Biblioteka Sweave w akcji, czyli jak pozytywnie zaskoczyc szefa albo (niekoniecznie pozytywnie) studentow

czas zabierze:
Przemysław Biecek

Literate programming

Literate programming (programowanie objaśniające) to styl programowania, w którym wymaga się by programy były czytelne i zrozumiałe tak dla komputera jak i dla zwykłych ludzi (czytaj: nie tylko dla nawiedzonych programistów). Pierwsza implementacja powstała w roku 1981 i była opracowana przez D. Knutha.

Plik źródłowy przygotowany zgodnie z tym stylem programowania zawiera instrukcje w języku programowania (początkowo Pascal, później też C, teraz dostępne są parsery dla każdego popularnego języka programowania) oraz opisy czytelne dla człowieka.

Używając pakietu **Sweave** możemy w jednym pliku źródłowym umieszczać komendy R i opisy złożone w LaTeX.

Przykład

Zobaczmy plik *przyklad.Snw*.

```
> # Generujemy kod LaTeXa z plików Snw
> Sweave("przyklad.Snw")
Writing to file przyklad.tex
Processing code chunks ...
1 : echo term verbatim
2 : term verbatim eps pdf
You can now run LaTeX on 'przyklad.tex'
```

Zobaczmy teraz plik *przyklad.tex*.

Zobaczmy teraz plik *przyklad.pdf*.

Przykład

Zobaczmy plik *przyklad.Snw*.

```
> # Generujemy kod LaTeXa z plików Snw
> Sweave("przyklad.Snw")
Writing to file przyklad.tex
Processing code chunks ...
1 : echo term verbatim
2 : term verbatim eps pdf
You can now run LaTeX on 'przyklad.tex'
```

Zobaczmy teraz plik *przyklad.tex*.

Zobaczmy teraz plik *przyklad.pdf*.

Przykład

Zobaczmy plik *przyklad.Snw*.

```
> # Generujemy kod LaTeXa z plików Snw
> Sweave("przyklad.Snw")
Writing to file przyklad.tex
Processing code chunks ...
1 : echo term verbatim
2 : term verbatim eps pdf
You can now run LaTeX on 'przyklad.tex'
```

Zobaczmy teraz plik *przyklad.tex*.

Zobaczmy teraz plik *przyklad.pdf*.

Przykład

Zobaczmy plik *przyklad.Snw*.

```
> # Generujemy kod LaTeXa z plików Snw
> Sweave("przyklad.Snw")
Writing to file przyklad.tex
Processing code chunks ...
1 : echo term verbatim
2 : term verbatim eps pdf
You can now run LaTeX on 'przyklad.tex'
```

Zobaczmy teraz plik *przyklad.tex*.

Zobaczmy teraz plik *przyklad.pdf*.

Chunks, wstawki

Składnia wstawki jest następująca:

```
<< argumenty >>=
kod R
@
```

Modyfikując argumenty wstawek, możemy określać jak formatowane mają być wyniki.

Np. jeżeli nie chcemy, by wynik wykonania fragmentu kodu był wpisywany do pliku wynikowego, to za argumenty wstawki należy podać *echo=false, results=hide*.
Jeżeli wynikiem poleceń R jest wykres, to aby wykres ten znalazł się w pliku wynikowym, należy za argument wstawki podać argument *fig=true*.

Chunks, wstawki

```
<<>>=
print(by(wiek, plec, summary))
@
```

Wyniki ze wstawek są umieszczane w otoczeniu **Schunk**.

```
\begin{Schunk}
\begin{Sinput}
> print(by(wiek, plec, summary))
\end{Sinput}
\begin{Soutput}
INDICES: kobieta
Min. 1st Qu. Median Mean 3rd Qu. Max.
23.00 32.00 47.00 46.38 57.00 75.00
------------------------------------------------
INDICES: mezczyzna
Min. 1st Qu. Median Mean 3rd Qu. Max.
22.00 30.00 43.00 41.97 53.00 74.00
\end{Soutput}
```

Chunks, wstawki

Jeżeli wynikiem wstawki jest wykres / rysunek to we wstawce podajemy argument *fig=TRUE*.

```
<<fig=TRUE, echo=FALSE >>=
boxplot(wiek plec, data = dane, col=l̈ightgrey")
@
```

w wyniku przetwarzania stworzone zostaną pliki **przyklad-XXX.pdf** i **przyklad-XXX.eps** a w pliku wynikowym pojawi się komenda

```
\begin{center}
\includegraphics{przyklad-002}
\end{center}
```

Chunks, wstawki

Jeżeli wynikiem wstawki jest kod LaTeX (np. używając funkcji *xtable(xtable)*), to we wstawce podajemy argument *results=tex*.

```
<< results=tex,echo=false >>=
library(xtable)
mat=matrix(1:9,3,3,,list(LETTERS[1:3],LETTERS[6:8]))
xtable(mat, caption="Tabela kolejnych liczb");
@
```

w wyniku przetwarzania otrzymamy

```
\begin{tabular}{rrrr}
\hline
& F & G & H \\ \hline
A & 1 & 6 & 11 \\
B & 2 & 7 & 12 \\
C & 3 & 8 & 13 \\
\end{tabular}
\caption{Tabela kolejnych liczb}
```

Chunks, wstawki

Do tworzenia raportów przydać może się też funkcja *toLatex(utils)* lub funkcja *toBibtex(utils)* przygotowująca tekstową reprezentacje w języku LaTeXdla obiektu będącego jej argumentem.
Używanie wstawek jest wygodne dla większych fragmentów kodu R. Dla krótkich wygodniej używać *Sexpr*.

```
Wynik do wyliczenia w poleceniu \Sexpr
$$
\frac{1-log(3)}{\sqrt 5}= \Sexpr{(1-log(3))/5^5}
$$
```

w wyniku przetwarzania zostanie zamieniony na

```
Wynik do wyliczenia w poleceniu \Sexpr
$$
\frac{1-log(3)}{\sqrt 5}= -0.0441007561757451
$$
```

Chunks, wstawki

- Oczywiście format *pdf* to nie jedyny możliwy format prezentacji wyników.
- Istnieje też możliwość automatycznego generowania wyników w formacie *HTML*.
- Więcej informacji znaleźć można w pomocy dla funkcji *HTML(R2HTML)*.

Case study: Kolokwia dla studentów

Problem:

- Przygotowujemy kolokwium dla 100 osób.
- Sala jest na tyle mała, że zagęszczenie studentów przekracza granice pozwalającą na kontrolowanie niechcianej komunikacji pomiędzy studentami.
- Nie chcemy (o ile nie będziemy zmuszeni) terroryzować sali wyrzucając rozgadane osobniki za drzwi.
- Jak przygotować zadania, żeby studenci nie spisywali odpowiedzi od kolegów?

Rozwiązanie:

- Użyj Sweave!