4:2
## [1] 4 3 2
-1:3
## [1] -1  0  1  2  3
-1:-3
## [1] -1 -2 -3
```

Operacje na wektorach; reguła zawijania.

```r
(v8 + v9)
## [1] 2 3 4 5 6 7 8
(v8 + a3)
## [1] 3.0 3.5 4.0 4.5 5.0 5.5 6.0
(v8 + v6)
## Warning in v8 + v6: długość dłuszego obiektu nie jest wielokrotnością długości
## krótszego obiektu
## [1] 2.0 1.5 2.0 3.5 3.0 3.5 5.0
```

Pomoc w R

```r
?seq
help(seq)
```

Operacje matematyczne (wybrane)

```r
ceiling(v8)
## [1] 1 2 2 3 3 4 4
floor(v8)
## [1] 1 1 2 2 3 3 4
log(v8)
## [1] 0.0000000 0.4054651 0.6931472 0.9162907 1.0986123 1.2527630 1.3862944
```

Składanie operacji

```r
sum(log(v8))
## [1] 5.752573
```

Wbudowany operator potoku

```r
v8 |>
  log() |>
  sum()
## [1] 5.752573
```

Operator potoku z *magrittr*

```r
library(tidyverse)
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
```

```
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## 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 error
v8 %>%
  log() %>%
  sum()
## [1] 5.752573
```

Wydzielanie podzbiorów/elementów

```
x <- c(2.1, 4.2, 3.3, 5.4)
```

- Nieujemne indeksy

```
x[c(3, 1)]
## [1] 3.3 2.1
x[order(x)]
## [1] 2.1 3.3 4.2 5.4
x[c(1, 1)]
## [1] 2.1 2.1
x[c(2.1, 2.9)]
## [1] 4.2 4.2
```

- Ujemne indeksy

```
x[-c(3, 1)]
## [1] 4.2 5.4
```

- Wartości logiczne

```
x[c(TRUE, FALSE)]
## [1] 2.1 3.3
x[c(TRUE, FALSE, TRUE, FALSE)]
## [1] 2.1 3.3
```

```
x[c(TRUE, TRUE, NA, FALSE)]
## [1] 2.1 4.2  NA
```

- Wydzielanie za pomocą pustego pola

```
x[]
## [1] 2.1 4.2 3.3 5.4
```

- Wydzielanie za pomocą zera

```
x[0]
## numeric(0)
```

- Wydzielanie za pomocą nazw elementów wektora

```
(y <- setNames(x, letters[1:4]))
##   a   b   c   d
## 2.1 4.2 3.3 5.4
y[c("d", "c", "a")]
##   d   c   a
## 5.4 3.3 2.1

y[c("a", "a", "a")]
##   a   a   a
## 2.1 2.1 2.1
```

- Wydzielanie w macierzach

```
a <- matrix(1:9, nrow = 3)
colnames(a) <- c("A", "B", "C")
a[1:2, ]
##      A B C
## [1,] 1 4 7
## [2,] 2 5 8
a[c(TRUE, FALSE, TRUE), c("B", "A")]
##      B A
## [1,] 4 1
## [2,] 6 3
a[0, -2]
##      A C
```

```
(vals <- outer(1:5, 1:5, FUN = "paste", sep = ","))
##      [,1]  [,2]  [,3]  [,4]  [,5]
## [1,] "1,1" "1,2" "1,3" "1,4" "1,5"
## [2,] "2,1" "2,2" "2,3" "2,4" "2,5"
## [3,] "3,1" "3,2" "3,3" "3,4" "3,5"
## [4,] "4,1" "4,2" "4,3" "4,4" "4,5"
## [5,] "5,1" "5,2" "5,3" "5,4" "5,5"

vals[c(4, 15)]
## [1] "4,1" "5,3"
```

```
select <- matrix(ncol = 2, byrow = TRUE, c(
  1, 1,
  3, 1,
  2, 4
))
vals[select]
## [1] "1,1" "3,1" "2,4"
```

- Wydzielanie w ramkach danych

