Bielefeld University
Faculty of Technology

Genome Data Science
Dr. Daniel Dörr

Programming
Summer 2020

Exercises

Number 05, Submission Deadline: May 24, 2020

1. Preprocessing climate data (3P)

In the lecture, a preprocessed table of climate data from the GISTEMP website (https://data.giss.nasa.gov/gistemp/) has been used. In this exercise, you will perform the preprocessing yourself, using the original table that you can download from https://data.giss.nasa.gov/gistemp/tabledata_v4/GLB.Ts+dSST.csv.

Here are the first lines of the original file, which corresponds to a table stored in comma-separated-values (CSV) format:

```
Land-Ocean: Global Means
Year,Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec,J-D,D-N,DJF,MAM,JJA,SON
1880,-.17,-.24,-.09,-.16,-.09,-.20,-.17,-.09,-.14,-.23,-.21,-.17,-.16,***,***,-.11,-.15,-.19
1881,-.19,-.13,.04,.06,.07,-.18,.01,-.02,-.14,-.21,-.18,-.06,-.08,-.09,-.16,.05,-.06,-.18
1882,.17,.15,.05,-.16,-.14,-.22,-.15,-.06,-.14,-.23,-.15,-.35,-.10,-.08,.09,-.08,-.14,-.17
1883,-.28,-.36,-.12,-.17,-.17,-.07,-.06,-.13,-.20,-.10,-.22,-.10,-.16,-.19,-.33,-.15,-.08,-.18
1884,-.12,-.07,-.35,-.39,-.34,-.35,-.31,-.26,-.26,-.24,-.32,-.30,-.28,-.26,-.10,-.36,-.31,-.28
1885,-.58,-.32,-.25,-.41,-.44,-.42,-.32,-.29,-.27,-.22,-.22,-.09,-.32,-.34,-.40,-.37,-.35,-.24
1886,-.42,-.49,-.42,-.27,-.23,-.33,-.17,-.29,-.23,-.26,-.26,-.24,-.30,-.29,-.33,-.31,-.26,-.25
1887,-.70,-.55,-.34,-.33,-.29,-.23,-.24,-.34,-.24,-.34,-.25,-.32,-.35,-.34,-.50,-.32,-.27,-.28
1888,-.33,-.35,-.40,-.19,-.21,-.16,-.09,-.14,-.11,.03,.04,-.03,-.16,-.18,-.33,-.27,-.13,-.01
1889,-.07,.18,.07,.10,.00,-.09,-.07,-.19,-.23,-.25,-.32,-.28,-.10,-.08,.03,.06,-.12,-.27
```

After the first two (header) lines, each line reports (i) the year in which (ii) monthly temperature measurements have been collected, along with (iii) five additional aggregated measurements. Measurements that could not be collected are indicated by '***'.

Write Python code that converts this file into another CSV table where

1. years with incomplete measurements are discarded (1P),
2. the table is transposed, meaning that each column corresponds to measurements taken in a certain year (1P),
3. the row representing the column header lists the years, and (0.5 P)
4. only the 12 monthly measurements are reported (0.5P).

Store this table in a separate file and ensure that it can be loaded with numpy's `loadtxt` function. Your file should be identical to file 'Temp_global-mean-monthly.csv' provided in the course material:

```
1881,1882,1883,1884,1885,1886,1887,1888,1889,1890,...
-.19,.17,-.28,-.12,-.58,-.42,-.70,-.33,-.07,-.41,...
-.13,.15,-.36,-.07,-.32,-.49,-.55,-.35,.18,-.45,...
.04,.05,-.12,-.35,-.25,-.42,-.34,-.40,.07,-.39,...
.06,-.16,-.17,-.39,-.41,-.27,-.33,-.19,.10,-.29,...
.07,-.14,-.17,-.34,-.44,-.23,-.29,-.21,.00,-.39,...
-.18,-.22,-.07,-.35,-.42,-.33,-.23,-.16,-.09,-.24,...
.01,-.15,-.06,-.31,-.32,-.17,-.24,-.09,-.07,-.27,...
-.02,-.06,-.13,-.26,-.29,-.29,-.34,-.14,-.19,-.38,...
```

```
-.14,-.14,-.20,-.26,-.27,-.23,-.24,-.11,-.23,-.36,...
-.21,-.23,-.10,-.24,-.22,-.26,-.34,.03,-.25,-.24,...
-.18,-.15,-.22,-.32,-.22,-.26,-.25,.04,-.32,-.43,...
-.06,-.35,-.10,-.30,-.09,-.24,-.32,-.03,-.28,-.30,...
```

Hint: Use Python's `csv` reader/writer for this task.

4. Climate Data / Regression (5P)

1. Repeat the linear regression analysis shown in the lecture to fit a *line* to the observed temperature data, but this time, fit the line to data from every odd month (January, March, ...), and then compute the coefficient of determination using even months (Febrary, April, ...). Hint: First think about how you can integrate the monthly data. Do not use any aggregator such as mean, min max, etc. (2P)
2. Write a function that, given some (i) X and (ii) Y_0 coordinates for training and (iii) Y_1 coordinates for testing and (iv) a collection of degrees, fits for each degree a polynomial to the X/Y_0 data and returns the degree with the best fit to the X/Y_1 data set as measured by the coefficient of determination R^2. Then apply the function to the temperature data for each month, except for December, which you should use as testing data set. Use the range degrees from 1 to 6 in all your 11 comparisons. Finally, visualize the results using a scatter or line plot. (3P)

2. Visualizing demographic data (7P)

The course material contains file '12111-04-01-4-B_processed.tsv' which contains demographic data of Germany from the most recent census that was carried out in 2011. The original data is available at https://www.regionalstatistik.de/genesis/online/data?operation=statistic&levelindex=0&levelid=1589541151793&code=12111.

The table is organized as follows:

- Columns: federal state (in number code), age, population (male), population (female).
- Age '100' summarizes the population that is 100 years or older.

Visualize age distribution in this data set with `matplotlib` using three different plots. **Decorate all your plots with title, legend, axis labels, or other features that are necessary to interpret the visualization.**

1. Histogram showing the total age distribution of both the male and female population. Details on visualizing multiple data sets in one histogram can be found here https://matplotlib.org/3.1.1/gallery/statistics/histogram_multihist.html (2P)
2. Whisker plots of all 16 federal states showing the total age distribution. You can find a mapping between the number codes and the names of the federal states in file 'federal_states.tsv' (2P)
3. Pie chart showing the age distribution of the total population as percentages over age ranges $(0, 20]$, $(20, 40]$, $(40, 60]$, $(60, 80)$, and $(100, \ldots$ (2P)
4. Integrate all plots into one figure using `matplotlib`'s `subplot` function. Draw all plots next to each other. Alter the figure size (details see lecture) if necessary. (1P).