

Programowanie w R - Praca na listach

Listy

Dane mogą przybierać bardziej skomplikowaną strukturę od "tradycyjnej" postaci ramki danych. W takich przypadkach najczęściej stosuje się listy, które wymagają nieco odmiennej filozofii postępowania. . .

Tworzenie listy

Konstruktorem listy jest polecenie `list()`, w którym jako argumenty definiujemy dowolne nowe obiekty. Dposób tworzenia nowej listy przedstawia poniższy schemat dla obiektu składającego się z 4 elementów.

```
lista = list(ramka = data.frame(A = 1:5, B = runif(5)),
            wektor = 0:10/10,
            imie = "Anna",
            0:3)
```

Nie wszystkie obiekty listy muszą być nazwane. Który z nich nie ma nazwy?

Wywoływanie obiektów listy

Do obiektów listy odwołujemy się operatorem `[[]]`.

```
lista[[1]] # pierwszy element listy
```

```
##   A         B
## 1 1 0.1177653
## 2 2 0.5917274
## 3 3 0.4639012
## 4 4 0.1999783
## 5 5 0.4879800
```

```
lista[[4]] # czwarty element listy
```

```
## [1] 0 1 2 3
```

Możliwe jest także odwołanie się do kolejnych pod-elementów listy, indeksując je w odpowiedni sposób dla danej klasy obiektu:

```
lista[[1]][1,]
```

```
##   A         B
## 1 1 0.1177653
```

Jeśli przez przypadek zamiast podwójnego nawiasu kwadratowego użyjemy pojedynczego nawiasu kwadratowego uzyskamy ponownie obiekty klasy `list`

```
lista[1]
```

```
## $ramka
##   A         B
## 1 1 0.1177653
## 2 2 0.5917274
## 3 3 0.4639012
## 4 4 0.1999783
## 5 5 0.4879800
```

Drugim sposobem odniesienia do elementów listy jest ich wywołanie poprzez nazwę po znaku \$ (ale tylko jeśli pod-obiekt listy posiada nazwę):

```
lista$wektor

## [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
# to samo co: lista[[2]]
```

Modyfikowanie / Dodawanie obiektów listy

Elementy listy mogą być modyfikowane operatorami przypisania. Zmieńmy zawartość drugiego obiektu naszej listy wartościami od -2 do +2 z interwałem co 1:

```
lista[[2]] = -2:2
lista[[2]] # wyświetla obecna zawartosc drugiego elementu listy

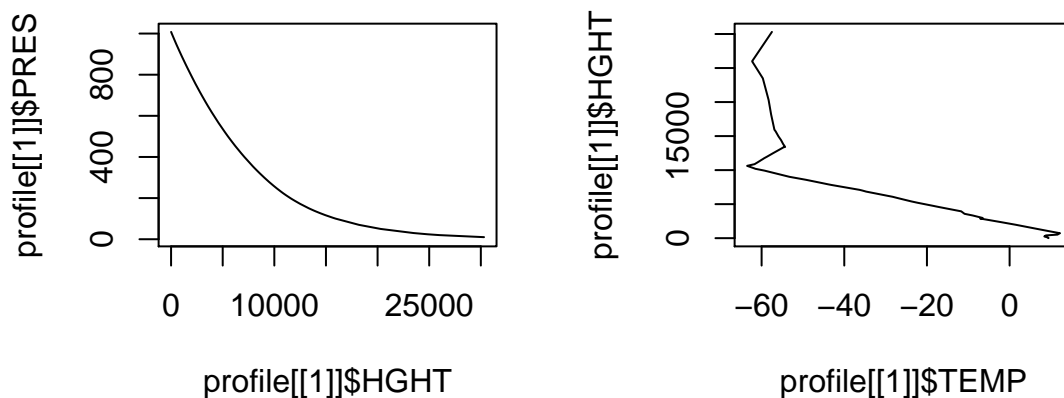
## [1] -2 -1 0 1 2
```

Dodawanie nowego elementu listy wygląda bardzo podobnie jak w przypadku innych obiektów **R**. Jeśli chcemy dopisać 5-ty i kolejne elementy do istniejącej już listy możemy to zrobić poprzez:

```
lista[[5]] = "Ala"
lista$stolice = c("Berlin", "Londyn", "Warszawa")
```

Listy w praktyce - dane radiosondażowe

- Zainstaluj pakiet `climate`.
- Korzystając z instrukcji pomocy dla funkcji `sounding_wyoming()` pobierz jedno (dowolne) sondowanie atmosferyczne z obszaru Polski i nazwij je `profile`
- Z zapisanego obiektu będącego listą dwóch obiektów wyciągnij pierwszy przechowujący dane pomiarowe i zapisz jako `dane`
- Jaka była maksymalna wysokość do której w danym dniu doleciał balon meteorologiczny?
- Stwórz wykres liniowy z obiektu `dane` przedstawiający zmienność wysokości i ciśnienia atmosferycznego
- Stwórz kolejny wykres liniowy z obiektu `dane` przedstawiający zmienność temperatury powietrza na poszczególnych wysokościach



```
## max. wysokosc znaleziona w profilu: 30310
```

