zanuRkuj w R

Przemyslaw.Biecek@gmail.com

IBM Polska / MIM UW / SmarterPoland

Plan zwiedzania:

-100m Sweave, programowanie objaśniające a automatyczne generowanie kolokwiów.

-200m Graficzne profilowanie kodu.

-300m Atrybuty, czyli jak przypiąć mu łatkę.

-400m Leniwa ewaluacja, zalety bycia leniwym.

-500m Środowiska i ekosystemy.

-600m Singleton - funkcja, która chciała więcej.

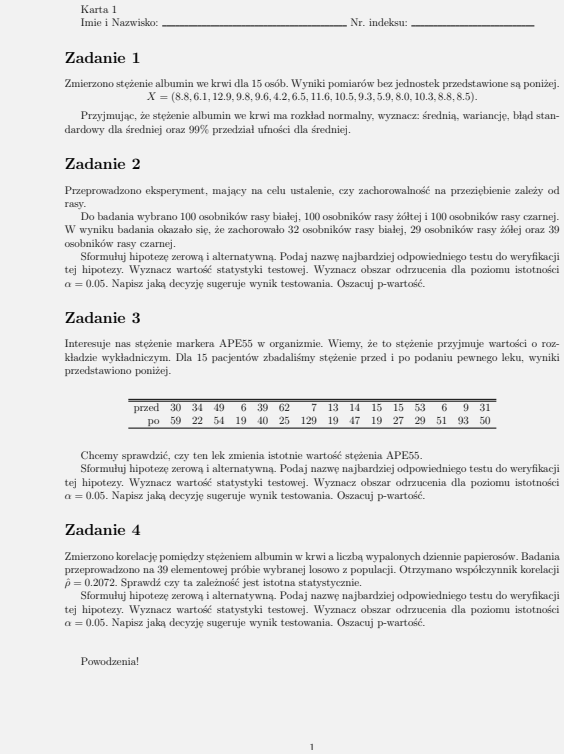
-700m Język w którym ***wszystko*** jest obiektem.

Termin *literate programing* zaproponował Donald Knuth, autor między innymi języka TEX. Opisuje on filozofię integracji kodu oraz czytelnego opisu kodu. Jest wiele implementacji tej filozofii, w R najpopularniejsza to Sweave. Plik

Snw zawierający kod LATEXjak i wstawki kodu R jest przetwarzany. Kod R jest wykonywany a wynik wklejany jest zamiast wstawki. Następnie plik tex może być przetworzony interepreterem LATEX'a.

Typowe zastosowanie Sweave to:

- Generowanie dokumentacji. Daje pewność, że wynik widziany w dokumentacji to dokładnie wynik wygnerowany przez aktualną wersje R. Używany w winietach.
- Generowanie raportów. Pozwala na uzyskanie pełnej automatyzacji generowania raportu z szablonu. Kod R jest często ukrywany.
- Generowanie kolokwiów i egzaminów.



Karta 1
Imie i Nazwisko: _____ Nr. indeksu: _____

Zadanie 1

Zmierzono stężenie albumin we krwi dla 15 osób. Wyniki pomiarów bez jednostek przedstawione są poniżej.
$X = (8.8, 6.1, 12.9, 9.8, 9.6, 4.2, 6.5, 11.6, 10.5, 9.3, 5.9, 8.0, 10.3, 8.8, 8.5)$.

Przyjmując, że stężenie albumin we krwi ma rozkład normalny, wyznacz: średnią, wariancję, błąd standardowy dla średniej oraz 99% przedział ufności dla średniej.

Zadanie 2

Przeprowadzono eksperyment, mający na celu ustalenie, czy zachorowalność na przeziębienie zależy od rasy.

Do badania wybrano 100 osobników rasy białej, 100 osobników rasy żółtej i 100 osobników rasy czarnej. W wyniku badania okazało się, że zachorowało 32 osobników rasy białej, 29 osobników rasy żółej oraz 39 osobników rasy czarnej.

