



Technologia i rozwiązania

Analiza statystyczna w środowisku R dla początkujących

Dowiedz się, jak zwiększyć skuteczność analiz danych dzięki R!

- Jak rozpocząć przygodę ze środowiskiem R?
- Jak przeprowadzić analizę w środowisku R?
- Jak przygotować elegancki wykres?

Helion



John M. Quick



open source*
community, respect for the user

» Idź do

- Spis treści
- Przykładowy rozdział
- Skorowidz

» Katalog książek

- Katalog online
- Zamów drukowany katalog

» Twój koszyk

- Dodaj do koszyka

» Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

» Czytelnia

- Fragmenty książek online

» Kontakt

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 32 230 98 63
e-mail: helion@helion.pl
© Helion 1991–2011

Analiza statystyczna w środowisku R dla początkujących

Autor: John M. Quick
Tłumaczenie: Jacek Janusz
ISBN: 978-83-246-3409-5
Tytuł oryginału: [Statistical Analysis with R](#)
Format: 170×230, stron: 280



Dowiedz się, jak zwiększyć skuteczność analiz danych dzięki R!

- Jak rozpocząć przygodę ze środowiskiem R?
- Jak przeprowadzić analizę w środowisku R?
- Jak przygotować elegancki wykres?

Na wyrażenie „analiza statystyczna” spora grupa studentów dostaje nerwowych drgawek, a fascynatom rozszerzają się oczy. Ta dziedzina nauki budzi skrajne emocje. Skomplikowana, czasochłonna, wymaga dużej ilości obliczeń, jednak dostarcza istotnych informacji, których właściwą interpretację można przekuć w sukces w różnych dziedzinach życia. Statystyka, tak jak każda inna dyscyplina, dzięki postępującej informatyzacji otrzymała ogromne wsparcie w postaci specjalistycznego oprogramowania. Ta książka poświęcona została środowisku R, którego możliwości w stosunku do ceny są naprawdę imponujące, ponieważ środowisko to jest darmowe i rozpowszechniane na licencji GPL!

Książka ta jest podręcznikiem dla osób, które chciałyby się zaprzyjaźnić z analizą statystyczną i środowiskiem R. W trakcie lektury dowiesz się, jak zainstalować i skonfigurować środowisko R, a potem w nim pracować. Nauczysz się importować dane ze źródeł zewnętrznych, korzystać z funkcji wieloargumentowych, obliczać średnie, standardowe odchylenia i wiele innych wartości, typowych dla statystyki. Ponadto błyskawicznie opanujesz sposoby modelowania za pomocą prostej i wielokrotnej regresji liniowej oraz zobaczysz, jak interpretować wyniki. Poznasz także zaawansowane mechanizmy tworzenia wykresów, grafów i diagramów oraz odkryjesz najlepsze źródła informacji na temat środowiska R. Brzmi interesująco? Przekonaj się sam!

- Instalacja środowiska R
- Elementarz użytkownika R
- Importowanie danych zewnętrznych
- Wykorzystanie zmiennych
- Wyświetlanie, zapamiętywanie i odczytywanie zawartości obszaru roboczego
- Funkcje wieloargumentowe i o zmiennej liczbie argumentów
- Średnie, standardowe odchylenia, zakresy
- Prosta i wielokrotna regresja liniowa
- Interpretacja zmiennych interakcyjnych
- Prognozowanie wyników za pomocą modeli regresji
- 5 kroków do przeprowadzenia analizy w środowisku R
- Tworzenie wykresów, grafów i diagramów
- Przydatne zasoby w sieci

Zaprzyjaźnij się z analizą statystyczną!

Spis treści

| | |
|---|-----------|
| O autorze | 5 |
| <hr/> | |
| O recenzentach | 7 |
| <hr/> | |
| Wprowadzenie | 9 |
| <hr/> | |
| Zagadnienia poruszane w książce | 9 |
| Czego potrzebujesz do przeczytania tej książki? | 10 |
| Dla kogo jest przeznaczona ta książka? | 10 |
| Konwencje | 11 |
| Pomoc techniczna | 12 |
| Rozdział 1. Odkrywanie narzędzia analizy danych | 15 |
| <hr/> | |
| Co to jest R? | 16 |
| Jakie są korzyści z używania środowiska R? | 16 |
| Dlaczego powinienem używać środowiska R? | 17 |
| Dlaczego powinienem przeczytać tę książkę? | 17 |
| Jakie zagadnienia zostaną poruszone w tej książce? | 18 |
| Podsumowanie | 25 |
| Rozdział 2. Przygotowanie środowiska R do bitwy | 27 |
| <hr/> | |
| Rozdział 3. Odkrywanie tajemniczego narzędzia analizy danych | 41 |
| <hr/> | |
| Opis magicznego kwadratu Zhuge Lianga | 42 |
| Podsumowanie | 48 |
| Rozdział 4. Gromadzenie i porządkowanie informacji | 49 |
| <hr/> | |

| | |
|---|------------|
| Rozdział 5. Ocenianie sytuacji | 69 |
| Analiza naszych danych | 71 |
| Regresja | 87 |
| Podsumowanie | 106 |
| Rozdział 6. Planowanie ataku | 107 |
| Przegląd modeli | 107 |
| Prognozowanie wyników za pomocą modeli regresji | 112 |
| Rozważania logistyczne | 120 |
| Ocenianie szans realizacji strategii | 124 |
| Podsumowanie | 128 |
| Rozdział 7. Przygotowywanie planów bitwy | 129 |
| Przegląd i doprecyzowanie pełnej analizy | 129 |
| Czynności wspólne dla wszystkich analiz przeprowadzanych w środowisku R | 147 |
| Podsumowanie | 152 |
| Rozdział 8. Odprawa u Imperatora | 153 |
| Wykresy, grafy i diagramy w środowisku R | 153 |
| Podsumowanie | 185 |
| Rozdział 9. Odprawa u generałów | 187 |
| Więcej wykresów, grafów i diagramów w środowisku R | 188 |
| Podsumowanie | 251 |
| Rozdział 10. Bądź mistrzem strategii | 253 |
| Zasoby dostępne w środowisku R | 253 |
| Zasoby środowiska R dostępne w internecie | 262 |
| Podsumowanie | 265 |
| Dodatek A. Sprawdzian — odpowiedzi na pytania | 267 |
| Skorowidz | 273 |

Gromadzenie i porządkowanie informacji

Gdy dzięki rozwiązaniu lamiglówki Zhuge Lianga, udało Ci się zaprezentować swoje umiejętności, jego pomocnik dostarczył Ci dokumenty opisujące zasoby armii Shu. Zawierają one dane o złocie, wyposażeniu oraz żołnierzach. Zanim przystąpisz do ich analizowania w środowisku R, musisz je najpierw odpowiednio przygotować i uporządkować. Dzięki temu Twoje kolejne działania będą bardziej zrozumiałe i sprawne.

W tym rozdziale skoncentrujemy się na gromadzeniu i porządkowaniu istniejących informacji. Podczas tych działań zapoznasz się z kilkoma nowymi technikami dostępnymi w środowisku R. Pod koniec tego rozdziału będziesz mógł wykonywać następujące czynności:

- Importować dane zewnętrzne do środowiska R.
- Używać zmiennych w celu porządkowania i przetwarzania swoich danych.
- Zarządzać obszarem roboczym R.

Czas na działanie — importowanie danych zewnętrznych

Naszym pierwszym zadaniem będzie wczytanie do środowiska R zewnętrznych danych z informacjami o zasobach. Aby je wykonać, otwórz konsolę R i postępuj zgodnie z poniższymi instrukcjami:

1. Zdefiniuj katalog roboczy środowiska R za pomocą funkcji `setwd(dir)`. Ścieżka użyta w poniższym kodzie jest przykładowa. Katalog roboczy w Twoim komputerze powinien zostać zdefiniowany w odpowiedniej dla niego lokalizacji.
 - > #zdefiniuj katalog roboczy środowiska R
 - > #zastąp przykładową ścieżkę taką, która jest odpowiednia dla Ciebie
 - > `setwd("C:/Users/RUser/Strateg/")`
2. Skopiuj plik `zasobyHanzhong.csv` do swojego katalogu roboczego. Plik ten zawiera informacje o zasobach dotyczących armii królestwa Shu, która obecnie stacjonuje w mieście Hanzhong.
3. Wczytaj plik z zasobami do środowiska R za pomocą polecenia `read.csv(file)`:
 - > #użyj polecenia `read.csv(file)`, by wczytać zewnętrzny plik z danymi do środowiska R
 - > #plik zawiera informacje o zasobach królestwa Shu znajdujących się w mieście Hanzhong w Chinach
 - > `read.csv("zasobyHanzhong.csv")`
4. Środowisko R odczyta i wyświetli zawartość tego pliku, a wynik będzie wyglądać jak na poniższym rysunku:

```
> read.csv("zasobyHanzhong.csv")
  Złoto Zapasy Żołnierze StanWyposażenia
1 1000000 1000000 100000 1
```

Dane te zawierają informacje, zgodnie z którymi Twoje siły w Hanzhong dysponują obecnie zasobami złota w liczbie 1 000 000 sztuk, zestawami zapasów także w liczbie 1 000 000 sztuk, a ponadto posiadają 100 000 żołnierzy oraz wyposażenie, które jest w doskonałym stanie.

