Universiteit van Stellenbosch
Departement Wiskundige Wetenskappe
Afdeling Rekenaarwetenskap

University of Stellenbosch
Department of Mathematical Sciences
Computer Science Division

Wetenskaplike Berekening 272 / Scientific Computing 272

Werkopdrag 9: Datastipwerk / Assignment 9: Data Plotting

2019-10-25 Opgestel deur Willem Bester Gemodereer deur Willem Visser

Instruksies / Instructions

1. Hierdie is ’n oopboek tuisopdrag. Enige bronne word toegelaat, maar u moet hulle as kommentaar in u kode erken.

   This is an open-book home assignment. Any resources are allowed, but you have to acknowledge them as comments in your code.

2. U mag enige Python-module gebruik wat in Narga geïnstalleer is, maar geen ander modules nie. Indien u oplossings nie in Narga werk nie, kry u nul.

   You may use any Python module that is installed in Narga, but no other modules. If your solutions do not work in Narga, you get zero.

3. U mag nie oplossings van enigiemand of enige plek kopieer nie; dit sluit studente uit vorige jaargange in. Sulke kopiëring word as plagiaat beskou.

   You may not copy solutions from anyone or anywhere; this includes students from previous years. Such copying is viewed as plagiarism.

4. U mag nie met enigiemand saamwerk of om hulp vra nie. In die besonder mag u nie kode met enigiemand deel of op Internet-forums om hulp vra nie. Dit word as plagiaat beskou.

   You may not collaborate with anybody, or ask anybody for help. In particular, sharing code with anybody and asking for help on Internet forums are not allowed. This is viewed as plagiarism.

5. U mag nie u oplossings vir dié werkopdrag op enige openbare bêreplek aanhou of op enige manier versprei nie. Dit word as die fasilitering van plagiaat beskou.

   You may not host your solutions to this assignment in any public repository, or distribute them in any way. This is viewed as facilitating plagiarism.

6. Die sperdatum is Vrydag, 25 Oktober 2019, om 17:00. Wanneer u klaar is, versamel u Python-lêers in ’n `tar.bz2`-argief en handig dié op die kursuswebwerf in. Maak seker dat u die korrekte lêers—en slegs die korrekte lêers—inhandig. Indien u die verkeerde lêers inhandig, word nul aan u toegeken.

   The deadline is Friday, 25 Oktober 2019, at 17:00. When you are finished, collect your Python files in a `tar.bz2` archive and submit this on the course website. Ensure that you submit the correct files—and only the correct files. If you submit the wrong files, you will receive zero.

7. Stoor u werk vir Vraag 1 in die lêer `fill_map.py`. Dit moet met twee argumente, die CSV-lêernaam en die SVG-kaartnaam (in dié volgorde), vanaf die bevelreël geroep kan word. Enige afvoer moet geskryf word as `out1.svg`, ens.

   Save your work for Question 1 to the file `fill_map.py`. It must be able to be called with two arguments, the CSV filename and the SVG map name (in this order), from the command line. Any output must be written as `out1.svg`, etc.

8. U werk sal vir plagiaat getoets word. Deur die verklaringskassie in die oplaaivorm te merk, verklaar u dat u die kursusbeleid oor plagiaat gelees en verstaan het, en dat u nie oneerlik was nie.

   Your work will be checked for plagiarism. By marking the declaration box on the upload form, you declare that you have read and understood the course policy on plagiarism, and that you have not cheated.

9. Indien daar enigsins plagiaat voorkom, sal alle betrokke partye verwys word vir dissiplinêre aksie insake die US Beleid oor Plagiaat.

   If there are any instances of plagiarism, all involved parties will be referred for disciplinary action according the SU Policy on Plagiarism.

Vrae / Questions

1. Die volksmond wil dit hê dat ’n prentjie ’n duisend woorde werd is. As wetenskaplikes gebruik ons dikwels grafieke, diagramme, vloeidiagramme en kaarte—nie net om ons idees aan ons wetenskaplike eweknieë te illustreer nie, maar ook aan die breë publiek.

   In popular parlance, a picture is worth a thousand words. As scientists, we often use graphs, diagrams, flow diagrams, and maps—not only to illustrate our ideas to our scientific peers, but also to the general public.

Vir dié werkopdrag moet u 'n choropleth-kaart [5] van Suid-Afrika maak volgens bevolkingsgrootte. Die kaart in Figuur 1 en bevolkingsdata (in CSV-formaat) volgens die 2011-sensus is beskikbaar vanaf die kursuswebwerf. Die kaart is in SVG-formaat [3, 7], wat gebaseer is op XML [9], mens-leesbaar is en vertoon kan word deur 'n aantaal Linux-programme, weblesers soos Chrome en Firefox ingesluit.

