TSKS37 Lab 1
Plot assignment

Fall 2024
Version of this document: January 20, 2025

1 Getting started

This part of the assignment is here to give an introduction to data visualization. In Section 2 we give a tutorial on how to create plots that show the given data in a good way. As a second part, Section 3 focuses on creating a good looking histogram. Finally, you will in Section 4 include these figure in a given LaTeX lab report template.

Note that there are details regarding the data given in `lab1.ipynb`, while this document is more general.

Note on how to display the plots

Creating plots in a regular python script file the plot only appear after actively summoning the figure, the most common ways are:

- Insert `plt.ion()` before the plots to turn on automatic updates. If you are working with large plots this may slow down the program.
- Call `plt.show()` to summon the figure. This blocks further execution until the window is closed.
- Call `plt.draw()` to summon the figure. This command in non-blocking, thus the execution of the program continues.
- Call `plt.savefig(file_name_with_suffix)` to save the figure to file. Note: You can do this without first displaying the figure, as described in the bullets above.

If you are using Jupyter it automatically makes makes function calls to show the plot in the cell output!

2 Prey and Predator

2.1 Background

The interaction between predators and prey is a classic research topic in ecology. The basic hypothesis is that if there are plenty of prey (e.g., moose), then there will be plenty of food for the predators (e.g., wolves) which will grow in numbers. But as the number of predators grows larger, these will kill more and more prey, thus it makes sense that the number of prey will eventually begin to reduce. When the number of prey is too small to feed all the predators, then the predators will also reduce in numbers. Then the number of prey will increase again since there are not so many natural enemies left. And so the cycle starts all over again, at least according to the basic theory.

2.2 Description of the data set

In this exercise, we will study some real data from the national park of Isle Royale in USA: `http://www.isleroyalewolf.org`. The data set contains the number of moose and wolves that lived on the island in different years. The data set is available in the file `ecology.npy` that can be downloaded from Lisam.

We give a description in the jupyter notebook on how to load the data into python.

The data is organized as follows:

- The first column describes which year.
- The second column describes the number of wolves (predator).
- The third column describes the number of moose (prey).

2.3 Visualization

The purpose of this exercise is to investigate the data set by plotting the data in different ways. We want you to create these three plots:

1. One where you plots both lines in the same plots.
2. One where you plots both lines in the same plots using two different y-axis in the same plot.

3. One where you use subplots to show each species.

Before staring we describe the difference between a `matplotlib` `Figure` and `Axes` object.

- A `Figure` can be thought of as the window in which one (or more) plots exits, i.e. the top level container for visualization. A `Figure` has properties such as size, which control the scale of objects placed in the figure. When not using jupyter you must call
- An `Axes` is the region of the plot where data is actually plotted. You can plots many things in an `Axes` object, both 2D and 3D plots. The `Axes` is important since its used to set everything from labels (for the plot or axis) to creating the legend Each `Figure` can contain one (or more) `Axes`.
- To actually create a plot, such as line or bar graphs, you call the corresponding function on the `Axes` object. Consider the following snippet based on matplotlibs documentation:

```
# It is common to name it plt for easy access.
import matplotlib.pyplot as plt
import numpy as np

t = np.arange(0.0, 2.0, 0.01)
s = np.sin(2 * np.pi * t)

# Here we get a Figure object and an Axes object
fig, ax = plt.subplots()
# Plot using Axes object
ax.plot(t, s)

# Set title and axis labels
ax.set_title('An oscillating voltage')
ax.set_xlabel('time (s)')
ax.set_ylabel('voltage (mV)')
ax.grid()

# Save the figure our to a file
fig.savefig("test.png")
# Show a graphical interface (automatically handled in jupyter!)
plt.show()
```

Follow the following steps:

1. Plot 1: Plot the number of wolves and moose as a function of time. Everything in

the same figure. Use `set_xlabel()`, `set_ylabel()`, `set_title()`, and `legend()` to describe what is shown in the figure. Use `set_xlim()` to change the interval of the horizontal axis, so that it matches exactly the data.
What could be improved in this plot?

2. Plot 2: As you probably have noticed, there are two orders of magnitude (100 times) more moose than wolves, which means that it is hard to see the exact numbers of wolves in the previous figure. One way to solve this is to have two different scales in the same figure. Use the function `.twinx()` on an `Axes` object to get seconds `Axes` object, *i.e.*, `twin_ax = ax.twinx()`, that share xaxis but have a different y axis.

   Copy your the code from the first plot to the next cell in the notebook and modify such that it uses two y axis, and remember to including labels for both y axis.

   - Creating legends for multi axis plots is hard so you do not need to put legend in this figure.
   - Color can be applied to many things in `matplotlib`, and we want you to color the y labels such that it matches the plot line so it can understand which axis corresponds to which data.