Co się zdarzyło?

Po zdefiniowaniu katalogu roboczego zapoznałeś się z nową funkcją. Jej składnia różni się od składni poleceń, które do tej pory poznałeś.

read.csv(file)

Nazwa funkcji `read.csv(file)` zawiera znak kropki, który znajduje się między określeniem `read` a atrybutem `csv`. Termin `csv` „informuje” funkcję `read` o tym, że dane w naszym pliku składają się z wartości oddzielonych od siebie przecinkami. Ważne jest, aby wiedzieć, jakiego rodzaju funkcji `read` powinniśmy używać, ponieważ może istnieć kilka alternatywnych form tej funkcji, np. `read.S` lub `read.SPSS`.

Element `file` funkcji `read.csv(file)` jest podobny do składnika `dir` w funkcji `setwd(dir)`. Ponieważ umieściliśmy plik z danymi w naszym katalogu roboczym, argument `file` powinien zawierać jedynie nazwę pliku oraz jego rozszerzenie. Jeśli dane zostałyby umieszczone w innej lokalizacji, należałoby podać pełną ścieżkę dostępu do pliku.

Pliki z wartościami oddzielonymi przecinkami (csv)

W tej książce będziemy wykorzystywać pliki danych zawierające **wartości oddzielone przecinkami**, zwane inaczej plikami **CSV**. Jest to zalecany rodzaj plików służący do importowania danych do środowiska R. Powinieneś wiedzieć, że R może odczytywać dane z wielu rodzajów źródeł. Zazwyczaj będziesz mógł zaimportować dane z dowolnego źródła, którego właśnie używasz.

Sprawdzian

1. Jaka jest zasadnicza różnica między argumentami `dir` i `file`?
 - a) Argument `dir` zawiera ścieżkę, a argument `file` zawiera nazwę pliku.
 - b) Argument `dir` zawiera ścieżkę do katalogu, natomiast argument `file` zawiera ścieżkę do pliku.
 - c) Funkcje, których nazwy rozpoczynają się od `read`, używają argumentu `file`, natomiast funkcje, których nazwy rozpoczynają się od `set`, używają argumentu `dir`.
 - d) Nie ma różnicy między argumentami `dir` i `file`.

Czas na działanie — tworzenie i używanie zmiennych

Wczytanie danych do środowiska R pozwala Ci na ich wyświetlenie w konsoli oraz użycie w celu wykonania podręcznych obliczeń (jak uczyniliśmy w rozdziale 3.). Zasadniczo potrzebujesz jednak bardziej uporządkowanej i uniwersalnej metody pozwalającej na manipulowanie danymi. W tym celu możesz użyć zmiennych środowiska R. Zamiast jedynie wczytać do środowiska R nasz plik z zasobami, spróbujmy tym razem odczytać i zapisać dane w zmiennej:

1. Użyj następującego kodu w celu zapisania danych z pliku zasobów w zmiennej o nazwie `hanzhongResources`:

```
> #wczytaj dane z pliku zasobyHanzhong.csv do zmiennej o nazwie
↳hanzhongResources
> hanzhongResources <- read.csv("zasobyHanzhong.csv")
```

2. Zwróć uwagę na to, że środowisko R nie wyświetliło żadnego rezultatu po wykonaniu powyższej czynności i po prostu udostępniło nowy wiersz w konsoli. Aby sprawdzić, jaka jest zawartość naszej nowej zmiennej, musimy ją wywołać poprzez podanie jej nazwy w konsoli R.

```
> #wyświetl zawartość zmiennej hanzhongResources
> #zasoby królestwa Shu w mieście Hanzhong w Chinach
> hanzhongResources
```

3. Środowisko R wyświetli zawartość zmiennej:

```
> hanzhongResources
  Złoto Zapasy Żołnierze StanWyposażenia
1 1000000 1000000 100000 1
```

Prawdopodobnie zauważyłeś, że wywołanie zmiennej `hanzhongResources` powoduje powstanie takiego samego wyniku, jak wczytanie pierwotnego pliku CSV do środowiska R. Użycie zmiennej jest jednakże dużo bardziej efektywne, ponieważ nie musimy podawać całego polecenia `read.csv(file)` za każdym razem, gdy chcemy wyświetlić jej zawartość. Zamiast tego możemy po prostu wprowadzić nazwę zmiennej.

Co się zdarzyło?

W środowisku R utworzyłeś swoją pierwszą **zmienną** i jej użyłeś. Zmienne są niezbędne do przechowywania i przetwarzania danych. Gdy zamierzasz utworzyć zmienną w środowisku R, musisz za każdym razem postępować podobnie jak w ostatnim ćwiczeniu. Oto cztery etapy procesu tworzenia zmiennej:

1. Rozpocznij, podając nazwę zmiennej

W naszym poprzednim przykładzie ciąg znaków `hanzhongResources` był nazwą zmiennej. Nazwa powinna być pierwszym elementem, który pojawi się w nowym wierszu konsoli podczas tworzenia zmiennej w środowisku R.

2. Dodaj znak mniejszości oraz minus (<-)

Po nazwie zmiennej powinny zostać dodane dwa symbole: znak mniejszości oraz minus (<-)¹. Możesz przyjąć, że symbol <- ma następujące znaczenie: „została przypisana wartość”. W wyniku jego użycia informacja znajdująca się po prawej stronie symbolu <- zostaje przypisana do zmiennej leżącej po jego lewej stronie. Na przykład wiersz `> A <- B` może zostać odczytany jako „zmiennej o nazwie A została przypisana wartość zmiennej B”. Wynika stąd, że w naszym poprzednim przykładzie przypisaliśmy zmiennej o nazwie `hanzhongResources` zawartość pliku `zasobyHanzhong.csv`.

3. Dodaj źródło danych

W naszym przykładzie użyte zostało źródło danych `zasobyHanzhong.csv`. Źródło danych powinno być ostatnim składnikiem wprowadzonym w wierszu konsoli podczas tworzenia zmiennej w środowisku R. Źródła danych są zazwyczaj zbiorami danych, które są wczytywane do środowiska R, wartościami numerycznymi lub wcześniej utworzonymi zmiennymi.

4. Sprawdź zawartość zmiennej

Gdy po zatwierdzeniu wiersza z kodem środowisko R nie wyświetla żadnego widocznego wyniku, tak jak ma to miejsce w przypadku tworzenia nowej zmiennej, rozsądną czynnością jest zweryfikowanie wyniku naszych działań. Aby wyświetlić zawartość zmiennej, wprowadź w konsoli R jej nazwę, a następnie naciśnij klawisz *Enter*. W naszym przypadku zatwierdzenie ciągu znaków `hanzhongResources` spowoduje wyświetlenie w konsoli zasobów armii królestwa Shu stacjonującej w Chinach w mieście Hanzhong.

¹ Zamiast <- można też używać zwykłego znaku równości — *przyp. tłum.*

Sprawdzian

1. Które z poniższych stwierdzeń nie określa zalety przechowywania wyniku funkcji `read.csv(file)` w zmiennej?
 - a) Nazwa zmiennej jest łatwiejsza do wprowadzenia.
 - b) Nazwa zmiennej jest łatwiejsza do zapamiętania.
 - c) Dane zapamiętane w zmiennej nie zostaną utracone nawet wówczas, gdy pierwotny plik CSV zostanie przeniesiony lub usunięty.
 - d) Zmienna wskazuje jawnie na swoje źródło danych.
2. Opisz słowami znaczenie poniższego wiersza konsoli środowiska R:


```
> myVariable <- myData
```

 - a) Zmiennej `myVariable` została przypisana wartość zmiennej `myData`.
 - b) Zmiennej `myData` została przypisana wartość zmiennej `myVariable`.
 - c) Wartość zmiennej `myVariable` jest mniejsza niż negacja zmiennej `myData`.
 - d) Wartość zmiennej `myVariable` jest większa niż zero oraz mniejsza niż negacja zmiennej `myData`.

Do ćwiczenia

Znasz już proces kryjący się za tworzeniem nowej zmiennej w środowisku R. Plik *żołnierze* ↪ *Wmiastach.csv* zawiera informacje o całkowitej liczbie żołnierzy rozmieszczonych w każdym z większych miast na terytorium królestw Shu i Wei. Skopiuj ten plik do swojego katalogu roboczego. Następnie wykonaj czteroetapową procedurę umożliwiającą stworzenie i sprawdzenie wartości nowej zmiennej o nazwie `soldiersByCity`. Ta zmienna powinna zawierać wszystkie dane znajdujące się w pliku *żołnierzeWmiastach.csv*.

Czas na działanie — dostęp do danych w zmiennych

Obie nasze zmienne `hanzhongResources` i `soldiersByCity` zawierają cały zbiór wartości (w przeciwieństwie do pojedynczej wartości). Wiemy już, że wprowadzenie nazwy zmiennej w konsoli R spowoduje wyświetlenie jej zawartości na ekranie. Jednakże w przypadku zbioru danych często chcemy mieć dostęp do określonych kolumn, wierszy i komórek, aby móc przeprowadzać obliczenia.