Die CSV-lêer het 'n "`Districts`"-kolom wat 'n unieke identifiseerder vir elkeen van die areas op die kaart gee. Dié identifiseerder word ook in die SVG-kaart gebruik as waarde van die `g`-element se `id`-attribuut. Byvoorbeeld, die CSV-lêer gee "`WC024`" as identifiseerder vir die Stellenbosch Munisipaliteit, en die SVG-kaart bevat die volgende element vir die Stellenbosch-distrik.

```
<g id="WC024">
    <path d="..." />
</g>
```

Die `path`-element beskryf die kromme wat die buitelyne van elke distrik op die kaart spesifiseer; u mag dit ignoreer. Wat u moet doen, is om vir elke distrik sy bevolking relatief tot die digsbevolkte distrik te bereken: $p = n/N$, waar n die bevolking van die betrokke distrik is, en N die bevolking van die distrik met die meeste mense is. U moet p gebruik om 'n geskikte kleur uit 'n kleurkontinuum te kies, en die distrik se kaartarea met dié kleur te vul. Dit word gedoen deur die `style`-attribuut van die distrik se `g`-element by te werk om die `fill`-eienskap met 'n gepaste kleur, gespesifiseer as 'n heksadesimale RGB-webkleur [8], te hê.

```
<g id="WC024" style="fill:#fff3e3">
    <path d="..." />
</g>
```

Python verskaf 'n XML-module [1] waarmee u die kaart se attribute kan bywerk. Die funksies en metodes wat u van nut behoort te vind, word in Tabel 1 gegee. Let op dat `Element.attrib` 'n assosiatiewe lys [2] is, wat soos 'n lys is wat u met enige datatipe, nie net heelgetalle nie, kan indekseer.

Die laaste stuk in die legkaart is om 'n gepaste kleur te kies. Gebruik Matplotlib se kleurafbeeldingbiblioteek `matplotlib.cm` hiervoor. Die figure gebruik die `OrRd` sekwensiële kleurskema [4]. Die funksie `rgb2hex` in `matplotlib.colors` kan kleure na RGB omskakel.

2. Opsioneel (vir ekstra punte): In Figuur 2, behalwe die groot metropolitaanse areas van die Wes-Kaap, KwaZulu-Natal en Gauteng, lyk dit of daar nie veel meer in Suid-Afrika aangaan nie. Daar mag dalk waarheid in dié stelling lê, maar ons is hier eerstens geïnteresseerd in die illustrasie van die data, nie die interpretasie nie.

For this assignment, you have to make a choropleth map [5] of South Africa according to population size. The map in Figure 1 and population data (in CSV format) from the 2011 census is available from the course website. The map is in SVG format [3, 7], which is based on XML [9] and human-readable, and can be displayed by a number of Linux applications, including web browsers such as Chrome and Firefox.

The CSV file has a "`Districts`" column that gives a unique identifier for each of the areas in the SVG map. This identifier is also used in the SVG map as value of `id` attribute in the `g` element. For example, the CSV file gives "`WC024`" as the identifier for the Stellenbosch Municipality, and the SVG map contains the following element for the Stellenbosch district.

The `path` element describes the curve that specifies the outline of the district in the map; you may ignore it. What you have to do is calculate, for each district, its population relative to the most populous district: $p = n/N$, where n is the population of the district in question, and N is the population of the most populous district. You must use p to choose a suitable colour[†] from a colour continuum, and fill the map area of the district with this colour. This is accomplished by updating the `style` attribute of the district's `g` element to have a `fill` property with an appropriate colour, specified as a hexadecimal RGB web colour [8].

Python provides an XML module [1] with which you can update the map's attributes. The functions and methods you should find useful are given in Table 1. Note that `Element.attrib` is a dictionary [2], which is like a list you can index with any data type, not only integers.

The last piece in the puzzle is to get an appropriate colour. For this, use Matplotlib's colour map library `matplotlib.cm`. The figures uses the `OrRd` sequential colour scheme [4]. The function `rgb2hex` in `matplotlib.colors` can convert colours to RGB.

Optional (for extra marks): In Figure 2, besides the major metropolitan areas of the Western Cape, KwaZulu-Natal, and Gauteng, nothing much else seems to be going on in South Africa. There may be truth to this statement, but here we are, first and foremost, interested in the illustration of the data, not its interpretation.