```
df <- data.frame(x = 1:3, y = 3:1, z = letters[1:3])

df[df$x == 2, ]
##   x y z
## 2 2 2 b
df[c(1, 3), ]
##   x y z
## 1 1 3 a
## 3 3 1 c
```

```
df[c("x", "z")]
##   x z
## 1 1 a
## 2 2 b
## 3 3 c
df[, c("x", "z")]
##   x z
## 1 1 a
## 2 2 b
## 3 3 c
```

```
var <- "cyl"
mtcars$var
## NULL
```

```
mtcars[[var]]
##  [1] 6 6 4 6 8 6 8 4 4 6 6 8 8 8 8 8 8 4 4 4 4 8 8 8 8 4 4 4 8 6 8 4
```

Rozkłady prawdopodobieństwa (gęstość, dystrybuanta, kwantyle, liczby pseudolosowe)

```
?pnorm
help(pnorm)
```

Tablica wartości dystrybuanty rozkładu normalnego standardowego.

```
DystrybuantaNormalnego <- seq(0, 3.99, .01) |>
  pnorm() |>
  round(digits = 9) |>
  matrix(ncol = 10, byrow = TRUE) |>
  `colnames<-`(format(seq(0, .09, .01), nsmall = 2)) |>
  `rownames<-`(format(seq(0, 3.9, .1), nsmall = 1))

DystrybuantaNormalnego
##          0.00      0.01      0.02      0.03      0.04      0.05      0.06
## 0.0 0.5000000 0.5039894 0.5079783 0.5119665 0.5159534 0.5199388 0.5239222
## 0.1 0.5398278 0.5437953 0.5477584 0.5517168 0.5556700 0.5596177 0.5635595
## 0.2 0.5792597 0.5831662 0.5870644 0.5909541 0.5948349 0.5987063 0.6025681
## 0.3 0.6179114 0.6217195 0.6255158 0.6293000 0.6330717 0.6368307 0.6405764
## 0.4 0.6554217 0.6590970 0.6627573 0.6664022 0.6700314 0.6736448 0.6772419
## 0.5 0.6914625 0.6949743 0.6984682 0.7019440 0.7054015 0.7088403 0.7122603
## 0.6 0.7257469 0.7290691 0.7323711 0.7356527 0.7389137 0.7421539 0.7453731
## 0.7 0.7580363 0.7611479 0.7642375 0.7673049 0.7703500 0.7733726 0.7763727
## 0.8 0.7881446 0.7910299 0.7938919 0.7967306 0.7995458 0.8023375 0.8051055
## 0.9 0.8159399 0.8185887 0.8212136 0.8238145 0.8263912 0.8289439 0.8314724
## 1.0 0.8413447 0.8437524 0.8461358 0.8484950 0.8508301 0.8531409 0.8554277
## 1.1 0.8643339 0.8665005 0.8686431 0.8707619 0.8728568 0.8749281 0.8769756
## 1.2 0.8849303 0.8868606 0.8887676 0.8906514 0.8925123 0.8943502 0.8961653
## 1.3 0.9031995 0.9049021 0.9065825 0.9082409 0.9098773 0.9114920 0.9130850
## 1.4 0.9192433 0.9207302 0.9221962 0.9236415 0.9250663 0.9264707 0.9278550
## 1.5 0.9331928 0.9344783 0.9357445 0.9369916 0.9382198 0.9394292 0.9406201
## 1.6 0.9452007 0.9463011 0.9473839 0.9484493 0.9494974 0.9505285 0.9515428
## 1.7 0.9554345 0.9563671 0.9572838 0.9581849 0.9590705 0.9599408 0.9607961
```