Rozpocznijmy od przetestowania dwóch metod pozwalających na uzyskanie dostępu do kolumn w naszej zmiennej `soldiersByCity`:

1. Najpierw uzyskamy dostęp do zawartej w zmiennej `soldiersByCity` kolumny *Żołnierze* poprzez użycie notacji *zmienna\$kolumna*:


```
> #wyodrębnij pojedynczą kolumnę ze zbioru danych poprzez użycie notacji
↪zmienna$kolumna
> #wyświetl zawartość kolumny Żołnierze ze zmiennej soldiersByCity
> soldiersByCity$Żołnierze
```

2. Środowisko R wyświetli zawartość kolumny Żołnierze. Na poniższym rysunku przedstawiono wynik operacji.

```
> soldiersByCity$Żołnierze
 [1] 100000 25000 15000 15000 10000 10000
 [7]  5000  5000  5000  5000  5000 200000
[13] 50000 25000 20000 15000 15000 15000
[19] 15000 15000 15000 10000
```

3. Tym razem użyjmy funkcji `attach(zmienna)` w celu uproszczenia naszej operacji.

```
> #wyodrębnij pojedynczą kolumnę ze zbioru danych poprzez użycie funkcji
↳attach(zmienna) i uproszczonej notacji
> #przyłącz zmienną soldiersByCity
> attach(soldiersByCity)
> #wyświetl zawartość kolumny Żołnierze ze zmiennej soldiersByCity
> Żołnierze
```

4. Środowisko R wyświetli zawartość kolumny Żołnierze.

```
> Żołnierze
 [1] 100000 25000 15000 15000 10000 10000
 [7]  5000  5000  5000  5000  5000 200000
[13] 50000 25000 20000 15000 15000 15000
[19] 15000 15000 15000 10000
```

Obecnie uzyskamy dostęp do pojedynczego wiersza ze zmiennej `soldiersByCity`:

5. Użyj notacji macierzowej `zmienna[wiersz, kolumna]`, aby wyświetlić zawartość dziesiątego wiersza ze zmiennej `soldiersByCity`:

```
> #wyodrębnij pojedynczy wiersz ze zbioru danych poprzez użycie notacji
↳macierzowej zmienna[wiersz, kolumna]
> #wyświetl zawartość dziesiątego wiersza ze zmiennej soldiersByCity
> soldiersByCity[10,]
```

6. Środowisko R wyświetli zawartość dziesiątego wiersza ze zbioru danych `soldiersByCity`:

```
> soldiersByCity[10,]
      Królestwo Miasto Żołnierze
10          Shu Yunnan          5000
```

7. Notacji macierzowej możemy użyć w podobny sposób, aby uzyskać dostęp do pojedynczego elementu w naszym zbiorze danych. Użyjmy notacji macierzowej, aby wyświetlić zawartość elementu `[5, 3]` ze zmiennej `soldiersByCity`:

```
> #wyodrębnij pojedynczy element ze zbioru danych poprzez użycie notacji
macierzowej zmienna[wiersz, kolumna]
> #wyświetl zawartość elementu [5, 3] ze zmiennej soldiersByCity
> soldiersByCity[5, 3]
```

8. Środowisko R wyświetli zawartość elementu `[5, 3]`, jak pokazano na poniższym rysunku:

```
> soldiersByCity[5, 3]
[1] 10000
```

Co się zdarzyło?

Właśnie sprawdziłeś w praktyce uzyskiwanie dostępu do różnego rodzaju zestawów danych zawartych w zmiennej, to znaczy kolumn, wierszy i pojedynczych elementów. Przyjrzyjmy się bliżej sposobom, które pozwoliły nam uzyskać dostęp do danych w zmiennej.

Notacja *zmienna*\$kolumna

Pojedyncze kolumny znajdujące się w zbiorze danych mogą stać się dostępne dzięki użyciu notacji *zmienna*\$kolumna. Potraktuj znak dolara (\$) jako odpowiednik słowa „wybrać”. W ten sposób możemy przedstawić notację za pomocą słów. Na przykład wiersz `> A$B` może zostać odczytany jako „ze zmiennej A wybierz kolumnę B”. Podczas naszych działań wybraliśmy kolumnę Żołnierze ze zmiennej `soldiersByCity` poprzez wprowadzenie następującego kodu w konsoli R:

```
> soldiersByCity$Żołnierze
```

Funkcja `attach(zmienna)`

Funkcja `attach(zmienna)` pozwala na uproszczenie skomplikowanych notacji w niektórych, lecz nie we wszystkich przypadkach. Gdy zmienna zostaje **przyłączona** w konsoli środowiska R, jej kolumny mogą być dostępne po użyciu ich nazw bez potrzeby identyfikowania samej zmiennej. Na przykład po przyłączeniu zmiennej `soldiersByCity` mogliśmy wyświetlić wartość kolumny `Żołnierze` dzięki zwykłemu wprowadzeniu wiersza `> Żołnierze` w konsoli.

Ograniczeniem funkcji `attach(zmienna)` jest to, że w danej chwili do konsoli R może zostać przyłączona tylko pojedyncza zmienna. Jeśli na przykład w tym samym czasie przyłączylibyśmy obie nasze zmienne `hanzhongResources` i `soldiersByCity`, pojawiłby się problem związany z kolumną `Żołnierze`. Obie zmienne zawierają tę kolumnę, ale środowisko R mogłoby się odwoływać jedynie do tej zmiennej, która została przyłączona później. Dostęp do drugiej z nich byłby możliwy za pomocą notacji *zmienna*\$kolumna. W rzeczywistości środowisko R ostrzeże Cię, jeśli przyłączysz dwie zmienne zawierające kolumny o tej samej nazwie. Poniższy błąd pojawi się, gdy najpierw przyłączona zostanie zmienna `soldiersByCity`, a następnie `hanzhongResources`:

```
The following object(s) are masked from 'soldiersByCity':
  Żołnierze
```

Z drugiej strony, przyłączanie zmiennej może być przydatne i efektywne, jeśli pracujesz z pojedynczym, lecz dużym zbiorem danych. Jeżeli przetwarzasz dane pochodzące z jednej zmiennej, nie spowodujesz powstania powyższego błędu. Ponadto zawsze możesz mieć przyłączoną

jedną zmienną, nawet wówczas, gdy pracujesz ze zbiorami danych mającymi takie same nazwy kolumn. Oczywiście jeśli Twoje zmienne nie mają wspólnych kolumn, podłączanie ich wszystkich jest możliwe. W każdym razie możesz zawsze odwołać się do kolumn poprzez użycie notacji `zmienna$kolumna`. Zostanie to wykorzystane w dalszej części tej książki.

Zwróć uwagę na to, że gdy chcesz **odłączyć** zmienną, musisz użyć funkcji `detach(zmienna)`. Wykonanie polecenia spowoduje sprowadzenie zmiennej do jej wcześniejszego stanu, tak jakby nie została nigdy przyłączona.

Notacja `zmienna[wiersz, kolumna]`

Podczas odwoływania się do danych z wiersza lub pojedynczych elementów należy używać notacji `zmienna[wiersz, kolumna]`. W przypadku wierszy, co sprawdziliśmy podczas uzyskiwania dostępu do dziesiątego wiersza w zmiennej `soldiersByCity` poprzez wykonanie polecenia `> soldiersByCity[10,]`, pominięty zostaje fragment zapisu odnoszący się do kolumny. Taka instrukcja nakazuje środowisku R zwrócić dane ze wszystkich kolumn w wierszu.

Aby uzyskać pojedynczy element, należy podać odpowiednie wartości wiersza i kolumny. Gdy poprzez wykonanie polecenia `> soldiersByCity[5, 2]` odwoływaliśmy się do elementu `[5, 2]` ze zmiennej `soldiersByCity`, wartość 5 reprezentowała wiersz elementu, a 2 definiowała kolumnę. Taka operacja jest podobna do wyboru pojedynczego punktu z wykresu przy użyciu współrzędnych x-y, z tą różnicą, że w naszym przypadku wykres zastępujemy macierzą z danymi.

Na marginesie: być może zauważyłeś, że notacja `zmienna[wiersz, kolumna]` może również służyć do odwoływania się do kolumn. Osiąga się to dzięki pominięciu fragmentu dotyczącego wartości wiersza. Na przykład aby uzyskać dostęp do kolumny `Miasto` w zmiennej `soldiersByCity`, moglibyśmy użyć kodu `soldiersByCity[, 2]`, który nakazałby środowisku R zwrócić dane ze wszystkich wierszy w kolumnie `Miasto`.

Sprawdzian

- Opisz słowami znaczenie poniższego wiersza konsoli R:


```
> myVariable$myColumn
```

 - Pomnóż wartość zmiennej `myVariable` przez wartość zmiennej `myColumn`.
 - Podziel wartość zmiennej `myVariable` przez wartość zmiennej `myColumn`.
 - Ze zmiennej `myColumn` wybierz kolumnę `myVariable`.
 - Ze zmiennej `myVariable` wybierz kolumnę `myColumn`.
- W jakich okolicznościach przyłączenie w konsoli R zmiennych ze zbiorami danych nie jest najlepszym rozwiązaniem?
 - Przetwarzasz pojedynczy zbiór danych.
 - Przetwarzasz wiele zbiorów danych zawierających identyczne nazwy kolumn.

