Informatika pro moderní fyziky (11)
web scraping; API; zadání zápočtových úloh

František HAVLŮJ

e-mail: haf@ujv.cz

ÚJV Řež
oddělení Reaktorové fyziky a podpory palivového cyklu

akademický rok 2014/2015
10. prosince 2014

Obsah

Obecně:

- ke každému zadání jsou k dispozici vzorová data
- já to budu testovat i na datech jiných
- očekávám, že všechno proběhne na jedno spuštění skriptu / rake tasku
- každý má k dispozici jeden pokus řádný a jeden opravný

Klasifikace

- F - nejde to spustit, ani pro zadaná data to v podstatných bodech nesplňuje zadání
- E - pro zadaná data to funguje, ale pro jiná čísla to nechodí
- D - obecně to funguje, ale stejně chybí drobnosti ze zadání
- C - všechno funguje jak má
- B - funguje a navíc jsou výstupy hezké a přehledné, soubory nejsou generovány “na velkou hromadu”, ale roztříděné do složek apod.
- A - kromě výše uvedeného jsou splněny i požadavky formy (správné odsazování, rozumná jména funkcí a proměnných) a efektivity (je to rozumně naprogramované - vhodné použití funkcí, datových struktur atd.)

Známka se snižuje o stupeň, pokud:

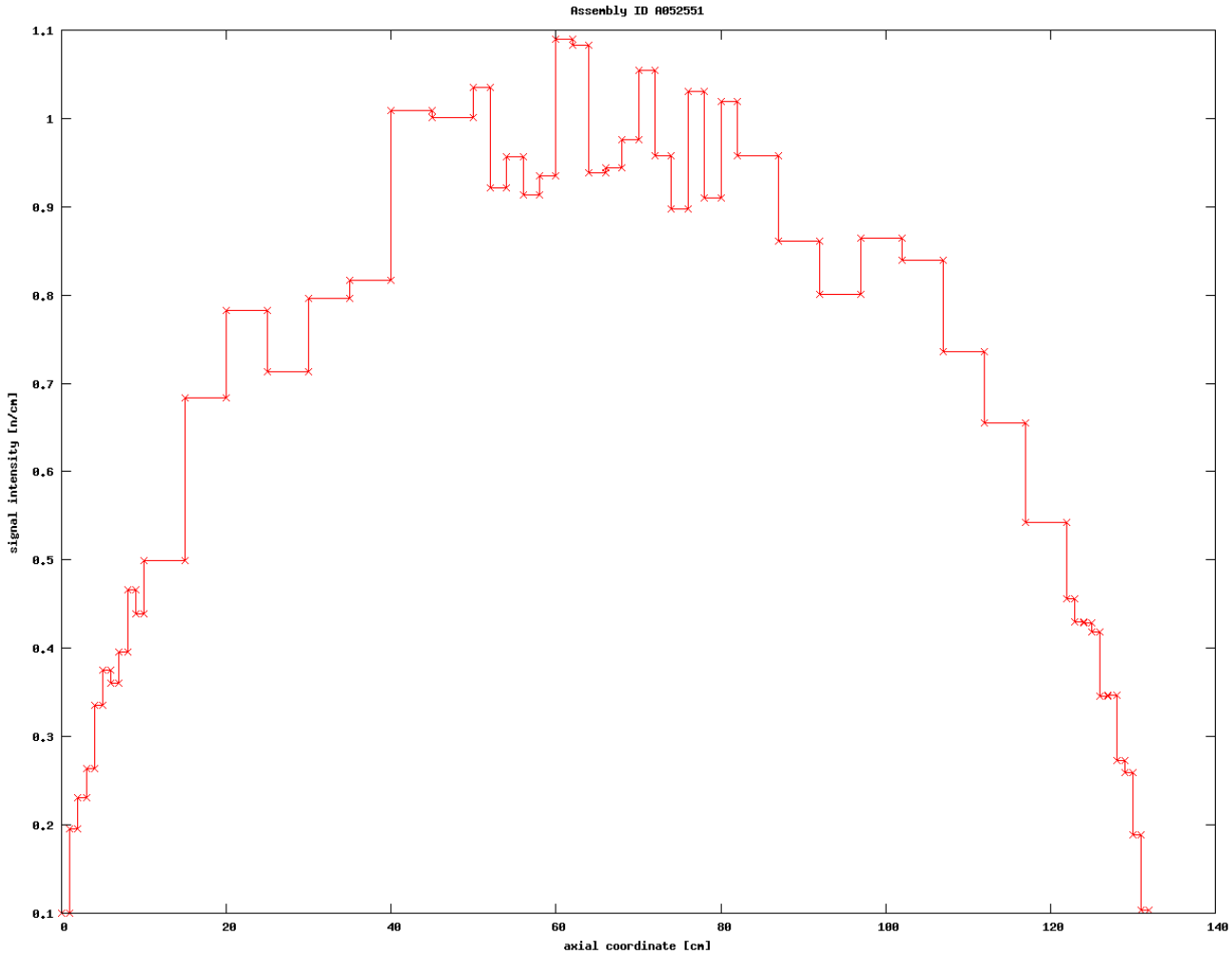
- jsou někde ve skriptech použity absolutní cesty, takže je budu muset upravovat (výjimkou jsou cesty k programům jako např. gnuplot, které ovšem musí být umístěny v proměnné někde na začátku skriptu (abych to nemusel lovit)
- bude v kódu něco, co limituje použití na OS Windows (backslash v cestě, kódování win1250 atd.)