Sformułuj hipotezę zerową i alternatywną. Podaj nazwę najbardziej odpowiedniego testu do weryfikacji tej hipotezy. Wyznacz wartość statystyki testowej. Wyznacz obszar odrzucenia dla poziomu istotności $\alpha = 0.05$. Napisz jaką decyzję sugeruje wynik testowania. Oszacuj p-wartość.

Zadanie 3

Interesuje nas stężenie markera APE55 w organizmie. Wiemy, że to stężenie przyjmuje wartości o rozkładzie wykładniczym. Dla 15 pacjentów zbadaliśmy stężenie przed i po podaniu pewnego leku, wyniki przedstawiono poniżej.

przed	30	34	49	6	39	62	7	13	14	15	15	53	6	9	31
po	59	22	54	19	40	25	129	19	47	19	27	29	51	93	50

Chcemy sprawdzić, czy ten lek zmienia istotnie wartość stężenia APE55.

Sformułuj hipotezę zerową i alternatywną. Podaj nazwę najbardziej odpowiedniego testu do weryfikacji tej hipotezy. Wyznacz wartość statystyki testowej. Wyznacz obszar odrzucenia dla poziomu istotności $\alpha = 0.05$. Napisz jaką decyzję sugeruje wynik testowania. Oszacuj p-wartość.

Zadanie 4

Zmierzono korelację pomiędzy stężeniem albumin w krwi a liczbą wypalonych dziennie papierosów. Badania przeprowadzono na 39 elementowej próbie wybranej losowo z populacji. Otrzymano współczynnik korelacji $\hat{\rho} = 0.2072$. Sprawdź czy ta zależność jest istotna statystycznie.

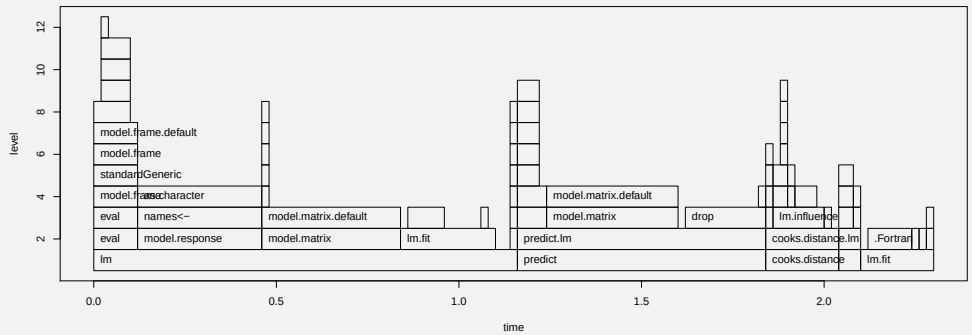
Sformułuj hipotezę zerową i alternatywną. Podaj nazwę najbardziej odpowiedniego testu do weryfikacji tej hipotezy. Wyznacz wartość statystyki testowej. Wyznacz obszar odrzucenia dla poziomu istotności $\alpha = 0.05$. Napisz jaką decyzję sugeruje wynik testowania. Oszacuj p-wartość.

Powodzenia!

1

Funkcja Rprof() włącza/wyłącza zbieranie informacji o aktualnym stosie wywołań. Informacje z profilera zapisywane są do pliku tekstowego. Można je przedstawić tekstowo z użyciem funkcji summaryRprof() lub graficznie z użyciem pakietu profr.

```
library(profr)
N <- 500000
df <- data.frame(x=matrix(rnorm(2*N),N,2),y=rnorm(N))
# ten fragment kodu jest profilowany
Rprof("tmp.txt")
  model <- lm(y~.,data=df)
  modelp <- summary(model)
  modelp <- predict(model,df)
Rprof()
# wyświetla wyniki
summaryRprof("tmp.txt")
plot(parse_rprof("tmp.txt"))
# usuwa tymczasowy plik
unlink("tmp.txt")
```



Macierz to wektor z atrybutem dim.