- c) Przetwarzasz wiele zbiorów danych zawierających identyczne nazwy kolumn, ale chcesz przyłączyć tylko jeden z nich.
 - d) Przetwarzasz wiele zbiorów danych, które nie zawierają identycznych nazw kolumn.
3. Notacja *zmienna*[*wiersz*, *kolumna*] może zostać użyta, by uzyskać dostęp do danych z następujących lokalizacji:
- a) Wierszy.
 - b) Kolumn.
 - c) Elementów.
 - d) Wszystkie powyższe opcje są poprawne.

Czas na działanie — przetwarzanie danych ze zmiennych

Umiejętność uzyskania dostępu do informacji przechowywanej w zmiennej jest wstępem do wykonania czynności przetwarzania danych. Zmienne i zawarte w nich dane mogą być wykorzystywane w ten sam sposób, w jaki w rozdziale 2. użyliśmy liczb do przeprowadzenia obliczeń. Mogą one zostać użyte w formułach matematycznych, a także wystąpić jako argumenty funkcji.

1. Użyj zmiennej `hanzhongResources`, aby obliczyć ilość zasobów, które posiadałaby armia Shu, jeśli powódź zniszczyłaby 75% każdego z zasobów:

```
> #jeśli powódź zniszczyła 75% zasobów Shu w mieście Hanzhong, ile ich
↳ pozostanie?
> #pomnóż zmienną hanzhongResources przez wartość 0,25, by wyznaczyć
↳ pozostałe 25% pierwotnych zasobów
> hanzhongResources * 0.25
```

2. Środowisko R wyświetli następujący wynik obliczeń:

```
> hanzhongResources * 0.25
  Złoto Zapasy Żołnierze StanWyposażenia
1 250000 250000      25000          0.25
```

3. Obecnie założymy, że hipotetyczna powódź miała wpływ jedynie na zapasy w mieście Hanzhong, natomiast inne zasoby pozostały nienaruszone. Musisz wykonać obliczenie jedynie dla kolumny Zapasy w zmiennej `hanzhongResources`:

```
> #jeśli powódź zniszczyła 75% zapasów w mieście Hanzhong, ile ich
↳ pozostanie?
> #pomnóż kolumnę Zapasy przez wartość 0,25, by wyznaczyć pozostałe 25%
↳ pierwotnych zasobów
> hanzhongResources$Zapasy * 0.25
```

4. Środowisko R wyświetli wynik obliczeń. Zwróć uwagę na to, że obliczenia mogą zostać wykonane w taki sam sposób dla wierszy, kolumn i elementów.

```
> hanzhongResources$Zapasy * 0.25
[1] 250000
```

5. Wartość zmiennej może także zostać użyta jako argument funkcji. Załóżmy mniej katastroficzny scenariusz i użyjmy zmiennej `soldiersByCity` w celu obliczenia średniej (przeciętnej) liczby żołnierzy stacjonujących w jednym mieście królestwa Shu:

```
> #użyj funkcji mean(dane), aby obliczyć średnią liczbę żołnierzy
↳stacjonujących w mieście królestwa Shu
> #w mieście królestwa Shu stacjonuje przeciętnie tylu żołnierzy:
> mean(soldiersByCity$Żołnierze)
```

6. Środowisko R wyświetli wynik obliczeń. Zwróć uwagę na to, że obliczenia mogą być wykonane w taki sam sposób dla danych w wierszach, kolumnach, poszczególnych elementów, a także dla całych zbiorów danych.

```
> mean(soldiersByCity$Żołnierze)
[1] 27045.45
```

7. Oprócz tego wyniki obliczeń mogą zostać zapamiętane w nowych zmiennych, aby można ich było użyć w późniejszym czasie. Tym razem zapiszemy wynik obliczeń z punktu 5. w nowej zmiennej o nazwie `meanSoldiersByCity`:

```
> #zapisz średnią liczbę żołnierzy przypadających na jedno miasto w zmiennej
↳o nazwie meanSoldiersByCity
> meanSoldiersByCity <- mean(soldiersByCity$Żołnierze)
```

8. Środowisko R nie wyświetli żadnego wyniku. Sprawdź zawartość zmiennej `meanSoldiersByCity` poprzez wprowadzenie w konsoli R następującego kodu:

```
> #wyświetl zawartość zmiennej meanSoldiersByCity
> meanSoldiersByCity
```

9. Środowisko R wyświetli zawartość zmiennej `meanSoldiersByCity`:

```
> meanSoldiersByCity
[1] 27045.45
```

Co się zdarzyło?

Zaledwie w kilku wierszach kodu przećwiczyłeś różne sposoby manipulowania danymi, których będziesz regularnie używał w środowisku R. Przeanalizujmy każdy z nich oddzielnie.

Wykonywanie obliczenia przy użyciu całego zbioru danych

Gdy użyłeś zmiennej `hanzhongResources` w celu określenia wpływu powodzi na każdy z zasobów, odkryłeś, że ta operacja wywołuje ten sam efekt dla wszystkich danych, które zawiera zmienna.

Dla celów prezentacyjnych załóżmy istnienie następującej tabeli z elementami o wartościach 1, 2, 3, 4 znajdującymi się odpowiednio w kolumnach a, b, c, i d:

| a | b | c | d |
|---|---|---|---|
| 1 | 2 | 3 | 4 |

Załóżmy, że ta tabela została zapamiętana w zmiennej środowiska R o nazwie `lettersAndNumbers`. Do tej zmiennej dodaliśmy wartość równą jeden, wykorzystując w tym celu następującą instrukcję:

```
> lettersAndNumbers + 1
```

Po tej operacji każdy z elementów tabeli zwiększył swoją wartość o jeden:

| a | b | c | d |
|---|---|---|---|
| 2 | 3 | 4 | 5 |

Jak widać, operacja przeprowadzona na zbiorze danych jest w rzeczywistości wykonywana na każdym z jego elementów. Jednakże środowisko R nie zawsze będzie potrafiło przeprowadzić udane obliczenia dla każdego elementu zbioru danych.

Na przykład jeśli chcielibyśmy przeprowadzić operację obliczenia numerycznego dla kolumn `Królestwo` i `Miasto` zawierających się w zmiennej `soldiersByCity`, środowisko R zwróciłoby ostrzeżenie, wyświetlając wartości `NA`². Wynika to stąd, że nasze kolumny `Królestwo` i `Miasto` zawierają tekst, co powoduje, że przetwarzanie ich w sposób numeryczny nie ma sensu. Aby pojawiło się takie ostrzeżenie, wprowadź w konsoli R poniższy kod:

```
> #co się stanie, gdy spróbujemy wykonać obliczenie numeryczne na danych
↳nienumerycznych?
> #otrzymamy ostrzeżenie, ponieważ przetwarzanie tekstu w sposób numeryczny nie
↳ma sensu
> soldiersByCity * 5
```

Uzyskamy następujący wynik (rysunek na następnej stronie).

Widzimy, że kolumna `Żołnierze` zawiera wartości numeryczne, dlatego każdy jej element został prawidłowo pomnożony przez pięć. Jednakże tekst w kolumnach `Królestwo` i `Miasto` nie mógł zostać pomnożony, dlatego pojawił się komunikat z ostrzeżeniem. Aby uniknąć otrzymywania wartości niemających zastosowania i dezorganizujących działanie konsoli R, ważne jest, żebyś był świadomy typu swoich danych i stosował dla nich odpowiednie rodzaje obliczeń.

² NA — ang. *not available* — niemający zastosowania, nie dotyczy — *przyp. tłum.*

```

> soldiersByCity * 5
  Królestwo Miasto Żołnierze
1          NA     NA    500000
2          NA     NA    125000
3          NA     NA     75000
4          NA     NA     75000
5          NA     NA     50000
6          NA     NA     50000
7          NA     NA     25000
8          NA     NA     25000
9          NA     NA     25000
10         NA     NA     25000
11         NA     NA     25000
12         NA     NA    1000000
13         NA     NA     250000
14         NA     NA    125000
15         NA     NA    100000
16         NA     NA     75000
17         NA     NA     75000
18         NA     NA     75000
19         NA     NA     75000
20         NA     NA     75000
21         NA     NA     75000
22         NA     NA     50000

Warning messages:
1: In Ops.factor(left, right) : * not meaningful for factors
2: In Ops.factor(left, right) : * not meaningful for factors

```

Wykonywanie obliczenia dla wiersza, kolumny lub elementu

Przetwarzanie danych w wierszu, kolumnie czy elemencie przebiega identycznie jak przetwarzanie całego zbioru danych zawartego w zmiennej. Różnicą nie jest samo obliczanie, lecz raczej to, co zostanie wybrane do jego przeprowadzenia. W zależności od tego, czy chcesz przetwarzać dane w wierszu, kolumnie lub pojedynczym elemencie, musisz uzyskać do nich dostęp w odpowiedni sposób. Zapoznaj się z podrozdziałem zatytułowanym „Czas na działanie — dostęp do danych w zmiennych”, aby dowiedzieć się więcej o tych metodach.

Użycie zmiennych jako argumentów funkcji

Zarówno cały zbiór danych zawarty w zmiennej, jak i jedynie jego określony podzbiór (wiersz, kolumna lub element) mogą być używane jako argumenty funkcji. W naszych poprzednich działaniach wykorzystywaliśmy funkcję `mean(dane)` w celu obliczenia średniej liczby żołnierzy przypadających na każde z miast królestwa Shu występujących w zmiennej `soldiersByCity`. Tę operację można równie łatwo wykonać dla całego zbioru danych, pojedynczego wiersza lub elementu. Najlepszy sposób użycia zmiennej jako argumentu zależy od tego, jaki jest cel przetwarzania danych, oraz od zastosowania określonej funkcji.