- Wykonaj ponownie ostatni z wykresów tym razem korzystając z danych przechowywanych w obiekcie (liście) `profile`.
- Wyciągnij z obiektu `profile` wartość indeksu `SWEAT`

Listy w praktyce - automatyzacja pobierania danych z użyciem pętli

- Napisz pętlę `for`, która pozwoli na pobieranie danych radiosondażowych dla kilkunastu (np. 10-15 kolejnych dni) sezonu letniego. (rok, miesiąc, zakres dni – dowolnie, stacja – jedna z 3 polskich)
- Pobierane dane zapisz do nowoutworzonej listy (np. `lista`), w której każdy kolejny dzień będzie zapisywany w postaci nowego pod-obiektu.
- Korzystając z pętli `for` określ maksymalną znaną wysokość do której dolatywał w tym czasie balon meteorologiczny
- Korzystając z pętli `for` określ średnią wysokość do której dolatywał w tym czasie balon meteorologiczny
- Korzystając z pętli `for` określ średnią i maksymalną wartość chwiejności atmosfery (parametr CAPE)

Wektoryzacja

Początkowo praca na listach może być kłopotliwa ze względu na skomplikowaną strukturę jaką mogą przybierać dane wewnątrz list. Niezbędne staje się w takich sytuacjach połączenie funkcjonalności instrukcji `lapply()` z własnymi funkcjami, które będą działać na podobiektach.

Zacznijmy od prostej listy, z 3 podobnymi obiektami w środku. Następnie sprawdzimy jak wywołać dowolną funkcję działającą na tych obiektach (np. `max`, `min`, `summary`, `range`, itp.) wraz z funkcją `lapply()`:

```
test = list(A = runif(10), B = rnorm(10), C = 1:10) # tworzymy listę
lapply(test, max)
```

```
## $A
## [1] 0.744627
##
## $B
## [1] 1.6876
##
## $C
## [1] 10
```

```
lapply(test, summary)
```

```
## $A
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01129 0.25276 0.43741 0.42914 0.65439 0.74463
##
## $B
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.4598 -0.7127 0.2754 0.2280 1.1363 1.6876
##
## $C
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.00   3.25   5.50   5.50   7.75  10.00
```

Wyciąganie interesujących elementów z listy

Wewnątrz instrukcji `lapply` możemy także wywołać własną funkcję, która będzie w bardziej precyzyjny sposób odwoływać się do interesujących nas elementów. Poniżej stworzymy funkcję, która “wyciąga” element z każdego z elementów listy `test`:

```
lapply(test, function(x) x[2])
```

```
## $A
## [1] 0.2871804
```

```
##
## $B
## [1] 1.166947
##
## $C
## [1] 2
```

Wynik działania `lapply` również jest listą. Można ją przekonwertować do wektora za pomocą `unlist()`:

```
a = lapply(test, function(x) x[2])
unlist(a)
```

```
##          A          B          C
## 0.2871804 1.1669474 2.0000000
```

Alternatywnie można zastosować komendę `do.call()`, która wykona dowolną operację łączącą z listy do wybranego, docelowego formatu:

```
do.call(rbind, a) # tworzymy ramkę danych
```

```
##      [,1]
## A 0.2871804
## B 1.1669474
## C 2.0000000
```

```
do.call(c, a) # tworzymy wektor
```

```
##          A          B          C
## 0.2871804 1.1669474 2.0000000
```

Zadanie

- Użyj `lapply` do wyciągnięcia z listy danych radiosondażowych tylko ramkę danych z pomiarami
- Następnie rozwiń własną funkcję do postaci “wyciągającej” max. wysokość radiosondażu
- Do powyższych instrukcji dopisz fragment kodu, który stworzy ramkę danych w postaci: Observation time, max_wysokosc, Lifted Index, CAPE

Wskazówka – dobrym zwyczajem jest wyciągnięcie pierwszego elementu listy i przetestowanie działania własnej funkcji na pojedynczym obiekcie listy. W momencie, w którym uda się potwierdzić jego działanie można sprawdzić jak zachowuje się nasza funkcja na pełnej liście w połączeniu z `lapply`.

Zadanie “z gwiazdką” - łatwiejsze

- Napisz instrukcję, która stworzy “długą” ramkę dla wszystkich pomiarowych danych radiosondażowych (pobranych uprzednio do obiektu `lista`) z datą (lub tylko dniem) sondażu jako kolejną kolumną.

Zadanie “z gwiazdką” - trudniejsze

- Napisz instrukcję, która będzie wyciągać wartości zmierzonych parametrów meteorologicznych na wysokości 500 hPa lub na wysokości najbliższej 500 hPa. Jako obiekt wynikowy zwróć ramkę danych z datą i poszukiwaną wartością/wartościami.