iSci 3A12 - Climate Change - Fall 2019

Individual Assignment II: Introduction to Scientific Computing

Date Assigned: 2019/11/06
Date Due: 2019/11/24
Weight: 5 points (~17% of final grade)

Introduction

Scientific computing—using computer algorithms to solve scientific problems—plays an important role in research for many modern scientific disciplines. By developing computer programs that collect, refine, analyze, and visualize experimental data, scientists are able to explore new scientific hypotheses and broaden their understanding of physical phenomena and relationships. Even in cases where datasets are small and relatively straightforward, using scripted, programming approaches can greatly improve the efficiency and reproducibility of scientific analysis.

Further to this point, it is very likely that as your scientific career progresses, you will find yourself in a situation where using a programmatic approach is required to complete your research work, and/or immensely beneficial and timesaving in comparison to other approaches. Though you may not have expertise with a specific programming application, having knowledge of the basics of programming and scientific computing, and the confidence to experiment and continue to learn will be invaluable assets. To this end, the goal of this workshop is to develop both of these aspects, while also introducing you to best practices in creating and maintaining scientific code.

The purpose of this workshop is to provide you with an introduction to the MATLAB software package and improve your knowledge of programming and scientific computing. In addition, you'll have an opportunity to further explore climate change over the past decade through a (programmatic) revisitation of the international weather station data used in your second assignment.

In-class workshops

On Wednesday, 06-November and 13-November, the class will participate in workshops aimed at introducing them to MATLAB software, programming, and the specifics of their assignment.

At the beginning of the class, you'll be asked to download the "Matlab_Data_Pack" zipfile from avenue and extract it to your working computer

Parts 1 & 2: Introduction

Follow along with Jay as he demonstrates the following:

- Working with variables
  - Assign numeric variables with desired names
  - Clearing variables
- Scalars, vectors, matrices
  - Create Scalars
  - Create row vectors and column vectors
  - Create Matrices
- Reference (index) elements within vectors and matrices
- Character arrays / strings
  - Create character arrays
  - Reference elements within
- Cell arrays
  - Create / reference cell arrays

Commands and operations

- Arithmetic operations on scalars / vectors / matrices
- Matrix multiplication
- Mathematical & statistical operations
- Concatenating strings

Scripts and Functions

- Functions vs scripts
- Running functions
- Commenting
- Using Help and other resources

Saving and Loading data

Plotting figures

- Saving figures

Deliverables

For your final deliverables, you are going to submit a few functions, a script, an image, and a short document.

The full list of deliverables (described below) that will be uploaded to Avenue to Learn is:

- A function named **my_lucky_numbers.m**
- A figure named **lucky_numbers.png**
- A function named **simple_stats.m**
- A script named **process_adelaide.m**
- A function named **plot_station_data.m**
- A **pdf document** that provides a very short reflection on your experience
- [BONUS] A script named **plot_global_trends.m**

These are described below:

1. From your introductory and intermediate work:

- Submit the function called **my_lucky_numbers.m** that you modified during the workshop to produce lucky numbers between 1 and 100 (this is a modified version of iSci_lucky_numbers.m)
- Submit the figure (in png format) that you created while investigating the output of your lucky number function. Title this figure **lucky_numbers.png**
- Submit a function called **simple_stats.m**, which takes a column vector of positive integers (which could include NaNs) as input and creates an output value (see more information in the simple_stats.m file that is started for you).

2. From your advanced work:

Complete the script **process_adelaide.m** so that it is properly commented, runs without errors, and creates two fully annotated and properly named figures. This script performs an analysis similar to that you performed in your first assignment, in that it calculates and plots temperature anomaly data for the ADELAIDE AIRP station. This script creates an additional 'barcode' plot that visualizes anomalies a bit differently. Here's what the script does to create the figures:

Figure 1

- Calculate the annual mean temperature and annual temperature anomaly.
- Plot the station time series of temperature anomaly as a line plot with markers
- Calculate the linear trend for the temperature anomaly and plot it as a line alongside the time series
- Style the time series, axes, legend, etc. Label the x-axis and y-axis.
- Add a title that displays the station name.
- Save the figure as **ADELAIDE AIRP_timeseries.png** in the /Figs directory.

Figure 2

- Calculate and plot a 'barcode chart' (ex.) of ***annual*** anomalies with a colorbar
- Label the X-axis by year
- Label the colorbar and axes.
- Style the time series, axes, legend, etc.
- Add a title that displays the station name.
- Save the figure as **ADELAIDE AIRP_annual_barcode.png** in the /Figs directory.

Using what you learned in the previous exercise, complete the function **plot_station_data.m**, which takes the input of a column vector list of station ID numbers, a reference start and end year (e.g. 1951 and 1980) and runs a loop that for each station in the list:

- Performs the same calculations as above (with data specific to that station)
- Creates the same two figures as described above with different data and filenames that correspond to the site (<station name>_timeseries.png and <station name>_barcode.png
- The function also outputs an n x 2 matrix (where n is the number of stations in station_numbers) consisting of the following information:
  - column 1 - the station number
  - column 2 - the mean temperatures for the given station over the **entire period**

3. [BONUS] Map of global temperature trends

This exercise is completely voluntary and worth up to 3 bonus marks on the assignment.
Create and submit a script called **plot_global_trends.m**. This script will load a filtered list of station data (***Data\stn_data_filtered.csv***) and fit a linear trend to each annual temperature time series. The script will then plot each station on a figure according to its longitude and latitude, with a colour reflecting the sign and magnitude of the temperature trend.
Here's what this function should perform:

- Load the station data file and the station list files.
- Get a list of all unique station names in the list
- Run a loop through all stations. For each station:
  - Calculate the annual average temperature and anomaly
  - Use the function ***polyfit*** to fit a linear trend to the time series
  - To a separate matrix, write the following information to a new row:
    - | Longitude | Latitude | slope (first output value of polyfit) |
- In a new figure, plot each station as a scatterplot of longitude and latitude. The colour of the marker should be as follows:
  - Slope > 0.01: Dark Red
  - 0.002 < Slope < 0.01: Light Red
  - -0.002 < Slope < 0.002: Green
  - -0.01 < Slope < -0.002: Light Blue
  - Slope < -0.01: Dark Blue
- Your figure should display on the screen and save to the /Figs directory as **global_trends.png**

3. (oh no, not another!) Reflection

Don't worry too much. This one is short and sweet. In 100 words or less, I would like to know what you thought of the workshop and the programming experience. Was this a useful exercise? Do you see the benefits of taking a scripted approach to analysis? What was/wasn't useful? Is there anything that you would have liked to have done differently? Include this submission as a .pdf document.

Assessment

Assessment Rubric

Element	Points
Introductory and intermediate work ● Functions run without error, are flexible to different inputs and produce correct outputs ● Figure elements are styled in an appealing and effective manner ● Function and script are appropriately commented so as to be understood by an external reviewer	 /20 /10 /10
Advanced work ● **Process_adelaide.m** runs as expected and creates appropriately styled and named figures ● **Plot_station_data.m** runs as expected and creates appropriately styled and named figures ● Function and script are appropriately commented so as to be understood by an external reviewer	 /20 /20 /10
Question response / Reflection ● The response is complete, well-composed, and shows a depth of thought.	 /10