exams: Automatic Generation of Exams in R

Studium doktoranckie

Wydział Geodezji Górniczej i Inżynierii Środowiska

Ewelina Saran

Zastosowanie

- Funkcja exams pozwala na automatyczne generowanie egzaminów, w tym pytań wielokrotnego wyboru i problemów arytmetycznych. Egzaminy mogą być produkowane w różnych formatach, w tym PDF.

LaTeX

Oprogramowanie do zautomatyzowanego składu tekstu, a także związany z nim język znaczników, służący do formatowania dokumentów tekstowych i tekstowo-graficznych (na przykład: broszur, artykułów, książek, plakatów, prezentacji, a nawet stron HTML). W istocie LaTeX nie jest samodzielnym środowiskiem programistycznym. Jest to jedynie zestaw makr stanowiących nadbudowę dla systemu składu TeX, automatyzujących wiele czynności związanych z procesem poprawnego składania tekstu.

Tworzenie tekstu w LaTeX-u opiera się na zasadzie WYSIWYM (What You See Is What You Mean - To co widzisz jest tym, o czym myślisz). Od zasady WYSIWYG odróżnia go to, że autor tekstu określa jedynie logiczną strukturę dokumentu (tzn. zaznacza, gdzie zaczyna się rozdział, co jest przypisem itp.), natomiast samym graficznym "ułożeniem" tekstu na stronie zajmuje się TeX, zwalniając tym samym użytkownika z tego zadania.

Argumenty funkcji exams

exams(file, n = 1, nsamp = NULL, dir = NULL, template = "plain",inputs = NULL, header = list(Date = Sys.Date()), name = NULL,quiet = TRUE, edir = NULL, tdir = NULL, control = NULL)

Gdzie:

file: określa listę / wektor zawierający nazwy plików Sweave

```
R> myexam <- list ("boxplots"
+                  C ("confint", "ttest", "tstat"),
+                  C ("ANOVA", "regression""),
+                  "scatterplot"
+                  "Relfreq")
```

Egzaminy generowane przez myexam składają się z pięciu ćwiczeń: "boxplots", "scatterplot„ i "relfreq„ są zawsze zawarte, drugie ćwiczenie jest losowo wybierane z "confint", "ttest", "tstat„. Podobnie trzecie jest losowo wybierane z "anova" i "regression„.

Ta strategia pobierania próbek jest przydatna, gdy istnieje kilka ćwiczeń dotyczących tego samego tematu.

n: liczba losowo generowanych egzaminów

dir: katalog wyjściowy do przechowywania wyników w plikach PDF

template: wektor nadrzędny plików LaTeX. Jeśli więcej niż jeden template jest określony, wyjściowy plik PDF jest tworzony dla każdego w każdym działaniu n

header: dodatkowe komendy dla zastąpienia exinput{header} w nadrzędnym pliku LaTeX. To musi być list() z command=wartość par, gdzie wartość może tez być ciągiem statycznym lub funkcją obliczania ciagu do indeksu *i* z *i*-tego egzaminu

Potrzebne przy kompilacji:

name: (wektor) przedrostek końcowego pliku PDF

quiet=TRUE usuwanie wyjściowe kiedy zostaje wywołany Sweave() i texi2dvi()

edir: ścieżka do ćwiczeniowego katalogu (domyślnie jest to bieżący katalog roboczy)

tdir: tymczasowy katalog gdzie wszystkie pliki są kopiowane i gdzie Sweave i texid2i() są wywoływane

control: dodatkowe opcje sterowania

Przykład

```
library("exams")
options(device.ask.default = FALSE)
## define an exams (= list of exercises)
myexam <- list(
"boxplots",
c("tstat", "ttest", "confint"),
c("regression", "anova"),
"scatterplot",
"relfreq"
)
if(interactive()) {
## compile a single random exam
(displayed on screen)
sol <- exams(myexam)
sol
}
```

Ta część planuje testy wielokrotnego wyboru

```
## generate multiple exams (stored in
output directory)
odir <- tempfile()
sol <- exams(myexam, n = 5, dir = odir,
template = c("exam", "solution"))
sol
## inspect solution for a particular exam
print(sol, 3)
if(interactive()) {
## modify control argument for printing
mymchoice.control <- list(mchoice.print =
list(True = LETTERS[1:5], False = "_"))
sol <- exams("boxplots", template =
"solution",
control = mymchoice.control)
sol
}
```

R Console

```
> myexam <- list(
+ "boxplots",
+ c("tstat", "ttest", "confint"),
+ c("regression", "anova"),
+ "scatterplot",
+ "relfreq"
+ )
> if(interactive()) {
+ ## compile a single random exam (displayed on screen)
+ sol <- exams(myexam)
+ sol
+ }

plain1
    1. Multiple choice: ae
    2. Multiple choice: abe
    3. Multiple choice: cde
    4. Multiple choice: bd
    5. Multiple choice: ade

> ## generate multiple exams (stored in output directory)
> odir <- tempfile()
> sol <- exams(myexam, n = 5, dir = odir, template = c("exam", "solution"))
> sol
```

