

Wczytywanie plików danych do R, formatowanie i drukowanie

Wczytywanie plików Microsoft Excel do R

Następujące trzy sposoby można wykorzystać do wczytania pliku Excela do R jako obiektu:

a) Plik może być przechowywany jako plik *.txt lub *.csv, a następnie read.table(), scan() lub read.csv() może być użyty do wczytania pliku do R.

b) Plik Excel można wczytać bezpośrednio do R, korzystając z następujących instrukcji:

```
> library(RODBC)

> my.object <- odbcConnectExcel("Filename.xls")

# with the full path included in Filename

> sqlTables(my.object)

#to obtain information about my.object.

> SENIC.data <- sqlFetch(my.object,"Sheet1")
```

gdzie Sheet1 jest nazwą arkusza zawierającego dane. Ta nazwa pojawia się pod nagłówkiem TABLE_NAME podczas wykonywania instrukcji sqlTables(my.object) i należy ją wpisać dokładnie tak, jak się pojawia, ale bez znaku \$ na końcu. Obiekt R SENIC.data zawiera teraz wymagany plik danych.

c) Wybierz dane w Excelu. Skopiuj wybrany zakres. W R:

```
read.table(file="clipboard")
```

Wskazówka: Uważaj z pustymi komórkami w Excelu: może być potrzebne trochę przygotowania pliku Excel.

Wczytywanie innych plików danych do R

Pakiet R Foreign() udostępnia funkcje do wczytywania danych z innych pakietów do R:

```
> library(foreign)

> objects(name="package:foreign")

[1] "data.restore" "lookup.xport" "read.arff"

[4] "read.dbf" "read.dta" "read.epiinfo"

[7] "read.mtp" "read.octave" "read.S"

[10] "read.spss" "read.ssd" "read.systat"

[13] "read.xport" "write.arff" "write.dbf"

[16] "write.dta" "write.foreign"
```

Zapoznaj się z plikami pomocy tych funkcji, aby wczytać dane binarne R, format SAS XPORT, format Weka Attribute-Relation File Format, rodzinę języków baz danych Xbase dBase, Clipper i FoxPro, Stata, pliki Epi Info i EpiData, przenośne arkusze Minitab, tekst Octave pliki, pliki data.dump, które zostały utworzone w wersji S 31, pliki zapisu lub eksportu SPSS, zestawy danych SAS do konwersji do formatu ssd2 i pliki Systat.

Wysyłanie wyjścia do pliku

Funkcja `sink("nazwapliku")` może być użyta do przekierowania wyjścia, które normalnie pojawia się w konsoli, do pliku. Opcja `options(echo=TRUE)` zapewnia, że instrukcje języka R będą również zawarte w pliku. Instrukcja `sink()` sprawia, że dane wyjściowe pojawiają się ponownie w konsoli. Czym różnią się funkcje `write(x)` i `sink("nazwapliku")`? Przystuduj dokładnie argumenty `write()`.

Zapisywanie obiektów R do transportu

Funkcja `R save(file=)` zapisuje zewnętrzną reprezentację obiektów R do określonego pliku. Obiekty te można odczytać z powrotem z pliku za pomocą funkcji `load(file=)`. Sprawdź, jak działają te dwie funkcje, zapoznając się z plikami pomocy. Przystuduj również, jak `dump(list, file="outdata")` i `source("outdata")` działają w R. Funkcje `dump()` i `source()` są bardzo przydatne do przenoszenia obiektów R między komputerami. Należy zauważyć, że źródło instrukcji („instrukcje”) może być użyte podczas sesji R do wykonania instrukcji w instrukcjach pliku zewnętrznego.

Wykorzystanie pliku .Rhistory i funkcji history()

Plik `.Rhistory` jest tworzony w tym samym folderze, w którym istnieje `.RData`. Można go przeglądać za pomocą dowolnego edytora tekstu lub programu MS Word i jako taki zapewnia dokładny zapis wszystkich działań na konsoli R (okno poleceń). Przystuduj plik pomocy funkcji `history()`.

Ponowna edycja poleceń

- a) Użyj strzałek w górę iw dół, aby przywołać poprzednie polecenia. Klawisze Delete, Backspace, Home i End do edycji.
- b) Zwróć uwagę na użycie okna skryptu do wykonywania całych funkcji lub tylko wybranych instrukcji.

Dostosowane wyświetlania

Podstawowym narzędziem do indywidualnego wyświetlania jest funkcja `cat()`. Ta funkcja może być używana do wysyłania komunikatów do konsoli lub do pliku. Zwróć uwagę na różne argumenty dostępne dla `cat()`:

- i. Domyślnie wyjście jest wyświetlane na ekranie; aby wyjście było skierowane do pliku, użyj argumentu `file = "nazwa pliku wraz ze ścieżką"`.
- ii. Domyślnie wyjście skierowane do pliku zastępuje poprzednią zawartość pliku; użyj argumentu `append=TRUE`, aby dołączyć nowe wyjście do poprzedniej zawartości.
- iii. Użyj `sep="xx"`, aby automatycznie wstawić znaki między nienazwanymi argumentami funkcji `cat()` w danych wyjściowych.
- iv. Aby automatycznie wstawić znaki nowej linii do wyniku, użyj `fill=TRUE`.
- v. Argument etykiety= umożliwia wstawienie ciągu znaków na początku każdego wiersza wyjściowego. Jeśli etykiety są wektorami, ich wartości są używane cyklicznie.

- Wpisz dzisiejszą datę podaną przez funkcję `date()` w formularzu

„Dzisiejsza data to: dzień tygodnia, xx, miesiąc, 20xx”.

jako nagłówek do pliku. Wskazówka: przywołaj funkcje `cat()`, `match()`, `substring()`, `paste()`, `replace()`.