Zapisywanie wyniku obliczenia w nowej zmiennej

Nie zapominaj, że celem istnienia zmiennych jest przechowywanie i porządkowanie Twoich informacji. Dość często dla potrzeb kolejnych przetwarzań będziemy musieli zapisywać w nowej zmiennej wyniki obliczeń lub rezultaty zwracane przez funkcję. Zmienne i inne obiekty, które powstają podczas naszych działań, są przechowywane w **obszarze roboczym** środowiska R, który zostanie omówiony w kolejnym podrozdziale.

Sprawdzian

Tabela `myTable` składa się z dwóch wierszy, trzech kolumn i sześciu elementów o wartościach od 1 do 6. Użyj tej tabeli, aby odpowiedzieć na pytania z punktu 1. i 2.

| <code>myTable</code> | | |
|----------------------|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |

1. Rozważ następujący wiersz kodu:

```
> myTable * 10
```

Jeśli ten kod zostałby użyty dla tabeli `myTable`, jaki byłby wynik działania?

Wpisz odpowiednie wartości w puste pola tabeli `myTableAfterManipulation1`:

| <code>myTableAfterManipulation1</code> | | |
|--|--|--|
| | | |

2. Rozważ następujący wiersz kodu:

```
> myTable[1, 2] + 10
```

Jeśli ten kod zostałby użyty dla tabeli `myTable`, jaki byłby wynik działania?

Wpisz odpowiednie wartości w puste pola tabeli `myTableAfterManipulation2`:

| <code>myTableAfterManipulation2</code> | | |
|--|--|--|
| | | |

3. Zinterpretuj słownie następujący wiersz konsoli R:

```
> myVariable <- mean(myData$myColumn)
```

- a) Oblicza średnią dla zmiennej `myColumn`, a następnie przypisuje wynik do zmiennej `myVariable`.

- b) Oblicza średnią dla zmiennej `myData`, a następnie przypisuje wynik do zmiennej `myVariable`.
- c) Dla zmiennej `myData` wybiera składnik `myColumn`, oblicza jego średnią, a następnie przypisuje wynik do zmiennej `myVariable`.
- d) Przypisuje zmiennej `myVariable` wartość zmiennej `myData`, a następnie oblicza jej średnią.

Do przećwiczenia

Aby przetestować działanie różnych metod służących do manipulowania danymi, użyj danych o zasobach oraz swojej wiedzy o środowisku R i spróbuj rozwiązać następujące zadania:

1. Przypuśćmy, że obawiasz się powodzi, która mogłaby doprowadzić do zniszczenia Twoich zasobów. Oblicz ilość zasobów, które pozostałyby, gdyby powódź zniszczyła połowę każdego zasobu zapisanego w zmiennej `hanzhongResources`. Zapamiętaj wynik w pojedynczej zmiennej `hanzhongResourcesAfterFlood`.
2. Aby zanotować ostatnie przemieszczenie 5000 żołnierzy z miasta Guanghan do Baxi, odejmij 5000 od elementu reprezentującego liczbę żołnierzy w Guanghan, a następnie dodaj 5000 do elementu reprezentującego liczbę żołnierzy w Baxi. Informacje o liczbie żołnierzy w miastach są zapisane w zmiennej `soldiersByCity`. Zapamiętaj wynik każdego z obliczeń w nowej zmiennej. Zmienne powinny mieć nazwy `guanghanSoldiersAfterRelocation` i `baxiSoldiersAfterRelocation`.
3. Użyj funkcji `min(dane)` i `max(dane)` oraz zmiennej `soldiersByCity`, aby obliczyć najmniejszą i największą liczbę żołnierzy w mieście. Zapisz wyniki odpowiednio w zmiennych `minSoldiersByCity` i `maxSoldiersByCity`.
4. Użyj funkcji `sum(dane)` oraz zmiennej `soldiersByCity`, aby obliczyć całkowitą liczbę żołnierzy armii Shu i Wei. Następnie zapamiętaj wynik w zmiennej `totalSoldiers`.

Jeśli w trakcie realizacji powyższych zadań otrzymasz komunikat ostrzeżenia lub błędu, postaraj się dokładniej zdefiniować, jakie dane powinny zostać użyte w obliczeniu lub argumencie funkcji. Aby dowiedzieć się więcej o reagowaniu w przypadku takich sytuacji, wróć do wcześniejszego podrozdziału „Wykonywanie obliczenia przy użyciu całego zbioru danych”.

Czas na działanie — zarządzanie środowiskiem R

Obszar roboczy środowiska R przechowuje wszystkie obiekty utworzone przez użytkownika (w naszym przypadku są to zmienne), które powstały w trakcie sesji. Jego zawartość może zostać zapamiętana i odtworzona, tak by można jej było użyć w przyszłości.

1. Użyj funkcji `ls()`, aby wyświetlić zawartość obszaru roboczego środowiska R:


```
> #wyświetl bieżącą zawartość obszaru roboczego środowiska R
> ls()
```
2. Środowisko R wyświetli listę obiektów w Twoim obszarze roboczym:

```

> ls()
[1] "baxiSoldiersAfterRelocation"
[2] "guanghanSoldiersAfterRelocation"
[3] "hanzhongResources"
[4] "hanzhongResourcesAfterFlood"
[5] "maxSoldiersByCity"
[6] "meanSoldiersByCity"
[7] "minSoldiersByCity"
[8] "soldiersByCity"
[9] "totalSoldiers"

```

3. Użyj funkcji `save.image(file)`, aby zapamiętać swój obszar roboczy w katalogu roboczym. Argument `file` powinien mieć postać sensownej nazwy pliku z rozszerzeniem `.RData`:

```

> #zapamiętaj obszar roboczy środowiska R w katalogu roboczym za pomocą
↳ funkcji save.image(file)
> save.image("rDlaPoczątkujących_R_04.RData")

```

4. Środowisko R zapamięta Twój obszar roboczy w pliku. Otwórz katalog roboczy za pomocą odpowiedniego narzędzia systemu operacyjnego, aby upewnić się, że plik ten został rzeczywiście utworzony.

5. Użyj polecenia `q()`, aby wyjść ze środowiska R. Zignoruj lub odrzuć komunikaty, które się pojawiają.

```

> #zakończ sesję R
> q()

```

6. Środowisko R zostanie zamknięte.

7. Ponownie uruchom środowisko R poprzez kliknięcie jego ikony. Następnie użyj polecenia `ls()`, aby sprawdzić, czy bieżący obszar roboczy jest pusty:

```

> #wyświetl bieżącą zawartość obszaru roboczego środowiska R
> ls()

```

Uzyskasz następujący wynik:

```

> ls()
character(0)

```

8. Zdefiniuj katalog roboczy poprzez wprowadzenie polecenia `setwd(dir)`:

```

> #zdefiniuj katalog roboczy
> setwd("C:/Users/RUser/Strateg")

```

9. Użyj funkcji `load(file)`, aby wczytać zapisany plik z obszarem roboczym. Argument `file` powinien być taki sam jak argument użyty w punkcie 3.:

```

> #wczytaj poprzednio zapamiętany obszar roboczy środowiska R za pomocą
↳ funkcji load(file)
> load("rDlaPoczątkujących_R_04.RData")

```

10. Użyj polecenia `ls()`, aby sprawdzić, czy zapamiętana zawartość jest obecnie dostępna w obszarze roboczym środowiska R:

```
> #wyświetl bieżącą zawartość obszaru roboczego środowiska R
> ls()
```

```
> ls()
[1] "baxiSoldiersAfterRelocation"
[2] "guanghanSoldiersAfterRelocation"
[3] "hanzhongResources"
[4] "hanzhongResourcesAfterFlood"
[5] "maxSoldiersByCity"
[6] "meanSoldiersByCity"
[7] "minSoldiersByCity"
[8] "soldiersByCity"
[9] "totalSoldiers"
```

Co się zdarzyło?

Właśnie sprawdziłeś w praktyce działanie podstawowych funkcji zarządzających obszarem roboczym, których będziesz potrzebować, by przekazywać swoje dane między różnymi sesjami środowiska R. Są wśród nich funkcje służące do wyświetlania, zapamiętywania i czytania zawartości Twojego obszaru roboczego w środowisku R.

Wyświetlanie zawartości obszaru roboczego R

Obszar roboczy w środowisku R zawiera wszystkie obiekty, które stworzyłeś w trakcie swojej sesji. Do tej chwili nasze obiekty przyjmowały formę zmiennych, które „odczytywały” dane z plików CSV lub przechowywały wyniki obliczeń. Funkcja `ls()` może zostać wywołana zawsze, aby wyświetlić zawartość obszaru roboczego R.