3. Plot 3: The figure will plot the data series in two subfigure, instead of a single. Use the command `plt.subplots(...)` to create multiple plots (`Axes` objects) in one figure, for example, underneath each other. Make sure to explain the figure in the same way as before.

   It is good practice to always include `layout="constrained"` when creating a figure, this will fix problems such as overlapping elements between subplots.

4. What is the largest and smallest numbers of moose in the data set? Which year did these occur? Compute this using what you previously learned using suitable numpy functions! Mark these points with stars in one of your previous figures (you can use plot to do this). Then you do the same thing for the wolves and mark the values with rings. If there are multiple points on a curve, mark only the first one.

5. Think about how your figures would look like in a printout. There are two important things to keep in mind:

   (a) Use "strong" colors, such as black, red, and blue. Green and particularly yellow is often hard to see in printouts.

   (b) Make sure that it is possible to separate the curves also in a **gray scale** printouts. This requires different line styles (e.g., solid, dotted, dashdotted and dashed). You can use plot to select color and line style.

Go through all figures that you created with plot and make sure that the curves are easy to distinguish.

To pass this exercise you need to show and demonstrate the code used to generate the plots. Leave written comments in the code and make sure that the plots are self-explaining. As mentioned in the beginning of Section 1, you will insert the plots into a LaTeX lab report template in Section 4.

3 Distribution of data rates in a wireless network

3.1 Background

According to the study Swedes and Internet 2019 (Stiftelsen för Internet infrastruktur), 99% of all Swedes above 12 years old has a cellular phone (mobile phone) and 92% has a smartphone. The more we use our cell phones, the higher our expectation become at being continuously connected to Internet. It begins to be necessary to have a smartphone and cellular coverage to take care of daily tasks, such as paying for parking.

Cellular telephony is based on having so-called base stations deployed on roof tops in cities and high masts in the country side. Each cell phone communicates wirelessly with the closest base station. This means that radio waves (i.e., light with a wavelength that our eyes cannot see) are sent back and forth between the phone and the base station. The physics of wave propagation tells us that the energy is spread more and more the further the radio waves propagate – it's like blowing up a balloon that first is small and thick, and then becomes large and thin. In other words, the signals get weaker and weaker the further it is between the phone and the base station. When the signals become too weak, it is no longer possible to communicate; this is known as not having cellular coverage.

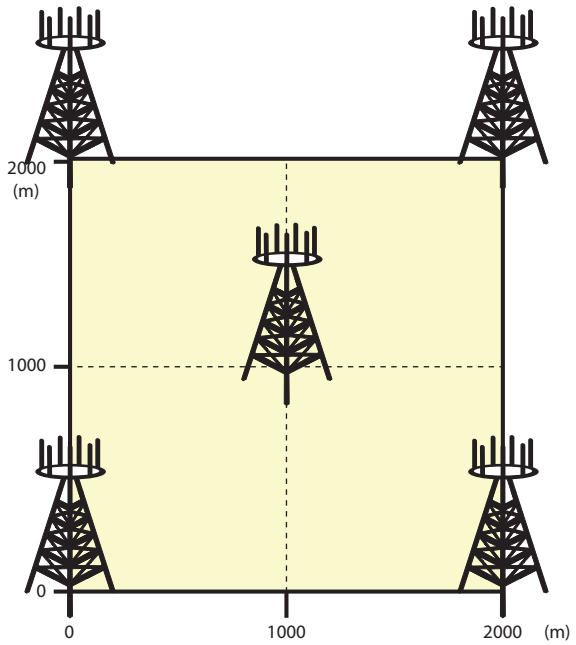

Figure 1: Example deployment of five base stations.

3.2 Theory

In order to compute the cellular coverage at different places, we first need to define what coverage actually means. We will do this in terms of the data rate, which is the number of information bits (zeros and ones) that can be transferred per second. If the data rate is larger than zero, then we have coverage. For simplicity, we assume a two-dimensional world; that is, we do not consider that the base stations and cell phones can have different elevation. We can then describe a position with two coordinates, (x, y), just as on a conventional map. If a cell phone is located at (x_0, y_0) and a base station is deployed at the position (x_1, y_1), then we can compute the distance by the classic distance formula

$$d = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \tag{1}$$

which follows from Pythagoras' theorem. If we let x_0, x_1, y_0, y_1 be measured in meter, then the distance d will also be in meter. Suppose that the cell phone or base station send signals with power P (in Watt). The signals will decay with the distance d. A simple model is that the power decays as $P/(1+d)^\kappa$. In other words, the power is P at $d = 0$ (no distance) and decays as P/d^κ when the distance d is much larger than one meter. The exponent κ is equal to 2 if there are no disturbing objects (e.g., buildings and asphalt) in the area, while measurements of κ in urban areas give values in the interval 3-6.

