

Shiny w biznesie:

wprowadzenie do webowego środowiska dostarczania analiz i prezentacji ich wyników

Cezary Dendek, IPI PAN
@cezden

Spotkania Entuzjastów R,
Warszawa, 20.03.2014

Plan prezentacji

- Shiny: łatwe tworzenie aplikacji webowych w R
 - Szybkie wprowadzenie
 - Zalety i wady
- Wykorzystanie potencjału technologii do dostarczania analiz

Shiny: szybkie wprowadzenie

- Czym jest Shiny (<http://www.rstudio.com/shiny/>)?
 - pakiet R
 - lekki serwer www (*node.js*)
 - środowisko przeglądarki www (*Twitter Bootstrap*)
 - abstrakcja komunikacji przeglądarki z serwerem
 - (środowisko dostarczania aplikacji, *Shiny Server*)

Shiny: szybkie wprowadzenie

Struktura aplikacji

- Model/View
- *View*: definicja interfejsu użytkownika (typowo plik **ui.R**)
opisuje jak wyświetlić aktywne komponenty na stronie
- *Model*: definicja logiki (typowo plik **server.R**)
opisuje jak wypełnić aktywne komponenty treścią

Shiny: szybkie wprowadzenie

Programowanie reaktywne

- Idea działania: programowanie reaktywne
 - Model/View/Controller (w pewnym sensie)
 - automatyczna zmiana wartości wyrażeń w odpowiedzi na akcje użytkownika

Shiny: szybkie wprowadzenie

Programowanie reaktywne

- Idea działania: programowanie reaktywne
 - w praktyce (teoretycznej):
 - w definicji logiki wyrażamy zależności pomiędzy komponentami

$$\text{np. } X = A + B$$

- w przypadku zmiany wartości reaktywnego komponentu A:
 - przeliczeniu ulegnie X
 - i wszystkie komponenty zależne.

Shiny: szybkie wprowadzenie

User Interface

- Zbudowany w oparciu o bibliotekę Twitter Bootstrap
- HTML5, JS, CSS, AJAX...
- w praktyce: wygląda “ładnie” na większości współczesnych przeglądarek (komputery i tablety)
- łatwo rozszerzalny* i integrowalny z bibliotekami wizualizacyjnymi*
* o ile zna się JavaScript
- Showroom możliwości: <http://www.rstudio.com/shiny/showcase/>

Shiny: zalety

- Łatwe tworzenie przyjemnego dla oka UI
- Łatwe rozszerzanie
- Separacja logiki i interfejsu
- Aplikacje można uruchomić wszędzie (np. na tablecie)
 - w przeciwieństwie do Tcl/Tk, GTK, Javy....
- możliwość debugowania w RStudio
- możliwość załączania dokumentów RMarkdown

Shiny: wady

- Definicja interfejsu w stylu LISP
 - nieco uciążliwe
 - przy rozbudowanych aplikacjach trzeba być skrupulatnym
- “zamknięta” webowość
 - brak linków do podstron
 - interfejs dla ludzi, nie dla maszyn:
nie zbudujemy w oparciu o niego usług sieciowych
(ale w połączeniu z OpenCPU.org...)

Zastosowania

Kilka słów o analityce...

- Kluczowe słowa: iteracyjność, reużywalność i powtarzalność oraz wygoda i optymalizacja działań
- Iteracyjność: zazwyczaj wielokrotnie powtarzamy analizy
- Reużywalność: bieżące wyniki wtedy, gdy są potrzebne
- Powtarzalność: skąd takie wyniki/wnioski?

Zastosowania

Iteracyjność

- Zazwyczaj wielokrotnie powtarzamy analizy
 - nowa wiedza, nowe dane, nowe modele
 - Data Mining: CRISP DM
 - konserwacja modeli: czy ciągle są aktualne?
 - bieżąca analityka

Zastosowania

Reużywalność

- Bieżące wyniki wtedy, gdy są potrzebne
- Dostępność on-line dla analityków biznesowych
- “Analityka reaktywna”

Zastosowania

Reużywalność

- Microsoft Excel:
 - szybkość, zwinność, prostota
 - z tej perspektywy R jest “silosem informacyjnym”
 - dane + parametry -> R (ładowanie, analiza, eksport) -> wyniki -> -> Excel (ładowanie, ...) -> ... SAS ... -> ... R ...
 - częsta duplikacja implementacji i “egzotyczne rozwiązania”
- Powszechna reużywalność -> szybsze wnioski -> \$

Zastosowania

Powtarzalność

- Repeatable research
- Wiele analiz w obrębie tego samego problemu
 - Jak je powtórzyć?
 - Czy są aktualne?
 - Na jakiej próbce danych się opierają?

Zastosowania Shiny

- Jedna aplikacja - wiele spójnych analiz
- Możliwość ściągnięcia danych i wyników
- Dobra praktyka:
 - kod analityczny w specjalizowanym pakiecie,
 - wizualizacja i wygodny interfejs w bibliotece Shiny
 - wersjonowanie danych, aplikacji i wniosków

Showcase

Dziękuję za uwagę...

...i zapraszam do dyskusji