(a podobně)

Gamma scanning palivových souborů

V souborech assembly*.csv jsou uloženy profily z gamma scanů. na prvním řádku je identifikátor kazety jednotlivé hodnoty jsou odděleny libovolným počtem mezer nebo novým řádkem – jedná se o integrální hodnoty signálu z jednotlivých nódů (tedy intenzita * výška nódu)
Axiální nodalizace je následující: 10 nódů po 1 cm, 8 nódů po 5 cm, 16 nódů po 2 cm, 8 nódů po 5 cm, 10 nódů po 1 cm
Úkol:

- vykreslit axiální profily intenzity signálů pro všechny PS (podle vzoru)
- do jednoho grafu vykreslit profily pěti PS s největší celkovou aktivitou


Assembly ID A052551
signal intensity [n/cm]
axial coordinate [cm]
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
20
40
60
80
100
120
140

Databáze vzorků v laboratoři (a,b)

Záznamy o vstupu a výstupu vzorků ze skladu - systém zapisuje datum průchodu, ID vzorku a naměřený dávkový příkon (miliSv/den); pro každý vzorek jsou v souboru právě dva záznamy. Pokles dávkového příkonu předpokládejte exponenciální (A * e ̂(-Bx)).
Úkol:

- najít vzorek s celkovou nejvyšší a nejnižší dávkou
- vykreslit histogram rozložení celkových dávek
- vykreslete histogram délky pobytu vzorku v laboratoři
- vykreslete oblak (scatter plot) zobrazující vztah mezi délkou pobytu (osa x) a celkovou aktivitou (osa y)

Komiks!

Komiks XKCD http://xkcd.com

- vygenerovat hezké PDF s obsahem (obsahujícím názvy jednotlivých dílů); každý díl včetně popisku (img/alt nebo img/title atribut)
- navíc HTML dokument umožňující prohlížení na jedné stránce (bez scrollování) - tedy rozumně vymyšlený seznam v levém sloupci (rozklikávací po částech, aby se nemuselo scrollovat), tlačítko dopředu+zpět

Univerzální vykreslovač

V adresáři "data"se nachází blíže neurčený počet CSV souborů se záznamem časového průběhu signálů z detektorů. V prvním řádku je záhlaví popisující jednotlivé sloupce, tedy například takto:

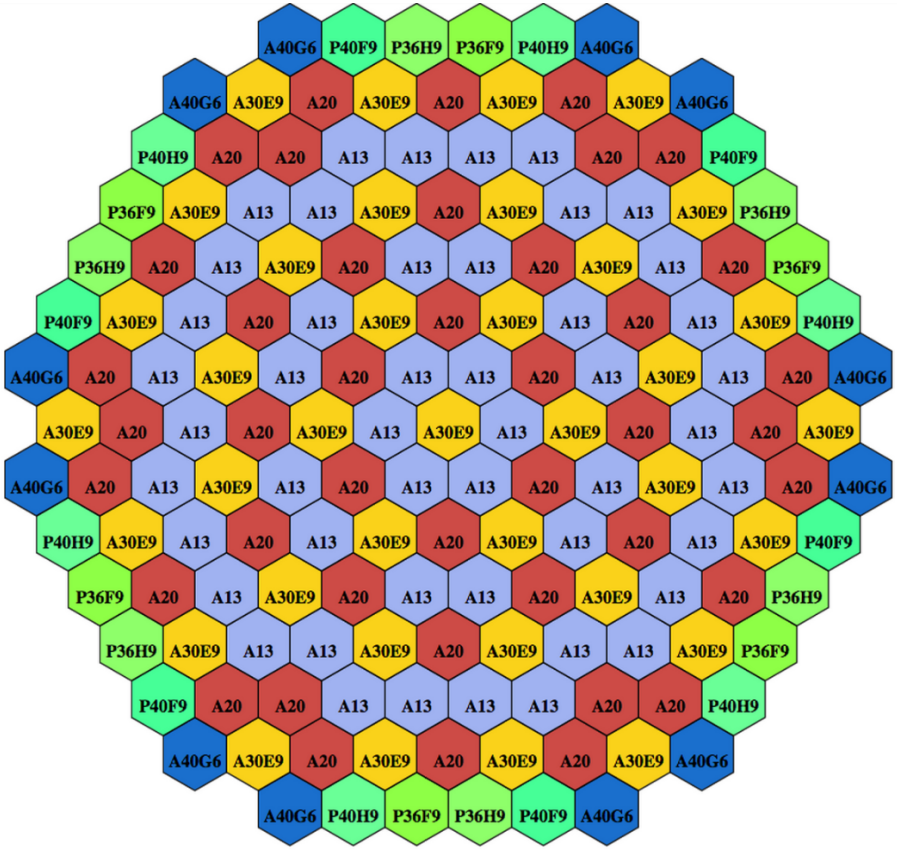
```
#y4 y1 y2 y3 time
1.1059 0.2212 0.1896 0.6777 0.01
0.2399 0.4539 0.428 1.1479 0.02
```

Sloupec "time"je přítomen právě jeden (nicméně pokaždé na jiné pozici).
Každý CSV soubor vykreslete do grafu, na ose X je čas, na ose Y jednotlivé signály – co soubor to graf, všechny signály z jednoho souboru vykreslené najednou. V legendě názvy sloupců.

Kartogramy pro VVER-1000

Na základě textového kartogramu jedné šestiny aktivní zóny VVER-1000 (levá dolní šestina + centrální PS) vykreslete kartogram celé zóny s barvičkami podle typu palivového souboru. Co dodat.
Příklad:

```
*     *   A30E9  A20   A13    A20   A30E9   A13    A30E9
   *  A40E6    A20        A13   A30E9   A13    A20
        *   P40E9   A30E9     A13    A20   A13
              *   P36E9   A20      A13   A30E9
                    *   P36E9   A30E9   A13
                          *  P40E9   A20
                               *  A40E6
                                    *
```


A40G6 P40F9 P36H9 P36F9 P40H9 A40G6
A40G6 A30E9 A20 A30E9 A20 A30E9 A20 A30E9 A40G6
P40H9 A20 A20 A13 A13 A13 A13 A20 A20 P40F9
P36F9 A30E9 A13 A13 A30E9 A20 A30E9 A13 A13 A30E9 P36H9
P36H9 A20 A13 A30E9 A20 A13 A13 A20 A30E9 A13 A20 P36F9
P40F9 A30E9 A13 A20 A13 A30E9 A20 A30E9 A13 A20 A13 A30E9 P40H9
A40G6 A20 A13 A30E9 A13 A20 A13 A13 A20 A13 A30E9 A13 A20 A40G6
A30E9 A20 A13 A20 A30E9 A13 A30E9 A13 A30E9 A20 A13 A20 A30E9
A40G6 A20 A13 A30E9 A13 A20 A13 A13 A20 A13 A30E9 A13 A20 A40G6
P40H9 A30E9 A13 A20 A13 A30E9 A20 A30E9 A13 A20 A13 A30E9 P40F9
P36F9 A20 A13 A30E9 A20 A13 A13 A20 A30E9 A13 A20 P36H9
P36H9 A30E9 A13 A13 A30E9 A20 A30E9 A13 A13 A30E9 P36F9
P40F9 A20 A20 A13 A13 A13 A13 A20 A20 P40H9
A40G6 A30E9 A20 A30E9 A20 A30E9 A20 A30E9 A40G6
A40G6 P40H9 P36F9 P36H9 P40F9 A40G6