Atrybuty określają jak dany obiekt będzie interpretowany. Dwu i wielowymiarowe macierze, tabele, ramki danych, to najczęściej wektory lub listy z odpowiednimi atrybutami. Obiekty w R mogą posiadać dowolna liczbę atrybutów.

Najczęściej używane atrybuty to:

- class - wektor określający klasy S3 obiektu,
- comment - opis obiektu,
- dim, dimnames, names - wymiary macierzy (macierz może mieć dwa lub więcej wymiarów),
- dimnames, names, colnames, rownames - nazwy wymiarów oraz nazwy wierszy/kolumn,
- levels - słownik poziomów zmiennej czynnikowej.

```
x <- 1:12
attr(x,"dim") <- c(3,4)
```

Gorliwa ewaluacja polega na tym, że operator przypisania wyznacza wartość wyrażenia, które ma być przypisane w chwili w której sam jest przetwarzany.

Lewniwa (odroczona) ewaluacja, polega na tym, że wyznaczenie wartości wyrażenia wykonywane jest w chwili gdy użyty zostanie symbol, do którego ta wartość ma być przypisana. Symbole w ewaluowanym wyrażeniu są wyszukiwane w kontekście/środowisku w którym dochodzi do ewaluacji. Odroczona ewaluacja nie gwarantuje, że wartościowania wyrażenia. Patrz poniższy przykład.

W R efekt odroczonej ewaluacji można uzyskać na kilka sposobów, np. funkcją delayedAssign() lub funkcją quote(). Wszystkie argumenty funkcji są ewaluowane z opóźnieniem.

```
# bardzo leniwa ewaluacja dla y
f <- function(x,y) x
f(cat("argument 1"), cat("argument 2"))
# jeszcze bardziej leniwa ewaluacja
f <- function(x) cat("Argument:", deparse(substitute(x)))
f(2+x^2)
```

Obiekty w R są przypisywane do symboli.
Symbole w R są pogrupowane w środowiska (przestrzenie nazw).

Środowiska tworzą hierarchiczną strukturę, której korzeniem jest puste środowisko (R_EmptyEnv). Aktualne środowisko na „poziomie konsoli" jest nazywane globalnym (.GlobalEnv).

Pakiety, funkcje, ramki danych (attach) mają lub mogą mieć własne przestrzenie nazw.

Funkcje w chwili tworzenia mają przypisane środowisko aktualne w chwili tworzenia. Podczas wywołania funkcja tworzy środowisko potomne do środowiska aktualnego w chwili wywołania (to mogą być trzy różne środowiska).

```
zewnetrzna <- function() {
  y <- 2
  function (x) cat(y)
}
y <- 0
zewnetrzna()()
```

Wykorzystamy mechanizm tworzenia nowej przestrzeni nazw, aby zbudować worek na zmienne, do którego przeciętny użytkownik nie będzie mógł się dostać.

Przestrzeń nazw stworzona przez anonimową funkcję będzie tym workiem, w tym przypadku tylko ze zmienną tmp.

Ten rejestr jest singletonem, wszystkie kopie rejestru odwołują się do tej samej przestrzeni nazw.

```
rejestr <- {function () {
  tmp <- 0

  get <- function ()    tmp
  set <- function (x)   tmp <<- x
  inc <- function ()    tmp <<- tmp + 1

  list(get=get, set=set, inc=inc)
}}()
rejestr$set(10)
rejestr$get()
rejestr$inc()
```

R czerpie z języków funkcyjnych, więc (prawie) wszystko tutaj jest funkcją.

```
2 + 2                              # jest równoważne
`+`(2,2)
`(`(sin,1)                         # jest równoważne
sin(1)
`[<-`(x,2,2)                       # jest równoważne
x[2] <- 2
`for`(i, 1:3, cat(i,"\n"))         # jest równoważne
for (i in 1:3) cat(i,"\n")
```

R jest też językiem obiektowym, więc wszystko jest obiektem.

```
f <- function(x,y) {
  z <- x+y
  z
}
attr(f, "source")
body(f); args(f); class(f)
```

Przy okazji chciałbym powiedzieć o ...