```
## 1.8 0.9640697 0.9648521 0.9656205 0.9663750 0.9671159 0.9678432 0.9685572
## 1.9 0.9712834 0.9719334 0.9725710 0.9731966 0.9738102 0.9744119 0.9750021
## 2.0 0.9772499 0.9777844 0.9783083 0.9788217 0.9793248 0.9798178 0.9803007
## 2.1 0.9821356 0.9825708 0.9829970 0.9834142 0.9838226 0.9842224 0.9846137
## 2.2 0.9860966 0.9864474 0.9867906 0.9871263 0.9874545 0.9877755 0.9880894
## 2.3 0.9892759 0.9895559 0.9898296 0.9900969 0.9903581 0.9906133 0.9908625
## 2.4 0.9918025 0.9920237 0.9922397 0.9924506 0.9926564 0.9928572 0.9930531
## 2.5 0.9937903 0.9939634 0.9941323 0.9942969 0.9944574 0.9946139 0.9947664
## 2.6 0.9953388 0.9954729 0.9956035 0.9957308 0.9958547 0.9959754 0.9960930
## 2.7 0.9965330 0.9966358 0.9967359 0.9968333 0.9969280 0.9970202 0.9971099
## 2.8 0.9974449 0.9975229 0.9975988 0.9976726 0.9977443 0.9978140 0.9978818
## 2.9 0.9981342 0.9981929 0.9982498 0.9983052 0.9983589 0.9984111 0.9984618
## 3.0 0.9986501 0.9986938 0.9987361 0.9987772 0.9988171 0.9988558 0.9988933
## 3.1 0.9990324 0.9990646 0.9990957 0.9991260 0.9991553 0.9991836 0.9992112
## 3.2 0.9993129 0.9993363 0.9993590 0.9993810 0.9994024 0.9994230 0.9994429
## 3.3 0.9995166 0.9995335 0.9995499 0.9995658 0.9995811 0.9995959 0.9996103
## 3.4 0.9996631 0.9996752 0.9996869 0.9996982 0.9997091 0.9997197 0.9997299
## 3.5 0.9997674 0.9997759 0.9997842 0.9997922 0.9997999 0.9998074 0.9998146
## 3.6 0.9998409 0.9998469 0.9998527 0.9998583 0.9998637 0.9998689 0.9998739
## 3.7 0.9998922 0.9998964 0.9999004 0.9999043 0.9999080 0.9999116 0.9999150
## 3.8 0.9999277 0.9999305 0.9999333 0.9999359 0.9999385 0.9999409 0.9999433
## 3.9 0.9999519 0.9999539 0.9999557 0.9999575 0.9999593 0.9999609 0.9999625
##          0.07      0.08      0.09
## 0.0 0.5279032 0.5318814 0.5358564
## 0.1 0.5674949 0.5714237 0.5753454
## 0.2 0.6064199 0.6102612 0.6140919
## 0.3 0.6443088 0.6480273 0.6517317
## 0.4 0.6808225 0.6843863 0.6879331
## 0.5 0.7156612 0.7190427 0.7224047
## 0.6 0.7485711 0.7517478 0.7549029
## 0.7 0.7793501 0.7823046 0.7852361
## 0.8 0.8078498 0.8105703 0.8132671
## 0.9 0.8339768 0.8364569 0.8389129
## 1.0 0.8576903 0.8599289 0.8621434
## 1.1 0.8789995 0.8809999 0.8829768
## 1.2 0.8979577 0.8997274 0.9014747
## 1.3 0.9146565 0.9162067 0.9177356
## 1.4 0.9292191 0.9305634 0.9318879
## 1.5 0.9417924 0.9429466 0.9440826
## 1.6 0.9525403 0.9535213 0.9544860
## 1.7 0.9616364 0.9624620 0.9632730
## 1.8 0.9692581 0.9699460 0.9706210
## 1.9 0.9755808 0.9761482 0.9767045
## 2.0 0.9807738 0.9812372 0.9816911
## 2.1 0.9849966 0.9853713 0.9857379
## 2.2 0.9883962 0.9886962 0.9889893
## 2.3 0.9911060 0.9913437 0.9915758
## 2.4 0.9932443 0.9934309 0.9936128
## 2.5 0.9949151 0.9950600 0.9952012
## 2.6 0.9962074 0.9963189 0.9964274
## 2.7 0.9971972 0.9972821 0.9973646
## 2.8 0.9979476 0.9980116 0.9980738
## 2.9 0.9985110 0.9985588 0.9986051
```