Zapamiętywanie zawartości obszaru roboczego R

Aby zapamiętać obszar roboczy środowiska R, użyj funkcji `save.image(file)`. Ponieważ działaliśmy w naszym katalogu roboczym, argument `file` musiał się składać jedynie z nazwy pliku i rozszerzenia `.RData`. Jeśli obszar roboczy został zapisany w innym miejscu, należy podać pełną ścieżkę. Pamiętaj, aby zawsze dodawać rozszerzenie `.Rdata` podczas zapamiętywania swojego obszaru roboczego, ponieważ jest ono niezbędne, by środowisko R mogło poprawnie rozpoznać plik podczas jego ładowania.

Odczytywanie zawartości obszaru roboczego R

Aby odczytać zawartość obszaru roboczego, użyj funkcji `load(file)`. Argument `file` jest w tym przypadku identyczny z argumentem używanym w funkcji `save.image(file)`. Jeśli więc plik, który chcesz załadować, znajduje się w Twoim katalogu roboczym, będziesz musiał podać je-

dynie jego nazwę oraz rozszerzenie *.RData*. Jeżeli jest on umieszczony gdzie indziej, należy wprowadzić pełną ścieżkę dostępu do niego.

Zauważ, że w zależności od posiadanej przez Ciebie wersji środowiska R zapamiętanie i wczytanie obszaru roboczego R może zostać wykonane automatycznie podczas uruchamiania lub zamykania aplikacji. Inną metodą jest kliknięcie odpowiedniej opcji w menu. Być może będziesz chciał sprawdzić dostępne opcje menu oraz ustawienia preferencji. Bez względu na to zalecamy, abyś podczas używania konsoli R zarządzał jej obszarem roboczym, ponieważ pozwoli to na pełną kontrolę wykonywanych działań.

Zamykanie środowiska R

Jak mogłeś zauważyć, polecenie `q()` może zostać wykorzystane do wyjścia ze środowiska R. W tym celu możesz oczywiście także użyć odpowiednich opcji menu, skrótów klawiszowych lub innych sposobów dostępnych w Twoim komputerze.

Jeśli nie zdecydujesz, że obszar roboczy ma zostać zapamiętany podczas zamykania środowiska R, cała jego zawartość zostanie utracona. Pamiętaj, aby zapisać obszar roboczy przed wyjściem ze środowiska R.

Rozróżnienie między konsolą R a obszarem roboczym

Gdy podczas wykonywania wcześniejszych czynności ponownie uruchomiłeś środowisko R i wczytałeś zapamiętany plik z obszarem roboczym, być może zauważyłeś, że zawartość ekranu samej konsoli nie została przywrócona. Ujawnia to ważną różnicę między konsolą środowiska R a jego obszarem roboczym. Zasadniczo obszar roboczy służy do przechowywania wszystkich obiektów użytkownika, konsola zaś zawiera historię operacji, które zostały wykonane na tych obiektach.

Wyobraź sobie czynność oglądania filmu w kinie jako analogię prezentującą związek między obszarem roboczym i konsolą. Widownia, ekran i fotele znajdują się w tym samym pomieszczeniu (obszarze roboczym). Wszystkie czynności, jakie wykonują te „obiekty” — kichają, śmieją się, jedzą, wyświetlają film, przyklejają gumę do żucia do foteli — są zapisywane w historii wrażeń odbioru filmu (konsoli).

Obszar roboczy zawiera obiekty (takie jak osoby w kinie), konsola zaś zapamiętuje interakcje, które wystąpiły między nimi (takie jak wysypanie komuś popcornu na głowę).

Zapamiętywanie stanu konsoli R

Ponieważ stan konsoli nie jest zapamiętywany podczas zapisywania obszaru roboczego, zapewne chcesz wiedzieć, w jaki sposób można zachować informacje z danej sesji w przypadku konsoli środowiska R. Mimo że w środowisku R nie ma dostępnej funkcji, która zapisywałaby zawartość konsoli, mamy do dyspozycji inne opcje.

Jedną z nich jest skopiowanie i wklejenie treści znajdujących się w konsoli do edytora tekstowego. Inną, zależną od posiadanej przez Ciebie wersji środowiska R, może być użycie menu w celu zapamiętania kopii konsoli jako pliku tekstowego. Są to zalecane techniki pozwalające na zapisanie stanu konsoli R. Możesz również zastanowić się nad metodami alternatywnymi.

W każdym razie zdecydowanie zalecane jest zapisywanie stanu konsoli R pod koniec każdej sesji. Posiadanie historii Twoich poprzednich działań może uchronić Cię przed wykonaniem tych samych operacji, a jednocześnie pomóc w przyszłej pracy. Pozwala także na zorganizowanie i zapamiętanie wszystkiego, co jest związane z bieżącym projektem, szczególnie wtedy, gdy zarządzasz dużą ilością danych oraz wieloma obiektami.

Sprawdzian

1. Jakiego rozszerzenia pliku powinieneś użyć podczas zapisywania obszaru roboczego R?
 - a) `.txt`
 - b) `.R`
 - c) `.RData`
 - d) Nie jest potrzebne użycie rozszerzenia.
2. Które z poniższych zdań najlepiej opisuje związek między konsolą a obszarem roboczym środowiska R?
 - a) Obszar roboczy oraz konsola środowiska R mogą zostać zapamiętane za pomocą funkcji `save.image(file)`.
 - b) Zawartość obszaru roboczego i konsoli środowiska R może zostać wyświetlona za pomocą funkcji `ls()`.
 - c) Konsola środowiska R przechowuje obiekty, a obszar roboczy historię związanych z nimi działań.
 - d) Obszar roboczy środowiska R przechowuje obiekty, a konsola historię związanych z nimi działań.
3. Które z poniższych stwierdzeń nie opisują opcji pozwalającej na zapisanie stanu konsoli R?
 - a) Użycie wbudowanej funkcji środowiska R.
 - b) Skopiowanie zawartości konsoli do edytora tekstowego.
 - c) Użycie opcji menu środowiska R do zapisania zawartości konsoli w pliku tekstowym.
 - d) Zrzut ekranu konsoli środowiska R.

Do przećwiczenia

Twoje końcowe zadanie w tym rozdziale będzie polegać na zebraniu i uporządkowaniu pozostałych danych o zasobach otrzymanych od Zhuge Lianga. Praca ta będzie wymagała wczytania pliku CSV do środowiska R, stworzenia nowych zmiennych, uzyskania dostępu do danych

i ich przetworzenia, a wreszcie zapisania obszaru roboczego środowiska R oraz zawartości konsoli. Poprzez wykonanie poniższych działań zaprezentuj wysoki poziom posiadanej wiedzy na temat wspomnianych zagadnień, przygotowując na potrzeby analizy dane archiwalne o bitwach.

1. Plik *historiaBitew.csv* zawiera dane dotyczące 120 bitew przeprowadzonych między siłami Shu i Wei. Wczytaj te dane do zmiennej o nazwie `battleHistory`.
2. Użyj danych zaimportowanych w punkcie 1., by odpowiedzieć na następujące pytanie: jaka liczba żołnierzy królestw Shu i Wei była przeciętnie zaangażowana w walki? Zapamiętaj wyniki w oddzielnych zmiennych o nazwach odpowiednio `meanSoldiersShu` i `meanSoldiersWei`.
3. Zapisz zawartość swojego obszaru roboczego R w nowym pliku o nazwie *rDlaPoczątkujących_R_04_ćwiczenie.RData*.
4. Zapisz zawartość swojej konsoli R w nowym pliku o nazwie *rDlaPoczątkujących_R_04_ćwiczenie.txt*.

W razie potrzeby korzystaj z poprzednich podrozdziałów w celu uzyskania pomocy podczas gromadzenia i porządkowania informacji.

Podsumowanie

W tym rozdziale dokładnie przeanalizowałeś metody służące do gromadzenia i porządkowania informacji w środowisku R. Mają one bardzo duże znaczenie dla odniesienia sukcesu w dziedzinach strategii i analityki. Umiejętność skutecznego zarządzania danymi jest kluczem do uproszczenia sposobu działania oraz sprawienia, że wyniki Twojej pracy będą zrozumiałe dla innych. Podczas gromadzenia i porządkowania danych zdobyłeś umiejętności niezbędne do:

- Importowania danych zewnętrznych do środowiska R.
- Używania zmiennych w celu porządkowania i przetwarzania swoich danych.
- Zarządzania obszarem roboczym R.

Dysponując przygotowanymi danymi, możesz obecnie rozpocząć szacowanie wyzwań militarnych dla sił zbrojnych królestwa Shu. W rozdziale 5. ocenimy opcje walki dostępne dla armii Shu. Od Ciebie będzie zależało przyjęcie rozważnego toku działań.