Obsah

HTML scraping

- získávání informací z webu, které nám někdo nechce dát
- umíme číst HTML (knihovna `nokogiri`, případně `ox`), umíme stahovat soubory, takže dobrý
- pozor, občas se hodně informací nahrává až zpožděně přes Ajax a člověk musí použít tzv. *headless browser*, např `capybara` (ze zkušenosti: sázkové weby)

Připomenutí obecného postupu

- najdu URL, které mě zajímá
- na stránce hledám vhodný CSS selektor, abych se dostal k tomu, co potřebuju
- stáhnu data, která mě zajímají

Nalezení vhodného selektoru

- v principu hledám minimální formu – aby tomu vyhovovalo to, co potřebuju, ale nic jiného
- pomůžou mi vývojářské nástroje v prohlížeči
- obvykle to jde hodně snadno, hlavně pomocí `class` atributů – `element.css("div.comicsImage")` nebo tak něco
- klidně to můžu "dofiltrovat" až ve skriptu, selektor nemusí být bezchybný

Práce s XML / HTML

Knihovna `nokogiri`

```
doc = Nokogiri::HTML(File.open("redmeat.html"))
doc.css("li.archiveImage a").each do |x|
  url = x.attributes['href']
  ...
end
```

Knihovna open-uri

umožňuje otevírat URL jako soubory

```
require 'open-uri'
doc = Nokogiri::HTML(File.open("http://redmeat.com"))
```

Stahování dat

Součást standardní knihovny – `open-uri`

```
require 'open-uri'
File.open(local_filename, 'wb') do |f2|
  open(remote_url, 'rb') do |f1|
    f2.write f1.read
  end
end
```

Komiks – TWP

- `http://threewordphrase.com/`
- otevřu si archiv, tam snadno najdu seznam stránek
- na každé stránce je snadné najít ten obrázek (teda ne úplně, ale skoro)
- rovnou můžu stáhnout i popisek (atribut `title`)

Obsah

K čemu to?

- spousta informací na webu je poskytována ve strojově čitelné formě
- API – rozhraní mezi aplikacemi
- s využitím webových služeb naše možnosti exponenciálně rostou
- spousta věcí se dá udělat jako *mashup* – sice nic neumím, ale umím to dát dohromady

Typy / formáty

- URL – rovnou dostanu např. obrázek po zadání správného URL
- XML – velmi obecný, ale komplikovaný formát (v̈ypadá jako HTML)̈
- JSON – velmi jednoduchý a kompaktní formát, vyvinutý pro JS (v podstatě jen číslo, řetězec, pole, hash)

URL API – google maps

- stačí správně vymyslet
- pozor na usage limits (v produkci je nutné lokální cache...)
- QR platba:
  `http://qr-platba.cz/pro-vyvojare/restful-api/#ge`
- Google Maps static API:
  `https://developers.google.com/maps/documentation`

JSON API – počasí

- `http://openweathermap.org/`
- aktuální počasí – `http://api.openweathermap.org/data/2.5/weather?q`
- úkol: vypište předpovězená minima a maxima teploty v následujících deseti dnech ve svém rodném městě
- `http://api.openweathermap.org/data/2.5/forecast/`
  ...

Práce s JSON

- v Ruby je k mání knihovna – `require 'json'`
- generování JSON: `hash.to_json`
- čtení JSON: `JSON[data]`
- hodí se i na serializaci (uložit si hash do souboru)

XML API – kalendář o-závodů

- ORIS API – http://oris.orientacnisporty.cz/API
- úkol: vypišme kalendář MTBO závodů v roce 2015
- `http://oris.orientacnisporty.cz/API/?format=xml&`
  ...

A to je vše, přátelé!