Assignment 2 – Storage, Hydro, and Transmission

Due Friday, 9 December 2022 [100 points]

Instructions and Rules

- Submission on ISIS requires two-factor authentication.
- Justified exceptions may be granted. Contact f.neumann@tu-berlin.de.
- Submission must include both written answers and code that shows how answers were obtained. All submitted material will be factored in for the grading.
  - **Option A**: combined answers and code in `.ipynb` file + `.html` export of notebook
  - **Option B**: `.pdf` with written answers (incl. figures) + `.py`/`.ipynb` with code and comments
- Submissions must be your own work, plagiarism from the web or your peers will be sanctioned!
- Always clearly mark which task and subtask you are working on.
- Always provide units for quantities (e.g. energy, power, emissions).
- It must be possible to run submitted code without manually setting variables or executing code cells multiple times to retrieve all results (exception: local file paths)!
- You may use additional Python packages as long as they are available via `pip` or `conda`.

Task 1: Dimensioning Lossless Storage [30 points]

Required Tools: `pandas`, `matplotlib`, `numpy`

Reconsider the time series on wind and solar capacity factors and electricity demand in Germany in 2015 from the first assignment:

File: `https://tubcloud.tu-berlin.de/s/ppRkB2mwsKkJrRm/download/time-series-lecture-2.csv`

The goal of this task is to dimension a lossless storage for different generation mixes of wind and solar to supply demand. I.e. assume, for simplicity, that the storing electricity does not incur any losses!

[1 point] (a) Load the provided dataset as a `pandas.DataFrame`.

[2 points] (b) Rescale the wind and solar capacity factor time series such in total as much energy is produced as is consumed according to the column “load [GW]”.

[1 point] (c) Store the two resulting new time series under the names “onwind [GW]” and “solar [GW]” respectively.

[2 points] (d) Determine two mismatch / residual load time series: one for the case where all power is supplied by solar, another for the case where all power is supplied by wind generation.

[2 points] (e) Plot the two mismatch time series for January in a single figure. Label the graph appropriately. Briefly describe what differences you notice.

[2 points] (f) For each of the two mismatch time series, determine the peak and total *absolute* mismatch.

[3 points] (g) For each of the two mismatch time series, calculate the state of charge profiles (i.e. time series) of the smallest possible storage that could align supply and demand at all times. Briefly explain the shape of the state of charge profiles.

[3 points] (h) What are the required energy [GWh], charge [GW], discharge [GW] capacities for a purely wind-based and a purely solar-based system? Briefly describe how you identify these capacities.

Until now, we only looked at two extreme cases (100% wind or 100% solar). In the following, we are going to look at mixes with both wind and solar generation.

[2 points] (i) Write a function that computes the residual load as a function of the wind generation share α with a corresponding solar generation share of $1-\alpha$.

[2 points] (j) Apply this function for steps of 5% between 0% and 100% for α and store the residual load time series in a `pandas.DataFrame` where each column denotes an α.

[1 point] (k) Plot the required storage charge and discharge capacities as a function of the wind generation share α. Choose appropriate labels!

[1 point] (l) Plot the total amount of energy that needs to be stored in the whole year as a function of the wind generation share α. Choose appropriate labels!

[1 point] (m) Plot the required energy capacity of a storage as a function of the wind generation share α. Choose appropriate labels!

[1 point] (n) Which of the α computed results in the lowest energy capacity required for the storage?

[3 points] (o) Plot the state of charge profile for this α and compare it to the cases with 100% wind or solar.

[3 points] (p) You are given a choice between two storage technologies with investment costs as outlined in the table below. Which storage technology would you choose? Include a cost comparison of both technologies.

Investment Cost	Unit	Storage Technology A	Storage Technology B
Energy Capacity	[€/kWh]	200	10
Charge Capacity	[€/kW]	50	500
Discharge Capacity	[€/kW]	50	500

Task 2: Hydropower in the World [18 points]

Required Tools: `pandas`, `matplotlib`, `cartopy`, `geopandas`

In this task, we are going to look at the *Global Power Plant Database* published by the *World Resources Institute* with a focus on the World's hydropower plants:

Source: https://datasets.wri.org/dataset/globalpowerplantdatabase

File: https://tubcloud.tu-berlin.de/s/aZZfWGd8rmbEx39/download/global_power_plant_database.csv

[2 points] (a) Read the provided CSV file with pandas and convert it into a `geopandas.GeoDataFrame`.

