Universiteit van Stellenbosch
Departement Wiskundige Wetenskappe
Afdeling Rekenaarwetenskap

University of Stellenbosch
Department of Mathematical Sciences
Computer Science Division

# Wetenskaplike Berekening 272 / Scientific Computing 272

## Tutoriaal 8: Dataverwerking en Stipwerk / Tutorial 8: Data Processing and Plotting

2020-09-24/10-01 Opgestel deur Willem Bester Gemodereer deur Brink van der Merwe

## Agtergrond

Dié tutoriaal is 'n praktiese oorsig van die inlees, verwerking en uitskryf van datalêers in Python, asook van stipwerk met Matplotlib. Ter voorbereiding, bestudeer die relevante voorbeelde en dokumente op modulewebwerf, spesifiek dié wat op Matplotlib en CSV-lêers betrekking het. Ek stel voor dat u die gids `~/wb272/tut08` skep vir u werk aan dié tutoriaal.

## Background

This tutorial is a practical overview of reading, processing, and writing data files in Python, as well as of plotting with Matplotlib. To prepare, study the relevant examples and documents on the module website, specifically those involving CSV files and Matplotlib. I suggest you create the directory `~/wb272/tut08` for your work on this tutorial.

## Uitkomste

Wanneer u die tutoriaal voltooi het, behoort u in staat te wees om die volgende te doen: (1) data van CSV-lêers in te lees, terwyl opskrifte óf verbygegaan óf verwerk word, (2) data in tabelvorm te verwerk, (3) data in tabelvorm na CSV-lêers te skryf, en (4) data as staafkaarte en histogramme in Matplotlib te stip.

## Outcomes

When you have complete the tutorial, you should be able to do the following: (1) read data from CSV files, allowing headers to be either skipped or processed, (2) processing tabular data, (3) writing tabular data to CSV files, and (4) plot the data as bar charts and histograms in Matplotlib.

## Vrae / Questions

1. Laai die lêer `marks.csv` van die kursuswebwerf af. Dit bevat die punte van 'n denkbeeldige klasgroep in CSV-formaat ("comma separated values") wat vanuit 'n sigblad geskryf is. Skryf 'n Python-program `process_marks.py` wat die punte verwerk volgens die beskrywing hieronder. U program moet vanaf die bevelreël geloop kan word soos volg.

   Download the file `marks.csv` from the course website. It contains marks for an imaginary class group in CSV ("comma separated values") format, written from a spreadsheet. Write a Python program `process_marks.py` that processes the marks according to the description below. Your program must be able to be run from the command line as follows.

   ```
   python process_marks.py marks.csv output.csv
   ```

   Hier verwys die eerste argument `marks.csv` na die toevoerlêer en die tweede argument `output.csv` verwys na die afvoerlêer.

   Here, the first argument `marks.csv` refers to the input file, and the second argument `output.csv` refers to the output file.

Die toevoerlêer is in CSV-formaat en begin met twee rye opskrifte. Let op dat die tweede ry die totaal vir elke betrokke assesseringsgeleentheid gee. U moet die klaspunt soos volg bereken: (1) Alle werkopdragte het dieselfde gewig en, saam, tel hulle 64% van die klaspunt. (2) Alle toetse het dieselfde gewig en, saam, tel hulle 32% van die klaspunt; slegs die drie maksimumpunte tel egter, en die minimum moet buite rekening gelaat word. Wanneer u die klaspunte bereken het, voeg 'n nuwe klaspuntkolom aan die einde van die puntetabel by, en skryf die volledige puntetabel uit na die afvoerlêer in CSV-formaat.

Vir onverklaarbare redes woon studente nie altyd alle assesseringsgeleenthede by nie. Dit beteken daar is sekere selle sonder punte: Hulle moet as nulle gehanteer word.

Die Universiteit Stellenbosch skryf die volgende vir punte voor: (1) Ongelukkig mag die klaspunt nie kleiner as nul of groter as eenhonderd wees nie. (2) Alle punte moet tot die naaste heelgetal afgerond word, behalwe (3) tussen 35 en 50, waar slegs veelvoude van 5 toegelaat word. U moet die laaste voorvereiste hanteer deur tot die naaste veelvoud van 5 af te rond.

The input file is in CSV format, and starts with two rows of header data. Note that the second row gives the total for each particular assessment opportunity. You must calculate the final mark as follows: (1) All assignments are weighted equally, and together, constitute 64% of the final mark. (2) All tests are weighted equally, and together, constitute 32% of the final mark; however, only the maximum three test marks count, and the minimum must be disregarded. When you have computed the final marks, add a new final mark column to the end of the table of marks, and write the complete table to the output file in CSV format.