Skorowidz

#, 47
\$, 55
+, 117
<-, 52
>, 47

A

abind, 260
abline(), 206, 211, 249
 lty, 211
 lwd, 218
 reg, 211
AIC, 140
 test, 140, 141
AIC(), 140, 141
Akaike Information Criterion, *Patrz* AIC
analiza
 regresji, 150, *Patrz* regresja
 wariancji, *Patrz* ANOVA
analysis of variance, *Patrz* ANOVA
angle, 193, 199
ANOVA, 103, 104
 tabela, 105
anova(), 104, 141
argument funkcji, 37
 przetwarzanie danych ze zmiennych, 58
 użycie zmiennej, 60
 zmienna liczba, 73
as.numeric(), 79, 80
at, 225
attach(), 54, 55
axis(), 223, 225, 240
 at, 225
 labels, 225
 las, 225

B

barplot(), 154, 155, 164
 dodatkowe argumenty, 188–200
beside, 198
blank, 211
bmp(), 183
boxplot(), 173, 176, 224
 range, 224
break, 227
breaks, 228

C

c(), 156
cex, 200, 201, 208
chooseCRANmirror(), 257, 259
coef(), 143, 146
col, 162, 163, 164, 218
colors(), 164
Comprehensive R Archive Network, *Patrz* CRAN
cor(), 82, 83, 85, 86
CRAN, 259
 serwer lustrzany, 259, 262
csv, 50, 51
czas bieżący, 36

D

dane
 importowanie, 49
 kategoryczne, 79
 przekodowanie na postać numeryczną, 79
 nominalne, 79
 szereg, 172
 zewnętrzne importowanie, 49

- dashed, 211
 - data bieżąca, 36
 - dataset, 169
 - date(), 37
 - density, 193, 199
 - detach(), 56
 - dev.off(), 183
 - digits, 233
 - dodanie osi do grafiki, 236
 - dopasowywanie grafiki, 159
 - argumenty, 162
 - col, *Patrz* col
 - main, *Patrz* main
 - xlab, *Patrz* xlab
 - xlim, *Patrz* xlim
 - ylab, *Patrz* ylab
 - ylim, *Patrz* ylim
 - dostęp do
 - kolumny w zmiennej, 53
 - pojedynczego elementu w zmiennej, 54
 - wiersza w zmiennej, 54
 - dotdash, 211
 - dotted, 211
- E**
- efekt
 - interakcji, 97
 - osłabiania, 99
 - eksportowanie grafiki, 182
 - element obliczenia dla, 60
- F**
- file, 50, 63, 64, 132
 - fill, 165
 - formuła, 46
 - przetwarzanie danych ze zmiennych, 57
 - freq, 228, 229
 - function(), 115, 116, 119
 - funkcja, 37
 - abline(), *Patrz* abline()
 - AIC(), 140, 141
 - anova(), 104, 141
 - as.numeric(), 79, 80
 - attach(), 54, 55
 - axis(), *Patrz* axis()
 - barplot(), 154, 155, 164
 - bmp(), 183
 - boxplot(), *Patrz* boxplot()
 - c(), 156
 - chooseCRANmirror(), 257, 259
 - coef(), 143, 146
 - colors(), 164
 - cor(), 82, 83, 85, 86
 - definiowanie własnej, *Patrz* function()
 - detach(), 56
 - dev.off(), 183
 - getwd(), 37, 38, 148
 - glm(), 136, 141
 - graficzna, 161
 - graphicsFunction(), 162
 - help(), *Patrz* help()
 - hist(), 177, 178
 - install.packages(), 258, 260
 - jpeg(), 183
 - legend(), 165, *Patrz* legend()
 - length(), 163
 - library(), 258, 260
 - lines(), *Patrz* lines()
 - lm(), 88, 89, 94
 - load(), 64, 69
 - ls(), 62, 64
 - magic(), 259
 - matrix(), 198
 - max(), 62
 - mean(), 58, 75
 - min(), 62
 - o zmiennej liczbie argumentów, 73
 - par(), 248
 - paste(), 233
 - pdf(), 183
 - pie(), *Patrz* pie()
 - plot(), *Patrz* plot()
 - png(), 183
 - points(), 204, 209, 240
 - pomocy, 254
 - rainbow(), 163
 - range(), 76, 77
 - read, 50
 - read.csv(), 50, 148
 - read.table(), 131, 132
 - round(), 233
 - save.image(), 63, 64, 150
 - sd(), 75, 77
 - setwd(), 37, 38, 148

subset(), 72, 74
 sum(), 62
 summary(), 76, 77, 91, 136
 tiff(), 183
 wieloargumentowa, 73
 własna użytkownika, *Patrz* function()
 zorientowana na zasoby, 119
 zagnieżdżona, 250

G

getwd(), 37, 38, 148
 gęstość, 229
 glm(), 136, 141
 GNU General Public License, 16
 Google, 265
 graficzne okno, 156
 grafika
 dopasowywanie, *Patrz* dopasowywanie grafiki
 eksportowanie, 182
 kolory
 automatyczne, 163
 zdefiniowane, 164
 linie poziome i pionowe, 249
 opis, 165
 tworzenie, 184, 235–241
 dodanie osi, 236
 par(), 248
 plot(), 235
 uzupełnianie danych, 237
 z wieloma elementami wizualnymi, 242–50
 wygląd, *Patrz* dopasowywanie grafiki
 graficzna funkcja, 161
 graphicsFunction(), 162

H

header, 132
 height, 155
 help(), 253, 254, 256
 topic, 255
 hierarchical linear regression, 101
 hierarchiczna regresja liniowa, 101
 hist(), 177, 178
 histogram, 177
 break, 227
 breaks, 228

dopasowywanie, 227–230
 freq, 228, 229
 pogrupowanie kolumn, 227, 228
 wyświetlanie gęstości zamiast liczebności,
 228, 229
 HLR, *Patrz* hierarchiczna regresja liniowa
 horiz, 190, 196

I

importowanie danych zewnętrznych, 49
 instalacja środowiska R, 27–36
 install.packages(), 258, 260
 interakcja, 97
 predyktor, 99
 zmienna, 97

J

jpeg(), 183

K

catalog roboczy, 37, 38
 zapamiętanie obszaru roboczego, 63
 kategoriyczne dane, *Patrz* dane kategoriyczne
 kolory
 automatyczne, 163
 zdefiniowane, 164
 kolumna obliczenia dla, 60
 komentarz, 45, 148
 konsola R, 36
 a obszar roboczy, 65
 importowanie danych zewnętrznych, 49
 komentarz, 45
 wiersz, 44
 rozszerzony, 117
 wyświetlanie informacji, 47
 zapamiętywanie stanu, 65
 korelacja, 82, 83, 85, 132
 interpretowanie, 84
 tabela, 85
 Kryterium Informacyjne Akaike, *Patrz* AIC
 kwartył
 dolny, 173
 górnny, 173

L

labels, 181, 225, 231
 las, 225
 legend, 165
 legend(), 165, 195, 200, 211, 220, 234, 241
 angle, 200
 cex, 200
 density, 200
 length(), 163
 library(), 258, 260
 liczebność, 229
 lines(), 215, 218
 col, 218
 type, 218
 lm(), 88, 89, 94
 load(), 64, 69
 load.image(), 149
 logistyka, 120
 longdash, 211
 ls(), 62, 64
 lty, 211
 lwd, 213, 218

M

magic, 260, 261
 magic(), 259
 magiczny kwadrat, 42
 main, 162
 maksimum, 62, 77
 matrix(), 198
 max(), 62
 mean(), 58, 75
 mediana, 173
 min(), 62
 minimum, 62, 77
 model regresji, 87

N

nadpisywanie zmiennych, 81
 najlepiej dopasowane proste, 211
 dodanie do wykresu punktowego, 211
 names, 189, 195
 nominalne dane, *Patrz* dane kategoryczne

O

obliczenia
 dla
 elementu, 60
 kolumny, 60
 wiersza, 60
 na zbiorze danych, 59
 obszar roboczy, 61, 62
 a konsola, 65
 odczytanie zawartości, 64
 wyświetlanie zawartości, 62, 64
 zapamiętanie, 64
 w katalogu roboczym, 63
 zarządzanie, 62
 od do, 172
 odchylenie standardowe, 75, 77
 odczytanie zawartości obszaru roboczego, 64
 odłączenie zmiennej, 56
 okno graficzne, 156
 operator
 nierównoważności, 73
 równoważności, 73
 opis, 165
 opisowa statystyka $\backslash t$, 75
 osłabiania efekt, 99
 oś wykresu
 dodanie przy tworzeniu grafiki, 236
 modyfikacja, 225
 przedefiniowanie skali, 210

P

Packages, 260
 pakiet, 257, 260
 użycie, 261
 wczytywanie, 260
 zależny, 260
 par(), 248
 paste(), 233
 pch, 201, 208
 pdf(), 183
 pie(), 179, 181
 labels, 181
 plik
 csv, 51
 danych, 51
 z wartościami oddzielnymi przecinkami,
 Patrz csv