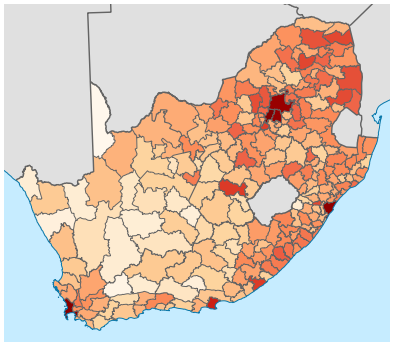
[†] Americans have been writing the word "colour" as "color" (without a *u*) since the American lexicographer Noah Webster decided to regularise English spelling single-handedly in the nineteenth century; see `http://www.merriam-webster.com/about-us/spelling-reform`. This means that most computer libraries use the spelling "color". In South Africa, we base our proper use on British orthography.

Tabel 1: Nuttige funksies en metodes in `xml.etree.ElementTree`, wat as `ET` ingevoer is.
Table 1: Useful functions and methods in `xml.etree.ElementTree`, which has been imported as `ET`.

FUNCTION/METHOD	DESCRIPTION
`ET.parse`	Parse an XML document into an element tree. A parameter specifies the name of the file to parse. It returns an `ET.ElementTree` (sic.) object.
`ET.register_namespace`	Register a namespace prefix; if called, it must be before any other `ElementTree` function. Strictly speaking, this is optional, but it does have some effect on the SVG file that is written as output.
`ET.Element.attrib`	A dictionary that contains the element's attributes.
`ET.Element.iter`	Create and return an iterator for the element, allowing us to loop over the elements with the element as root. If a tag as specified as parameter, visit only elements of the tag type. Note that `ET.ElementTree` also supports `iter`.
`ET.ElementTree.getroot`	Return the root element of this tree, which is an `ET.Element` object.

Figuur 1: 'n Kaart van Suid-Afrika.
Figure 1: A map of South Africa.

Figuur 2: 'n Lineêre kleurskaal vir Suid-Afrikaanse bevolkingsyfers.
Figure 2: A linear colour scale for South African population numbers.



Figuur 3: 'n Nie-lineêre kleurskaal vir Suid-Afrikaanse bevolkingsyfers.
Figure 3: A non-linear colour scale for South Africa population numbers.

Baie menslike verskynsels volg na beraming 'n magwet [6]. Gebruik dié idee om te eksperimenteer met nie-lineêre kleurskale, soos geïllustreer in Figuur 3.

Many human phenomena approximately follow a power law [6]. Use this idea to experiment with non-linear colour scales, as illustrated in Figure 3.

Verwysings / References

[1] Python Software Foundation. *The ElementTree XML API*. [Online; accessed 2017-10-12]. URL: `https://docs.python.org/2/library/xml.etree.elementtree.html`.

[2] Python Software Foundation. *The Python Tutorial: Dictionaries*. [Online; accessed 2017-10-12]. URL: `https://docs.python.org/2/tutorial/datastructures.html#dictionaries`.

[3] W3C. *Scalable Vector Graphics*. [Online; accessed 2017-10-12]. URL: `http://www.w3.org/Graphics/SVG/`.

[4] *Matplotlib Colormaps Reference*. [Online; accessed 2017-10-12]. URL: `http://matplotlib.org/examples/color/colormaps_reference.html`.

[5] Wikipedia. *Choropleth map — Wikipedia, the Free Encyclopedia*. [Online; accessed 2017-10-12]. 2004. URL: `https://en.wikipedia.org/wiki/Choropleth_map`.

[6] Wikipedia. *Power law — Wikipedia, the Free Encyclopedia*. [Onlines; accessed 2017-10-12]. 2004. URL: `https://en.wikipedia.org/wiki/Power_law`.

[7] Wikipedia. *Scalable Vector Graphics — Wikipedia, the Free Encyclopedia*. [Online; accessed 2017-10-12]. 2004. URL: `https://en.wikipedia.org/wiki/Scalable_Vector_Graphics`.

[8] Wikipedia. *Web color — Wikipedia, the Free Encyclopedia*. [Online; accessed 2017-10-12]. 2004. URL: `https://en.wikipedia.org/wiki/Web_colors`.

[9] Wikipedia. *XML — Wikipedia, the Free Encyclopedia*. [Online; accessed 2017-10-12]. 2004. URL: `https://en.wikipedia.org/wiki/XML`.