Formatowanie liczb

- a) Przystuduj, jak funkcje round() i signif() wraz z cat() mogą zostać użyte do ustawienia liczby drukowanych miejsc dziesiętnych.
- b) Przystuduj użycie opcji (cyfry=xx).
- c) Przystuduj, jak działa funkcja format(). Zwróć uwagę na użycie format() razem z paste() i cat().
- d) Co robi print()?
- e) Przystuduj plik pomocy write.table().
- f) Funkcje prmatrix() lub print() mogą być użyte do wyprowadzenia macierzy na konsolę podczas wykonywania funkcji. Jest to bardzo wygodne do sprawdzania wyników pośrednich. Określ, czym ta ostatnia funkcja różni się od cat().
- g) Zwróć uwagę na różnicę między > colnames(state.x77)

i

```
> format(colnames(state.x77))
```

- h) Przystuduj uważnie następujący przykład:

```
> format.mns <- format(apply(state.x77,2,mean))
```

```
> format.names <- format(colnames(state.x77))
```

```
> descrip.mns <- paste("Mean for variable",
```

```
format.names, " = ", format.mns)
```

```
> cat(descrip.mns, fill=max(nchar(descrip.mns)))
```

```
Mean for variable % Pop.<15 = 35.0898
```

```
Mean for variable % Pop. >75 = 2.2930
```

```
Mean for variable Disp. Inc. = 1106.7862
```

```
Mean for variable Growth = 3.7576
```

```
Mean for variable Savings = 9.6710
```

Tablice wyświetlania

Przystuduj poniższy przykład, jak przedstawić maksymalną i minimalną wartość zmiennych w zbiorze danych state.x77 w tabeli z nazwami krajów odpowiadającymi wartościami.

```
> mins <- apply(state.x77, 2, min)
```

```
> maxs <- apply(state.x77, 2, max)
```

```
> min.name <- character(ncol(state.x77))
```

```
> min.name
```

```
[1] "" "" "" "" "" "" "" ""
```

```
> for(i in 1:8)min.name[i] <-
```

```

rownames(state.x77)[state.x77[,i] == mins[i]][1]

> max.name <- character(8)

> for(i in 1:8)max.name[i] <-
rownames(state.x77)[state.x77[,i] == maxs[i]][1]

> my.table <- data.frame(mins, min.name, maxs, max.name)

> dimnames(my.table) <- list(names(mins),c("Minimum", "State with
Min","Maximum","State with Max"))

> colnames(my.table)[3] <- paste(" ",
colnames(my.table)[3])

> my.table

```

	Minimum	State with Min	Maximum	State with Max
Population	365.00	Alaska	21198.0	California
Income	3098.00	Mississippi	6315.0	Alaska
Illiteracy	0.50	Iowa	2.8	Louisiana
Life Exp	67.96	South Carolina	73.6	Hawaii
Murder	1.40	North Dakota	15.1	Alabama
HS Grad	37.80	South Carolina	67.3	Utah
Frost	0.00	Hawaii	188.0	Nevada
Area	1049.00	Rhode Island	566432.0	Alaska

Alternatywną wersję powyższej tabeli można uzyskać z następującymi instrukcjami:

```

> cat(paste(format(c(" ", "Statistic", " ", names(mins))),
+ format(paste(" ",c(" ", "Minimum", " ",format(mins)))),
+ format(c("State having","Minimum", " ",min.name)),
+ format(paste(" ",c(" ", "Maximum", " ",format(maxs)))),
+ format(c("State having","Maximum", " ",max.name))), fill=TRUE)
      State having State having
Statistic Minimum Minimum Maximum Maximum
Population      365.00 Alaska      21198.0 California
Income          3098.00 Mississippi 6315.0 Alaska
Illiteracy       0.50 Iowa          2.8 Louisiana
Life Exp        67.96 South Carolina 73.6 Hawaii
Murder           1.40 North Dakota 15.1 Alabama
HS Grad         37.80 South Carolina 67.3 Utah
Frost            0.00 Hawaii       188.0 Nevada
Area            1049.00 Rhode Island 566432.0 Alaska

```

Wprowadź niezbędne zmiany w powyższych wierszach kodu, aby poprawić odstępy między kolumnami

Komunikacja z systemem operacyjnym

Przestuduj, jak działa funkcja system(), korzystając z instrukcji DOS: „time”, „date” i „dir”.

Podpowiedź: Najpierw przestuduj plik pomocy funkcji R system(), a następnie następujące instrukcje:

```

> system(paste(Sys.getenv("COMSPEC"),"/c","time /t"),

```

```
show.output.on.console=TRUE,invisible=TRUE)
```

```
> system(paste(Sys.getenv("COMSPEC"),"/c","date /t"),
```

```
show.output.on.console=TRUE,invisible=TRUE)
```

```
> system(paste(Sys.getenv("COMSPEC"),"/c","dir c:\\"),
```

```
show.output.on.console=TRUE,invisible=TRUE)
```

Funkcja R `system()` może być również używana razem z Notatnikiem do tworzenia pliku tekstowego podczas sesji R:

```
> system(paste(Sys.getenv("COMSPEC"),"/c","notepad c:\\
```

```
temp\\test.txt"),show.output.on.console=TRUE, invisible=TRUE)
```

a) Użyj `system()`, aby utworzyć plik tekstowy bez kończenia sesji R.

b) Użyj `system()`, aby napisać funkcję `myfile.exists()`, która sprawdza, czy istnieje określony plik.