```
## 3.0 0.9989297 0.9989650 0.9989992
## 3.1 0.9992378 0.9992636 0.9992886
## 3.2 0.9994623 0.9994810 0.9994991
## 3.3 0.9996242 0.9996376 0.9996505
## 3.4 0.9997398 0.9997493 0.9997585
## 3.5 0.9998215 0.9998282 0.9998347
## 3.6 0.9998787 0.9998834 0.9998879
## 3.7 0.9999184 0.9999216 0.9999247
## 3.8 0.9999456 0.9999478 0.9999499
## 3.9 0.9999641 0.9999655 0.9999670
```

Dane i ich przekształcenia (zmienne kategoryczne/czynnikowe).

Korzystamy z danych Jareda P. Landera [3].

```
require(tidyverse)

acsNew <- read_csv("http://www.jaredlander.com/data/acsNew.csv")
```

```
## Rows: 2273 Columns: 19
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (9): Acres, FamilyType, NumUnits, OwnRent, YearBuilt, FoodStamp, Heatin...
## dbl (10): FamilyIncome, NumBedrooms, NumChildren, NumPeople, NumRooms, NumVe...
##
## 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.
```

```
summary(acsNew)
##     Acres            FamilyIncome      FamilyType         NumBedrooms
##  Length:2273        Min.   :   1125   Length:2273        Min.   :0.000
##  Class :character   1st Qu.:  53700   Class :character   1st Qu.:3.000
##  Mode  :character   Median :  89200   Mode  :character   Median :3.000
##                     Mean   : 110982                      Mean   :3.404
##                     3rd Qu.: 136630                      3rd Qu.:4.000
##                     Max.   :1014000                      Max.   :8.000
##   NumChildren       NumPeople         NumRooms        NumUnits
##  Min.   :0.0000   Min.   : 2.000   Min.   : 1.000   Length:2273
##  1st Qu.:0.0000   1st Qu.: 2.000   1st Qu.: 6.000   Class :character
##  Median :0.0000   Median : 3.000   Median : 7.000   Mode  :character
##  Mean   :0.8847   Mean   : 3.401   Mean   : 7.241
##  3rd Qu.:2.0000   3rd Qu.: 4.000   3rd Qu.: 8.000
##  Max.   :8.0000   Max.   :12.000   Max.   :21.000
##   NumVehicles      NumWorkers      OwnRent           YearBuilt
##  Min.   :0.000   Min.   :0.000   Length:2273        Length:2273
##  1st Qu.:2.000   1st Qu.:1.000   Class :character   Class :character
##  Median :2.000   Median :2.000   Mode  :character   Mode  :character
##  Mean   :2.118   Mean   :1.778
##  3rd Qu.:3.000   3rd Qu.:2.000
##  Max.   :6.000   Max.   :3.000
##    HouseCosts    ElectricBill    FoodStamp         HeatingFuel
##  Min.   :   4   Min.   :  1.0   Length:2273        Length:2273
##  1st Qu.: 670   1st Qu.:100.0   Class :character   Class :character
##  Median :1200   Median :150.0   Mode  :character   Mode  :character
##  Mean   :1488   Mean   :176.6
```

```
##  3rd Qu.:2000   3rd Qu.:220.0
##  Max.   :6500   Max.   :580.0
##    Insurance        Language            Income
##  Min.   :   0.0   Length:2273        Length:2273
##  1st Qu.: 400.0   Class :character   Class :character
##  Median : 720.0   Mode  :character   Mode  :character
##  Mean   : 968.1
##  3rd Qu.:1200.0
##  Max.   :6600.0
```

