Introduction to mathematical modeling
Week 3
Tutorial: python, parameter estimation
Homework 3: write a report with the results of this tutorial[1]

Exercise 1: Population growth

In this tutorial you will estimate parameters of a population growth model based on US population between 1790 and 2010.

Consider the logistic growth model and the US census data from week1, Table 1.

Step 1: plot the data.

- Open a new jupyter notebook. (Inside a terminal, type 'jupyter notebook'. Then in the browser window that opens, select 'New->Python3'.). To write a header line or any comments, you can use the commenting symbol #.
- import the numpy and matplotlib.pyplot modules:

```
import numpy as np
import matplotlib.pyplot as plt
```

- Define numpy arrays, one containing the years and one containing the population in millions:

```
#time in years
t   = np.asarray([1790, 1810, 1830, 1850, 1870, 1890, 1910, 1930,
1950, 1970, 1990, 2010])

#population in million
pop = np.asarray([3.93, 7.24, 12.87, 23.19, 39.82, 62.95, 91.97,
122.78, 150.70, 208.0, 248.14, 308.19])
```

- Open a figure, plot the data with the time in years on the x axis and the population in millions on the y axis.

```
#Plot the data
figure()
plt.plot(t, pop, 's')
plt.xlabel('Year')
plt.ylabel('population (million)')
plt.show()
```

---

[1] You should hand in your executable jupyter notebook electronically as part of the report. Additionally, hand in a written report that reports all your findings and supporting calculations. It is possible, using markdown text, to write the entire report in the jupyter notebook as well.

Step 2: can the data be reasonably approximated by a logistic growth model? If so, what are the values of the parameters γ and K? Estimate them from the ODE.

- Consider the ODE given in (1.5). Write down how the ODE can be transformed so that it has the form of a linear model (3.2). You will find that you have to introduce a 'proportional growth rate' P(t) as the ratio of dN/dt over N. Rewrite equation (1.5) as a linear model, where the proportional growth rate is a linear function of the population.
- Estimate the values of the derivative dN/dt from the census data. For that, use symmetric differences, i.e. dN/dt in year 1850 will be approximated by: (Pop(1870) – Pop(1830)) /(1870-1830). Using this approximation, we will only have estimates for the years 1810 until 1990. *Remark: If you haven't seen this before, review the rules for indexing of numpy arrays. pop[2:] accesses all but the first two elements of the array 'pop'. pop[:-2] accesses all but the last two elements.*

```
# Estimate the time derivative of the population
dN = (pop[2:]-pop[:-2])/(t[2:]-t[:-2]);
```

- Plot the computed proportional growth rate as a function of the population. *Remark: Use markers for the visualization (plt.plot(pop,P,'*')).*
- If the data follows a linear relationship, then the logistic model is a reasonable model. Does the data follow a linear relationship? On which portion of the data?
- Isolate the data (proportional growth rate and population) that follow the linear relationship (i.e until year 1930).
- Calculate the least square estimators for the slope and the intercept of the linear model. Start by defining the matrix X and the vector Y to be considered in the notation of equation (3.11).
  Remark: To calculate the least square estimator in python, use numpy's lstsq function (alpha = np.linalg.lstsq(X,Y)[0]). google 'numpy lstsq' for more information.

```
# Data matrix
X = np.asarray([np.ones(7), pop[1:-4]]).transpose()
```

- What are the LSE of the parameters γ and K for the census data until 1930?
- Consider the solution to the ODE given in equation (1.6). Compute the solution for the estimated parameters γ and K for the time period 1790-2010. Notice that the time is considered as starting at year 1790 (i.e. N0=N(1790)=3.93 millions). Plot the solution over the population data from the first figure.

```
# compute the model estimate for the population development
t_estimate = np.arange(1790, 2030, 5)
N0=pop[0];
t_model=t_estimate-1790;
N=K*(N0/K)*np.exp(gamma*t_model)/(1+(N0/
K)*(np.exp(gamma*t_model)-1));
```

```
figure()
plt.plot(t, pop, 's')
plt.plot(t_estimate, N, 'k')
plt.xlabel('Year')
plt.ylabel('population (million)')
plt.show()
```

Interpret your final parameter values in terms of early growth of the U.S. population and ultimate size of the population. Are these numbers realistic? Why or why not?

Step 3: Estimate the growth rate from the analytical solution of the ODE.

- Consider the solution of the logistic growth model given by equation (1.6).
- Given the population capacity K estimated in step 2, we want to compute the growth rate as the LSE of a linear problem defined based on equation (1.6). For that we need to write the problem as a linear problem. Consider the following change of variables: .

$$\widetilde{N}(t) = N(t)/K, \quad Y(t) = \ln\left(\frac{\widetilde{N}(t)}{1-\widetilde{N}(t)}\right)$$

You should find that Y is a linear function of time with slope γ. What is the intercept?

```
#change of variables
popt=pop/K
Y=np.log(popt/(1-popt))
```

- Plot the new variable Y as a function of time for the census data. Use markers in the plot. Do you obtain a linear relationship? For which period?
- Compute the LSE for the linear problem. What value of the slope γ do you obtain? Compare the results of both models with the data in a single plot.

Exercise 2a: European countries by area

Include **either** the results of exercise 2a **or** exercise 2b in your report!

Step 1: acquire and plot the data.

- search online (e.g. wikipedia) for a 'list of european countries sorted by area'. Define two column vectors '`rank`' and '`area`' containing the positions of the european countries in the ranking (largest countries first, smallest countries last) and their areas, respectively.
- Plot the data. You should find that a model of the form area = α*exp(β*rank) suggests itself (hint: the command `semilogy(x,y)` in the matplotlib.pyplot module plots the logarithm of the vector y against the vector x). Is this a good model for all the data, or only a portion of it?

Step 2: Linear regression.

- how do you have to transform the model above to obtain a linear model?
- compute the LSE and MLE of the parameters (α,β).
- check for outliers in the data. What is the effect on the LSE and MLE of (α,β) if you remove them?

Exercise 2b: US census data by country

Step 1: acquire and plot the data.

- in the online resources, download 'pop_change.csv' and 'census.ipynb'. The notebook explains how to load the census data for the 52 US states, located in pop_change.csv, into a numpy matrix M. You can also retrieve the names of the states in each row. Visualize the population growth for each state in a single plot, and visualize the total population growth across all states.

Step 2: Linear regression.

- using the techniques from exercise 1 step 2, estimate the parameters γ and K separately for each state and for the whole US (in order to do so, you need to compute the 'proportional growth rate' P(t) for each country, i.e. for each row of M).

Some useful python links

- matplotlib tutorial for plotting: http://nbviewer.jupyter.org/github/jrjohansson/scientific-python-lectures/blob/master/Lecture-4-Matplotlib.ipynb
- numpy tutorial: https://docs.scipy.org/doc/numpy-dev/user/quickstart.html
- 10 minutes to understanding pandas (this might help you to understand what the notebook in exercise 2b is actually doing): https://pandas.pydata.org/pandas-docs/stable/10min.html

Tips for using data

- never modify the original data! Always work on a copy.
- first inspect, then model: Try to get a feeling for the data first before formulating a model. Plot the data. Try a few different ways. Look for outliers, missing values, NaNs etc. Compute some elementary statistics (mean, variance)
- discrete data (i.e. a certain number of measurements) is best visualised with markers: `plot(data_x, data_y, 's')`
- only plot lines if the intermediate values make sense (e.g. the continuous solution of a differential equation)