For reasons beyond comprehension, students do not always attend all assessment opportunities. This means there are cells without marks: They must be treated as zeros.

The University of Stellenbosch prescribes the following for marks: (1) Regrettably, the final mark may not be less than zero or more than one hundred. (2) All marks are rounded to the nearest integer, except (3) between 35 and 50, where only multiples of 5 are allowed. You must handle the last requirement by rounding to the nearest multiple of 5.

2. Stip die data van die vorige vraag as 'n histogram. Eksperimenteer met verskillende intervalle om die punte te groepeer. Dink u dis 'n sterk, gemiddelde of swak klasgroep? Motiveer u antwoord.

Plot the data of the previous question as a histogram. Experiment with different bin sizes to group the marks. Do you think this is a strong, average or weak class group? Motivate your answer.

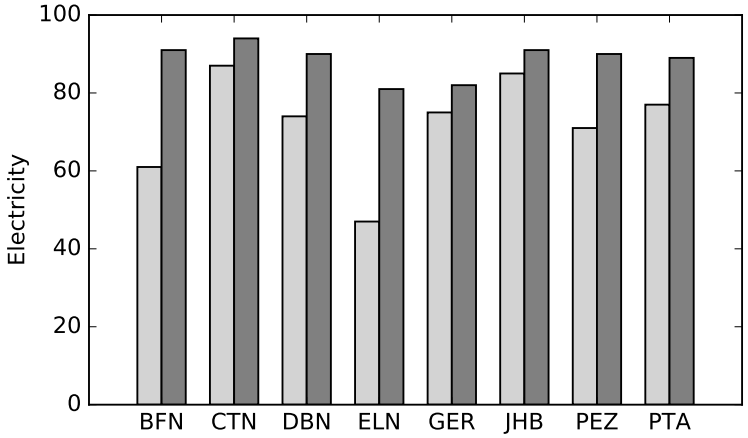
3. Skryf 'n Python-program wat die data in Tabel 1 as 'n staafdiagram soos in Figuur 1 stip. U mag die waardes in die tabel hardkodeer in u oplossing. Gebruik twee verskillende kleure van u keuse vir data van 1996 en 2011. Stoor u program in 'n lêer genaamd `plot_electricity.py`. U lêer moet van die bevelreelpor af geloop kan word, dit wil sê, die volgende bevel moet die staafdiagram vertoon:

Write a Python program that plots the data in Table 1 as a bar chart like the one given in Figure 1. In your solution, you may hardcode the values in the table. Use two different colours of your choice for the data of 1996 and 2011. Save your program to a file called `plot_electricity.py`. Your file must be runnable from the command line, that is, the following command must display the bar chart:

```
python plot_electricity.py
```

Tabel 1: Persentasies van huishoudings in die Suid-Afrikaanse metropole met elekstrisiteitstoevoer
Table 1: Percentages of households in the South African metropoles with electricity supply

| Metropole | 1996 | 2011 |
|---|---|---|
| Bloemfontein | 61 | 91 |
| Cape Town | 87 | 94 |
| Durban | 74 | 90 |
| East London | 47 | 81 |
| Ekurhuleni | 75 | 82 |
| Johannesburg | 85 | 91 |
| Port Elizabeth | 71 | 90 |
| Pretoria | 77 | 89 |




Figuur 1: Staafdiagram van elektrisiteitvoorsiening in 1996 en 2011.
Figure 1: Bar chart of electricity supply in 1996 and 2011.

4. Laai die lêer `marks.txt` van die kursuswebblad af. Neem aan die getalle in die lêer is die finale punte van 'n universiteitsmodule. Skryf 'n Python-program wat (1) die naam van puntelêer vanaf die bevelreël kry, (2) die punte in die lêer inlees, en (3) die punte as 'n histogram stip. Gebruik `marks.txt` as toetstoevoerlêer. Die histogram moet oor die gebied $[0, 100]$ wees. Oorweeg verskeie opstellings om die data op die beste manier aan te bied.

   Download the file `marks.txt` from the course website. Assume the numbers in the file are the final marks for a university module. Write a Python program that (1) gets the name of a marks file from the command line, (2) reads in the marks in the file, and (3) plots the marks as a histogram. Use `marks.txt` as test input file. The histogram must be over the range $[0, 100]$. Consider different settings to best present the data in the best way.