```
acsNewFactor <- acsNew |>
  mutate(across(where(is.character), ~ as.factor(.x), .names = "{.col}.factor"))

summary(acsNewFactor)
```

```
##     Acres            FamilyIncome      FamilyType         NumBedrooms
##  Length:2273        Min.   :   1125   Length:2273        Min.   :0.000
##  Class :character   1st Qu.:  53700   Class :character   1st Qu.:3.000
##  Mode  :character   Median :  89200   Mode  :character   Median :3.000
##                     Mean   : 110982                      Mean   :3.404
##                     3rd Qu.: 136630                      3rd Qu.:4.000
##                     Max.   :1014000                      Max.   :8.000
##
##   NumChildren       NumPeople         NumRooms        NumUnits
##  Min.   :0.0000   Min.   : 2.000   Min.   : 1.000   Length:2273
##  1st Qu.:0.0000   1st Qu.: 2.000   1st Qu.: 6.000   Class :character
##  Median :0.0000   Median : 3.000   Median : 7.000   Mode  :character
##  Mean   :0.8847   Mean   : 3.401   Mean   : 7.241
##  3rd Qu.:2.0000   3rd Qu.: 4.000   3rd Qu.: 8.000
##  Max.   :8.0000   Max.   :12.000   Max.   :21.000
##
##   NumVehicles      NumWorkers      OwnRent           YearBuilt
##  Min.   :0.000   Min.   :0.000   Length:2273        Length:2273
##  1st Qu.:2.000   1st Qu.:1.000   Class :character   Class :character
##  Median :2.000   Median :2.000   Mode  :character   Mode  :character
##  Mean   :2.118   Mean   :1.778
##  3rd Qu.:3.000   3rd Qu.:2.000
##  Max.   :6.000   Max.   :3.000
##
##    HouseCosts    ElectricBill    FoodStamp         HeatingFuel
##  Min.   :   4   Min.   :  1.0   Length:2273        Length:2273
##  1st Qu.: 670   1st Qu.:100.0   Class :character   Class :character
##  Median :1200   Median :150.0   Mode  :character   Mode  :character
##  Mean   :1488   Mean   :176.6
##  3rd Qu.:2000   3rd Qu.:220.0
##  Max.   :6500   Max.   :580.0
##
##    Insurance        Language            Income          Acres.factor
##  Min.   :   0.0   Length:2273        Length:2273        1-10 : 452
##  1st Qu.: 400.0   Class :character   Class :character   10+  :  90
##  Median : 720.0   Mode  :character   Mode  :character   Sub 1:1731
##  Mean   : 968.1
##  3rd Qu.:1200.0
##  Max.   :6600.0
```

```
##
##    FamilyType.factor        NumUnits.factor  OwnRent.factor    YearBuilt.factor
##  Female Head: 327    Mobile home    :  67   Mortgage:2008   Before 1939:588
##  Male Head  : 115    Single attached: 235   Outright:  15   1950-1959  :423
##  Married    :1831    Single detached:1971   Rented  : 250   1960-1969  :269
##                                                             1970-1979  :229
##                                                             1990-1999  :198
##                                                             1980-1989  :195
##                                                             (Other)    :371
##  FoodStamp.factor   HeatingFuel.factor       Language.factor Income.factor
##  No :2106         Coal       :  16     Asian Pacific :  62   Above: 456
##  Yes: 167         Electricity: 109     English       :1786   Below:1817
##                   Gas        :1387     Other         :  39
##                   None       :   4     Other European: 219
##                   Oil        : 622     Spanish       : 167
##                   Other      :  18
##                   Wood       : 117

acsNewFactor1 <- acsNew |>
  mutate(across(where(is.character), ~ as.factor(.x)))