[1 point] (b) Reduce the dataset such that it only includes hydropower plants.

[1 point] (c) Identify the oldest and newest hydropower plant in the dataset.

[1 point] (d) Determine the share of hydropower plants with capacities below 10 MW and below 100 MW.

[2 points] (e) Identify the 3 countries with the largest hydropower capacities and list their total capacities in units of GW.

[2 points] (f) Identify the 5 hydropower plants with the highest capacity factor based on the estimated generation in 2017. List their country, name, commissioning year, capacity and estimated generation in 2017.

[2 points] (g) Create a global map with all hydropower plants. The marker size should be proportional to the nominal capacity (appropriately scaled) and coloured according to the estimated 2017 capacity factor. Add coastlines and country borders for orientation.

[1 point] (h) Remove all hydropower plants with a missing commissioning year from the dataset. What share of hydropower plants is affected by this measure?

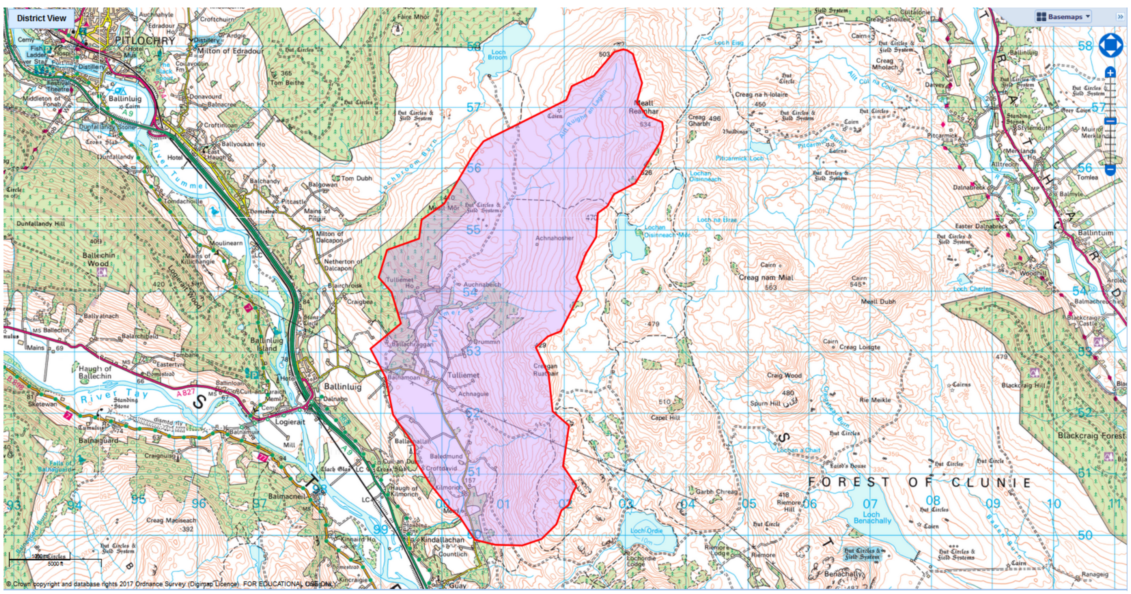
[2 points] (i) Compute the capacity-weighted mean age of all remaining hydropower plants.

[4 points] (j) For each country, compute the average age of its hydropower plants weighted by the plants' average estimated generation between 2014 and 2017. Hint: You can iterate over `GroupBy` objects with a loop. Which five countries have the oldest and newest fleet of hydropower plants?

Task 3: Designing a Run-of-River Power Plant [25 points]

Required Tools: `pandas`, `matplotlib`, `numpy`

You have been tasked with designing a small run-of-river power plant near Pitlochry in Scotland at the Tulliemet river with a catchment area of 20 km^2.



Catchment area of the desired run-of-river plant.

For the evaluation of the site, you have been given a time series of daily water flow rates (in m^3/s) from 1983 to 2014:

File: https://tubcloud.tu-berlin.de/s/af5G99i35dwNPCz/download/water-flows.csv

The project lead has also already identified a suitable location for the weir, which would result in a net head of 60 metres. For the following calculations, assume a generator efficiency of 85% and that the weir cannot store significant amounts of water inflow across days.