plot(), 166, 168, 170, 208, 218, 235, 239

cex, 208

lwd, 218

pch, 208

type, 171

png(), 183

pobranie środowiska R, 27–31

podsumowanie wyników, 76

points(), 204, 209, 240

polecenie, 37

predyktor, 99

prefiks

#, 47

+, 117

>, 47

brak, 46

prognozowanie, 140

metodą regresji, 114

wyników, 87

prosta

regresja liniowa, 88

regresji, 211

 dodanie do wykresu punktowego, 211

przedefiniowanie skali na osi, 210

przetwarzanie danych ze zmiennych, 57

 jako argument funkcji, 58

 w formułach matematycznych, 57

przewidywanie, 142

przyłączenie zmiennej, 55

p-wartość, 91, 105

Q

q(), 63, 65

R

R, 16, 17, 25

 formuła, 46

 funkcja, 37

 instalacja środowiska, 27–36

 konsola, 36

 obszar roboczy, *Patrz* obszar roboczy

 pobranie środowiska, 27–31

 polecenie, 37

 rozszerzanie o nowe pakiety, 257

 strony internetowe, 263

 wyjście ze środowiska, 65

 wynik, 46

 zasoby, 253

 dostępne w internecie, 262–265

 zmienna, *Patrz* zmienna

R Project for Statistical Computing, *Patrz* R

rainbow(), 163

range, 221, 224

range(), 76, 77

RData, 63, 64, 65

read, 50

read.csv(), 50, 148

read.table(), 131, 132

reg, 211

regresja

 regresja, 87, 150

 liniowa

 hierarchiczna, 101

 prosta, 88, 89, 90, 92

 wielokrotna, 94, 95, 97

 model, 87

 interakcyjny, 97

 liniowy, 90

 interpretacja, 92

 porównanie modeli, 101

 prognozowanie, 114, 140

 projektowanie optymalnego modelu, 136

 przecięcie, 90

 przewidywanie, 142

 p-wartość, 91

 równanie, 90

 statystyczna istotność, 91

 współczynnik, 90

 determinacji, 91

 współliniowość, 99

 wyznaczanie wyników z modeli, 114

 zmienna

 niezależna, 87

 objaśniająca, 87

 objaśniana, 87

 zależna, 87

round(), 233

rozstęp ćwiartkowy, 224

rozszerzenie RData, *Patrz* RData

równoważność, 73

S

save.image(), 63, 64, 150

sd(), 75, 77

sep, 132

serwer CRAN, 259
 lustrzany, 259
 setwd(), 37, 38, 148
 sliceData, 181
 sliceText, 181
 solid, 211
 space, 189, 196
 standardowe odchylenie, 75, 77
 statystyczna istotność, 91
 statystyka
 opisowa, *Patrz* statystyka zbiorcza
 zbiorcza, 75, 77, 78, 132, 149
 strony internetowe na temat R, 263
 subset(), 72, 74
 sum(), 62
 suma, 62
 summary(), 76, 77, 91, 136
 szeregi danych, 172

Ś

średnia, 58, 75
 środowisko
 graficzne R, *Patrz* R
 obliczeniowe R, *Patrz* R
 R, *Patrz* R

T

tabela
 ANOVA, 105
 korelacji, 85
 test AIC, 141
 tiff(), 183
 twodash, 211
 tworzenie
 grafiki, *Patrz* grafika tworzenie
 dodanie objaśnień, 238
 z wieloma elementami wizualnymi, 242–50
 własnego wykresu, *Patrz* grafika tworzenie
 zmiennej, 52
 type, 171, 218

U

urządzenie graficzne, 183
 zamknięcie, 184

W

wariancja, 91
 wartość
 maksymalna, 62, 77, 173
 minimalna, 62, 77, 173
 NA, 86
 zmiennej, 51, 52
 przypisanie, 52
 wektor, 156
 width, 189, 196
 wieloargumentowa funkcja, *Patrz* funkcja
 wieloargumentowa
 wielokrotna regresja liniowa, 94, 95, 97
 wiersz, 44
 obliczenia dla, 60
 rozszerzony, 117
 współczynnik determinacji, 91
 współliniowość, 99
 wycentrowanie zmiennej, 99
 wyjście ze środowiska R, 65
 wykres, 156
 kołowy, 179
 dodanie opisu, 231, 232
 dopasowywanie, 231–34
 labels, 231
 opis zawartości procentowych
 składników, 231
 liniowy, 170
 dodanie nowych relacji, 215, 218
 objaśnienia, 217
 dopasowywanie, 220
 grubość linii, 213, 218
 kolor linii, 218
 lwd, 213, 218
 modyfikacja osi, 225
 punktowy, 166
 abline(), 206
 cex, 201
 dataset, 169
 dodanie najlepiej dopasowanych
 prostych, 206
 dodanie nowych relacji, 204
 dodanie opisów, 205
 dopasowywanie, 201–212
 pch, 201
 points(), 204, *Patrz* points()
 pojedynczy, 168

symbole odpowiadające punktom danych, 208
 wygląd symboli oznaczających punkty, 201, 209
 skrzynkowy, 173
 długość wąsów, 221, 224
 dopasowywanie, 221–226
 pojedynczy, 176
 range, 221
 rozstęp ćwiartkowy, 224
 stworzenie własnych osi, 222, 223
 wieloelementowy, 176
 słupkowy, 154
 angle, 193, 199
 beside, 192, 198
 density, 193, 199
 dopasowywanie, 188–200
 horiz, 190, 196
 names, 189, 195
 odstęp między słupkami, 196
 skumulowany, 192
 space, 189, 196
 sposób cieniowania słupków, 199
 tło słupków, 193
 width, 189, 196
 zmiana orientacji, 190
 tworzenie własnego, *Patrz* grafika tworzenie
 wynik, 46
 podsumowanie, 76
 prognozowanie, 87
 z modelu regresji, 114
 zapisywanie w nowej zmiennej, 61
 wyświetlanie zawartości obszaru roboczego, 64

X

xaxt, 225
 xlab, 162
 xlim, 162, 163, 210

Y

yaxt, 225
 ylab, 162
 ylim, 162, 163, 210, 220

Z

zakres, 75
 nienumeryczny, 79
 zbioru danych, 76, 77
 zamykanie środowiska R, 65
 zapamiętanie obszaru roboczego, 64
 zasoby, 253
 zawartość obszaru roboczego, 62
 zbiorcza statystyka, 75, 77, 78, 132, 149
 zbiór danych, 58
 obliczenia na, 59
 zmiana
 orientacji wykresu słupkowego, 190
 tła słupków na wykresie, 193
 zmienna, 51, 52
 dostęp do
 kolumny, 53
 pojedynczego elementu, 54
 wiersza, 54
 interakcyjna, 97
 korelowanie, *Patrz* korelacja
 liczba argumentów, 73
 nadpisywanie, 81
 odczytanie danych, 51
 odłączenie, 56
 przetwarzanie danych, *Patrz* przetwarzanie
 danych ze zmiennych
 przyłączenie, 55
 przypisanie wartości, 52
 tworzenie, 52
 użycie jako argument funkcji, 60
 wektorowa, 156
 wycentrowanie, 99
 wyświetlenie zawartości, 52
 zapisanie danych, 51
 zmienna\$kolumna, 53, 55
 zmienna[wiersz,kolumna], 56

Ź

źródło danych, 52

Analiza statystyczna w środowisku R dla początkujących

Na dźwięk słów „analiza statystyczna” spora grupa studentów dostaje nerwowych drgawek, a fascynatom rozszerzają się oczy. Ta dziedzina nauki budzi skrajne emocje. Skomplikowana, czasochłonna, wymaga wielu obliczeń, jednak dostarcza istotnych informacji, których właściwą interpretację można przekuć w sukces w różnych dziedzinach życia. Statystyka, tak jak każda inna dyscyplina, dzięki postępującej informatyzacji otrzymała ogromne wsparcie w postaci specjalistycznego oprogramowania. Ta książka poświęcona została środowisku R, którego możliwości w stosunku do ceny są naprawdę imponujące, ponieważ środowisko to jest darmowe i rozpowszechniane na licencji GPL!

To podręcznik dla osób, które chciałyby się zaprzyjaźnić z analizą statystyczną i środowiskiem R. W trakcie lektury dowiesz się, jak zainstalować i skonfigurować środowisko R, a potem w nim pracować. Nauczysz się importować dane ze źródeł zewnętrznych, korzystać z funkcji wieloargumentowych, obliczać średnie, standardowe odchylenia i wiele innych wartości, typowych dla statystyki. Ponadto błyskawicznie opanujesz sposoby modelowania za pomocą prostej i wielokrotnej regresji liniowej oraz zobaczysz, jak interpretować wyniki. Poznasz także zaawansowane mechanizmy tworzenia wykresów, grafów i diagramów oraz odkryjesz najlepsze źródła informacji na temat środowiska R. Brzmi interesująco? Przekonaj się sam!

Zaprzyjaźnij się z analizą statystyczną!

- Instalacja środowiska R
- Elementarz użytkownika R
- Importowanie danych zewnętrznych
- Wykorzystanie zmiennych
- Wyświetlanie, zapamiętywanie i odczytywanie zawartości obszaru roboczego
- Funkcje wieloargumentowe i o zmiennej liczbie argumentów
- Średnie, standardowe odchylenia, zakresy
- Prosta i wielokrotna regresja liniowa
- Interpretacja zmiennych interakcyjnych
- Prognozowanie wyników za pomocą modeli regresji
- 5 kroków do przeprowadzenia analizy w środowisku R
- Tworzenie wykresów, grafów i diagramów
- Przydatne zasoby w sieci

helion.pl
księgarnia
internetowa

Nr katalogowy: 7362



Księgarnia internetowa:
<http://helion.pl>



Zamówienia telefoniczne:
0 801 339900
0 601 339900



Helion

Sprawdź najnowsze promocje:

● <http://helion.pl/promocje>

Książki najchętniej czytane:

● <http://helion.pl/bestsellery>

Zamów informacje o nowościach:

● <http://helion.pl/nowosci>

Helion SA
ul. Kosciuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
<http://helion.pl>

sięgnij po WIĘCEJ



KOD KORZYŚCI

ISBN 978-83-246-3409-5



Cena 39,90 zł

9 788324 634095

Informatyka w najlepszym wydaniu