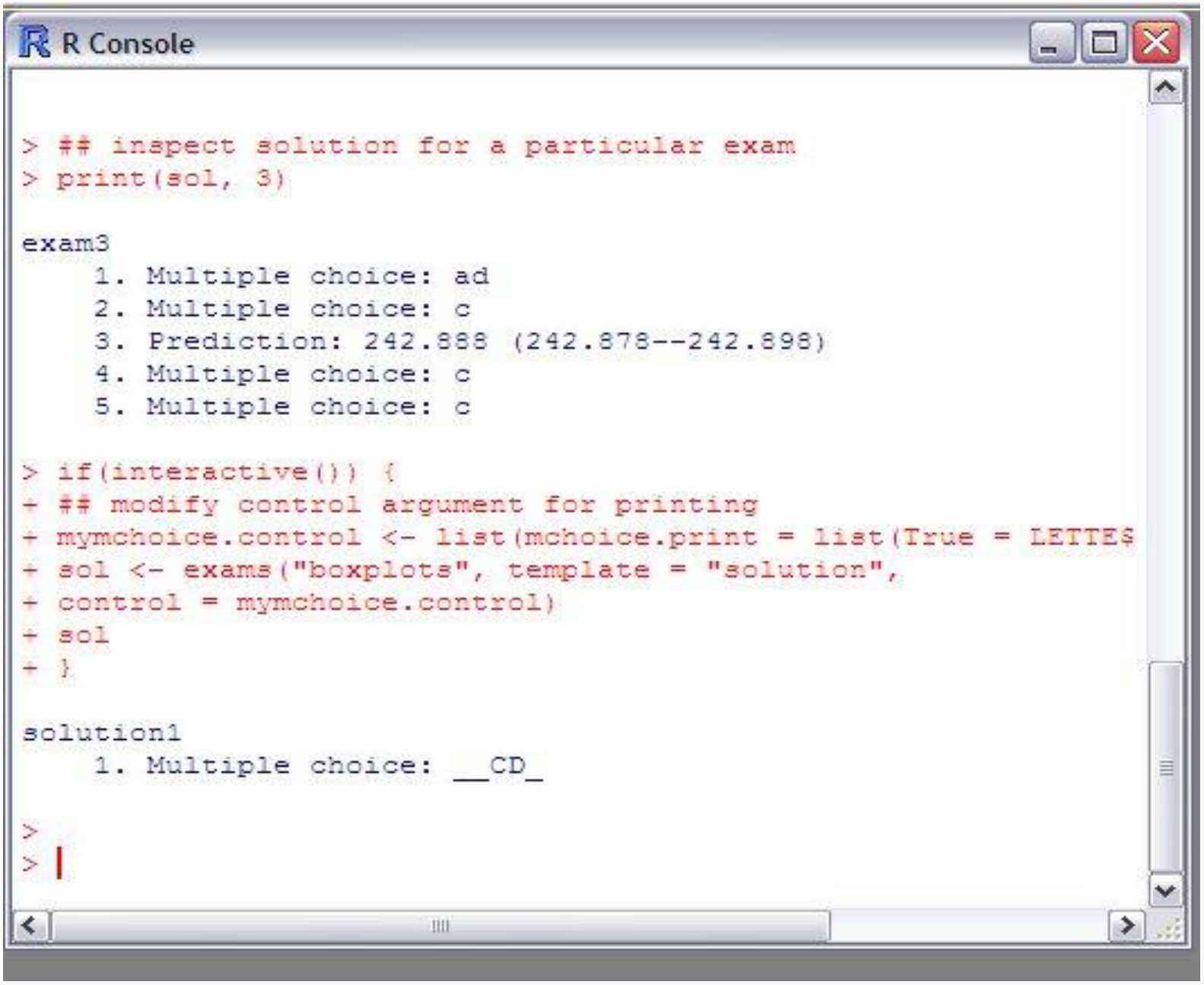
R Console

```
> ## generate multiple exams (stored in output directory)
> odir <- tempfile()
> sol <- exams(myexam, n = 5, dir = odir, template = c("exam", "solution"))
> sol

exam1
     1. Multiple choice: ace
     2. Confidence interval: [132.177, 136.823] ([132.167--132.187, 136.813--$
     3. Prediction: 273.564 (273.554--273.574)
     4. Multiple choice: acde
     5. Multiple choice: ade

exam2
     1. Multiple choice: bd
     2. Multiple choice: abc
     3. Multiple choice: ac
     4. Multiple choice: bcd
     5. Multiple choice: acde

exam3
     1. Multiple choice: ad
     2. Multiple choice: c
     3. Prediction: 242.888 (242.878--242.898)
     4. Multiple choice: c
     5. Multiple choice: c
```


R Console
> ## inspect solution for a particular exam
> print(sol, 3)
exam3
1. Multiple choice: ad
2. Multiple choice: c
3. Prediction: 242.888 (242.878--242.898)
4. Multiple choice: c
5. Multiple choice: c
> if(interactive()) {
+ ## modify control argument for printing
+ mymchoice.control <- list(mchoice.print = list(True = LETTE$
+ sol <- exams("boxplots", template = "solution",
+ control = mymchoice.control)
+ sol
+ }
solution1
1. Multiple choice: __CD_
>

Dziękuję za uwagę