summary(acsNewFactor1)
##    Acres       FamilyIncome           FamilyType    NumBedrooms
##  1-10 : 452   Min.   :   1125   Female Head: 327   Min.   :0.000
##  10+  :  90   1st Qu.:  53700   Male Head  : 115   1st Qu.:3.000
##  Sub 1:1731   Median :  89200   Married    :1831   Median :3.000
##               Mean   : 110982                      Mean   :3.404
##               3rd Qu.: 136630                      3rd Qu.:4.000
##               Max.   :1014000                      Max.   :8.000
##
##   NumChildren       NumPeople         NumRooms                NumUnits
##  Min.   :0.0000   Min.   : 2.000   Min.   : 1.000   Mobile home    :  67
##  1st Qu.:0.0000   1st Qu.: 2.000   1st Qu.: 6.000   Single attached: 235
##  Median :0.0000   Median : 3.000   Median : 7.000   Single detached:1971
##  Mean   :0.8847   Mean   : 3.401   Mean   : 7.241
##  3rd Qu.:2.0000   3rd Qu.: 4.000   3rd Qu.: 8.000
##  Max.   :8.0000   Max.   :12.000   Max.   :21.000
##
##   NumVehicles      NumWorkers        OwnRent           YearBuilt
##  Min.   :0.000   Min.   :0.000   Mortgage:2008   Before 1939:588
##  1st Qu.:2.000   1st Qu.:1.000   Outright:  15   1950-1959  :423
##  Median :2.000   Median :2.000   Rented  : 250   1960-1969  :269
##  Mean   :2.118   Mean   :1.778                   1970-1979  :229
##  3rd Qu.:3.000   3rd Qu.:2.000                   1990-1999  :198
##  Max.   :6.000   Max.   :3.000                   1980-1989  :195
##                                                  (Other)    :371
##    HouseCosts    ElectricBill   FoodStamp       HeatingFuel     Insurance
##  Min.   :   4   Min.   :  1.0   No :2106   Coal       :  16   Min.   :   0.0
##  1st Qu.: 670   1st Qu.:100.0   Yes: 167   Electricity: 109   1st Qu.: 400.0
##  Median :1200   Median :150.0              Gas        :1387   Median : 720.0
##  Mean   :1488   Mean   :176.6              None       :   4   Mean   : 968.1
##  3rd Qu.:2000   3rd Qu.:220.0              Oil        : 622   3rd Qu.:1200.0
##  Max.   :6500   Max.   :580.0              Other      :  18   Max.   :6600.0
```

```
##                                           Wood       : 117
##            Language      Income
##  Asian Pacific :  62   Above: 456
##  English       :1786   Below:1817
##  Other         :  39
##  Other European: 219
##  Spanish       : 167
##
##
```

Zadania

1. Ustaw ziarno generatora liczb pseudolosowych na 2024. Wylosuj wektor 20 liczb, liczby całkowite mają być losowane z zakresu od 1 do 3, z prawdopodobieństwami wylosowania (0.1, 0.3, 0.6).

```
##  [1] 2 3 2 2 3 2 3 3 2 3 1 1 2 3 3 2 3 2 3 3
```

2. Wykorzystaj instrukcje warunkowe (*if (WARUNEK) {AKCJA 1} else {AKCJA 2}*), aby sprawdzić czy liczba 1515 jest parzysta. Jeśli liczba jest parzysta to wyświetl stosowny komunikat, oraz jeśli liczba jest nieparzysta to również wyświetl stosowny komunikat.

```
## [1] "Liczba jest nieparzysta."
```

3. Wykorzystując wyłącznie operatory: wydzielania, arytmetyczne i porównania wyświetl na ekranie wszystkie liczby całkowite podzielne przez 3 bez reszty należące do zadanego przedziału [3,333]. Przedstaw też rozwiązania wykorzystujące wyłącznie *seq* oraz wyłącznie pętle *for*.

```
##   [1]   3   6   9  12  15  18  21  24  27  30  33  36  39  42  45  48  51  54
##  [19]  57  60  63  66  69  72  75  78  81  84  87  90  93  96  99 102 105 108
##  [37] 111 114 117 120 123 126 129 132 135 138 141 144 147 150 153 156 159 162
##  [55] 165 168 171 174 177 180 183 186 189 192 195 198 201 204 207 210 213 216
##  [73] 219 222 225 228 231 234 237 240 243 246 249 252 255 258 261 264 267 270
##  [91] 273 276 279 282 285 288 291 294 297 300 303 306 309 312 315 318 321 324
## [109] 327 330 333
```

Rozwiązanie z użyciem *seq*:

```
##   [1]   3   6   9  12  15  18  21  24  27  30  33  36  39  42  45  48  51  54
##  [19]  57  60  63  66  69  72  75  78  81  84  87  90  93  96  99 102 105 108
##  [37] 111 114 117 120 123 126 129 132 135 138 141 144 147 150 153 156 159 162
##  [55] 165 168 171 174 177 180 183 186 189 192 195 198 201 204 207 210 213 216
##  [73] 219 222 225 228 231 234 237 240 243 246 249 252 255 258 261 264 267 270
##  [91] 273 276 279 282 285 288 291 294 297 300 303 306 309 312 315 318 321 324
## [109] 327 330 333
```

Rozwiązanie z użyciem pętli *for*:

```
##   [1]   3   6   9  12  15  18  21  24  27  30  33  36  39  42  45  48  51  54
##  [19]  57  60  63  66  69  72  75  78  81  84  87  90  93  96  99 102 105 108
##  [37] 111 114 117 120 123 126 129 132 135 138 141 144 147 150 153 156 159 162
##  [55] 165 168 171 174 177 180 183 186 189 192 195 198 201 204 207 210 213 216
##  [73] 219 222 225 228 231 234 237 240 243 246 249 252 255 258 261 264 267 270
##  [91] 273 276 279 282 285 288 291 294 297 300 303 306 309 312 315 318 321 324
## [109] 327 330 333
```

4. Ustaw ziarno generatora liczb pseudolosowych na 2024. Utwórz ramkę danych, która w pierwszej kolumnie (*id*) przechowuje id od 1 do 10, w drugiej kolumnie (*kol2*) przechowuje 10 losowych wartości z

zakresu [1:10] bez zwracania, w trzeciej kolumnie (*kol3*) przechowuje 10 losowych wartości z zakresu [10:20] ze zwracaniem, następnie zwróć wartość średnią dla kolumny 2 i 3 z wykorzystaniem
   - pętli,
   - metod/funkcji z pakietów **tidyverse**.

Mają Państwo otrzymać poniższą ramkę danych:

```
##    id kol2 kol3
## 1   1    2   20
## 2   2    5   11
## 3   3    9   19
## 4   4    4   11
## 5   5    1   19
## 6   6    8   14
## 7   7    7   10
## 8   8    6   20
## 9   9   10   12
## 10 10    3   18
```

Rozwiązanie z wykorzystaniem pętli:

```
## kol2 kol3
##  5.5 15.4
```

Rozwiązanie z wykorzystaniem metod/funkcji z pakietów **tidyverse**:

```
## kol2 kol3
##  5.5 15.4
```

5. Wyświetl dane z wierszy o parzystych indeksach z danych **acsNew**.

Wynikiem ma być ramka danych, której 10 pierwszych elementów to

```
## # A tibble: 10 x 19
##    Acres FamilyIncome FamilyType  NumBedrooms NumChildren NumPeople NumRooms
##    <chr>        <dbl> <chr>             <dbl>       <dbl>     <dbl>    <dbl>
##  1 Sub 1       124000 Married               3           3         5        9
##  2 Sub 1        36000 Married               5           2         6        8
##  3 Sub 1       105500 Married               2           0         3        5
##  4 Sub 1       190200 Married               5           0         2       13
##  5 Sub 1       123700 Married               3           2         4        7
##  6 Sub 1       230000 Married               4           0         5       10
##  7 Sub 1        34700 Female Head          4           0         2        9
##  8 Sub 1       120700 Married               3           1         3        6
##  9 1-10         46300 Married               4           2         5        8
## 10 Sub 1        53700 Female Head          3           0         2        5
## # i 12 more variables: NumUnits <chr>, NumVehicles <dbl>, NumWorkers <dbl>,
## #   OwnRent <chr>, YearBuilt <chr>, HouseCosts <dbl>, ElectricBill <dbl>,
## #   FoodStamp <chr>, HeatingFuel <chr>, Insurance <dbl>, Language <chr>,
## #   Income <chr>
```

6. Nie używając operatora potoku i wektorowość języka R, a wykorzystując pętle skonstruuj tablicę rozkładu normalnego.

Bibliografia

1. R Core Team (2024) R: A Language and Environment for Statistical Computing, Vienna, Austria, R Foundation for Statistical Computing.

2. Wickham H, Averick M, Bryan J, et al. (2019) Welcome to the tidyverse. *Journal of Open Source Software* 4: 1686.
3. Lander JP (2020) Data.