The distant-dependent loss of signal power is a problem since the received signal power should be compared to the noise power at the receiver. Let σ^2 denote the power of the noise power (in Watt). The signal-to-noise ratio is defined as

$$\frac{P}{P(1+d)^\kappa \sigma^2} \tag{2}$$

in our model. This expression should preferably be much larger than 1, since then the received signal power $P/(1+d)^\kappa$ is much larger than the noise power σ^2.

Based on the signal-to-noise ratio, one can compute the data rate as

$$R = \begin{cases} 0, & \frac{P}{P(1+d)^\kappa \sigma^2} < 0.3 \\ B \log_2 \left(1 + \frac{P}{P(1+d)^\kappa \sigma^2}\right), & 0.3 \leq \frac{P}{P(1+d)^\kappa \sigma^2} \leq 63 \\ B \log_2 (1+63), & \frac{P}{P(1+d)^\kappa \sigma^2} > 63 \end{cases} \quad \text{[bit/s]} \tag{3}$$

where B is the bandwidth (in Hertz) and the logarithm of "one plus the signal-to-noise ratio" gives the number of information bits than can be transferred per second per Hertz of bandwidth. We do not expect you to understand where this formula comes from (you will have the possibility to learn this in later courses!), but only to apply it within this project. It can be seen from (3) that the data rate is zero if the signal-to-noise ratio is lower than 0.3.

This number originates from a 4G standard for mobil telephony and says that the received signal power cannot be too small. This is our definition of not having coverage: the data rate is zero. The middle formula in (3) gives larger values as the signal-to-noise ratio increases, but when it grows beyond 63 the rate will not continue to grow any more. This is a type of hardware limitation that originates from real systems.

To get practically reasonable results, you can use the parameter values $P = 1$ Watt, $B = 10$ MHz, exponent $\kappa = 4$, and noise power $\sigma^2 = 10^{-11,2}$ Watt.

3.3 Assignment

In this task you will present statistics and plots based on data calculated using the theory presented above. We will not use any analytical expressions to find the statistics, but rely on numerical methods to make approximations. In this case, the approximation is created by placing **many** users randomly in the area and evaluating the data rate for each user. Note that every user we add improves the approximation at the cost of needing more computations.

The setup is shown in Figure 1, where you have a 5 base stations deployed in a 2000 by 2000 area. We provide you with the code to generate the data rates for all users. But do take a look to make sure you understand what is happening in the code! Using the data you will:

1. Calculate the probability of having coverage.
2. For the user with coverage, calculate their average data rate.
3. Make a histogram showing the distribution of data rates. To ensure that it communicates the underlying data efficiently. You are allowed to change `number_of_users`, but do keep in mind that increasing it increases computational complexity of the program.

Remember to use what you learned in the previous assignment and format your plot.

4 Presentation of the created plots

You will now present two of plot in a small LaTeX lab report, for which a template is provided in `lab-report.tex`. LaTeX is a professional, and commonly used, typesetting engine in which one compiles LaTeX source code/text to a PDF. You have most likely encountered many document that are created using LaTeX (*e.g.* this document!), because it is a preferred tool for writing math.

You will now:

- Setup your latex environment.

  We recommend using `https://overleaf.com` because it provides a good environment for writing LaTeX. LiU has a institutional license, so make sure to select **Log in with SSO** and enter your liu email address. One can compile LaTeX locally with some know-how.

- Enter a good title and your names into `\title{...}` and `\author{...}`.

- Choose either plot 2 or plot 3 from Section 2 and save it in the following four common image formats: `.jpg`, `.png`, `.pdf`, and `.eps`. Insert the four plots in the figure corresponding to each file format. (Find the `\includegraphics` and replace "enter a path here" by the path, *i.e.*, name of the file, to the corresponding image after you have uploaded it to Overleaf.)

- Make sure that the inserted figures have clearly visible and readable elements (including: labels, titles, legend, and axis ticks.). We also expect you to convert the units to a suitable prefix when applicable.

  Scaling the image in LaTeX is not the best option here, but rather change the figures size in `matplotlib` which gives a more desirable results. Change the size of the figure in `matplotlib` and see how this impacts the readability. This is easiest when you create the figure, i.e. when you call `plt.figure(figsize(width=...,height=...))` (the unit is inches).

  What happens with the image in LaTeX when you change the size of the figure in `matplotlib`?

- In LaTeX, write your conclusion about which format(s) are best and why.

- Insert the plot from Section 3 using the file format you found is the best.

- Put the coverage percentage and average data rate you computed into the document, where you find it fits.