[1 point] (a) Read the provided dataset as a `pandas.Series` with parsed dates.

[2 points] (b) Plot the monthly mean flow rate across the whole time span. Use a figure size of 20 by 4 inches and apply appropriate labels. Describe the profile.

Not all of the water the river carries can be used to produce power. A compensation flow (or reserve flow) must remain in the river at all times. For the compensation flow assume the 5% quantile of flows (i.e. flow rate exceeded 95% of the time). Economically, it often does not make sense to dimension the penstock and generator for the peak flow rate. Initially, assume a design flow (i.e. the maximum flow the penstock and generator can handle) corresponding to the 70% quantile (i.e. flow rate exceeded 30% of the time).

[1 point] (c) What is the value for the compensation flow?

[1 point] (d) What is the value for the design flow?

[3 points] (e) Calculate a time series for the available flow for power production (i.e. deducted reserve flow and capped at design flow).

[3 points] (f) In one figure, plot a duration curve for the total flow and the available flow. Label the lines. The x-axis should indicate the percentage of time across the whole 30-year time span. Limit the y-axis to 2 m^3/s.

[1 point] (g) Determine the rated power of the run-of-river scheme using the design flow.

[2 points] (h) Compute a time series for the daily power production of the scheme based on the available flows.

[1 point] (i) What is the long-term average capacity factor of the run-of-river power plant?

[3 points] (j) Compute and plot the annual energy yields of the plant.

[2 points] (k) Compute and plot the relative deviation of the annual energy yields from the long-term average annual energy yield.

[5 points] (l) Plot the relationship between power rating and average annual energy yield based on design flows ranging between the 0% and 100% quantiles of flow in increments of 5%. Based on this plot, explain why it might make sense to limit the design flow.

Task 4: European Gas Transmission Network [27 points]

Required Tools: `pandas`, `matplotlib`, `cartopy`, `geopandas`, `networkx`

In the lectures and workshops, so far we exclusively studied the European electricity transmission network. In this task, you'll perform some network analysis on data about the European gas transmission network as provided by the SciGRID_gas project:

Source: https://www.gas.scigrid.de/posts/2020-Sep-02_iggielgn.html

Files:

https://tubcloud.tu-berlin.de/s/8SMwwWQyn6GiPez/download/scigrid-gas-nodes.geojson

https://tubcloud.tu-berlin.de/s/fF6KKpWtJyS3BmD/download/scigrid-gas-pipelines.geojson

[2 points] (a) Read the nodes and edges representing pipelines into two separate `GeoDataFrames`.

[3 points] (b) Plot the transmission network data on a map, such that the colors of the line indicate the capacity of the pipelines ("million cubic metres per day") and the width of the lines is proportional to the rated pipeline pressure. The figure should have a size of 15 by 10 inches, a valid projection, and a colorbar for the colours showing the pipe capacity. Add coastlines and country borders for orientation.

[2 points] (c) Which is the pipeline with the highest maximum pressure? Why might this high pressure be necessary?

[1 point] (d) What share of pipelines is bidirectional (i.e. they can transport gas in either direction)?

[3 points] (e) Identify the subset of pipelines which cross borders. Check your code visually by plotting only the crossborder pipelines on a similar map as before. What percentage of lines crosses borders?

Let's make use of some `networkx` functions to analyse the network graph.

[1 point] (f) Import `networkx` under the alias `nx`.

[3 points] (g) Create a `networkx.Graph` from the pipeline data using the function `nx.from_pandas_edgelist()`. As attributes include the length, diameter, capacity and pressure of the pipeline.

[1 point] (h) Determine the number of nodes and edges.

[2 points] (i) Determine the number of connected components (i.e. how many disconnected network parts there are). Also evaluate the number of nodes of each component. How would you interpret these numbers?

[2 points] (j) Compute the average degree of the network. Briefly describe what this number means.

[2 points] (k) Analyse the distribution of degrees. What do you notice?

[1 point] (l) Check whether the network is planar.

[4 points] (m) Using adjacency and degree matrix, compute the unweighted Laplacian of the network. How would you characterise the Laplacian? Suggest a scalar measure that describes this characteristic.