Assignment 1 – Renewables

Due Monday, 20 May 2024, 23:59 [100 points]

Instructions and Rules

- Submission on ISIS requires two-factor authentication. Justified exceptions can be granted.
- Submissions must include both written answers and code that shows how answers were obtained.
- Always clearly mark which task and subtask you are working on.
- Always provide units for quantities (e.g. energy, power, emissions) for your results.
- All plots must have appropriate axis labels with units if applicable.
- The quality of the presentation of your results will be factored into the grading.
- Partial points will be deducted for any missing labels or units.
- Submissions must be own work, plagiarism from the web or peers will be checked and sanctioned!
- Please upload a single `.ipynb` file and the corresponding `.html` export of the notebook.
- Do not upload data!
- 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: Conversion of Wind Speeds to Power Output of a Wind Turbine [19 points]

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

You have been tasked with picking a wind turbine model for a wind park project near the town of Cottbus. To execute this task, you have been given a dataset with time series of wind speeds at the site under consideration at ten meters above the ground in units of m/s. Moreover, your client has already narrowed down the choice of turbines to two models from two different manufacturers. The power curves of the two turbine types that are being considered are shown in the table below. Both wind turbine choices have a hub height of 90 meters and a nominal rating of 3.5 MW.

Wind speed time series:

https://tubcloud.tu-berlin.de/s/X8WGZxA6554iSz9/download/wind-speeds.csv

Power curves:

Wind Speed [m/s]	Vestas [MW]	Enercon [MW]
<4	0	0
4-6	0.2	0.4
6-8	0.5	0.8
8-10	1.3	1.5
10-12	2.5	2.1
12-14	3	2.4
14-16	3	2.9
16-	3	3

The cut-out speed beyond which the wind turbine is shut down is 22 m/s for the Vestas model and 25 m/s for the Enercon model.

[1 point] (a) Import pandas under the alias pd, numpy under the alisa np and matplotlib like in the tutorials.

[1 point] (b) Read the CSV file given under the link above into pandas such that the timestamps given in the first column become the indices. *Hint:* Use the keyword arguments `index_col=` and `parse_dates=` to ensure the timestamps are treated as `pd.DatetimeIndex`.

[1 point] (c) Demonstate with code that the index is indeed treated as `pd.DatetimeIndex`.

[2 points] (d) Convert the wind speeds to a hub height of 90 meters using log law assuming open flat grassland as terrain. Print the conversion factor as well as the converted wind speeds of the first ten hours of the year.

[3 points] (e) Write a Python function for each wind turbine that takes the wind speed as an argument and converts a given value into power production using the respective turbine's power curve.

[1 point] (f) Use this function convert from wind speeds to power production for the full year for each of the two turbine types. Show the last 20 hours of electricity generation for each wind turbine.

[2 points] (g) Which turbine would yield the most energy output in the year? By what relative margin (in %)?

Continue all subsequent calculations with the turbine yielding the highest electricity output per year:

[1 point] (h) Normalise the power output time series by the turbine's rated capacity.

[1 point] (i) What is the annual average capacity factor of the wind turbine?

[1 point] (j) What are the monthly average capacity factors?

[1 point] (k) What share of time does the wind turbine produce no power?

[1 point] (l) What share of time does the wind turbine produce power at rated capacity?

[1 point] (m) Does the wind turbine ever shut down because of too high wind speeds?

[2 points] (n) Plot the capacity factor time series for the full year using `matplotlib`. Include axis labels!

Task 2: Time Series Analysis [22 points]

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

In this task, you will reproduce the graphics presented in the second lecture on wind, solar, demand and price time series.

The CSV file needed is available at:

https://tubcloud.tu-berlin.de/s/nwCrNLrtL6LAN3W/download/time-series-lecture-2.csv

It includes hourly time series for Germany in 2015

1. electricity demand from OPSD in GW
2. onshore wind capacity factors from renewables.ninja in per-unit of installed capacity
3. offshore wind capacity factors from renewables.ninja in per-unit of installed capacity
4. solar PV capacity factors from renewables.ninja in per-unit of installed capacity
5. electricity day-ahead spot market prices in €/MWh from EPEX Spot zone DE/AT/LU retrieved via SMARD platform

For the tasks listed below, all requested plots above must be created with `matplotlib`. For all plots, choose suitable figure sizes, axis labels and line styles. Partial points will be deducted for incomplete figures.

For each column of the DataFrame:

[2 points] (a) What are the means, minimums, maximums? Which column has the highest standard deviation?

[2 points] (b) Find and list the timestamps where data is missing.

[3 points] (c) Fill up the missing data with the values observed 6 days ahead in the DataFrame. I.e. for missing data on 1 January 12:00, look for the values on 7 January 12:00. Check that there are no more missing values in the dataframe.

[2 points] (d) Plot the hourly time series for the full year (in a separate plot for each column).

[2 points] (e) Plot the hourly time series for the month August (in a separate plot for each column).

[4 points] (f) Resample the time series to daily, weeky, and monthly frequencies and create a plot that contains the variously resampled time series in one figure.

[3 points] (g) Sort the values in descending order and plot the duration curve. Hint: Run `.reset_index(drop=True)` to drop the index after sorting. The x-axis should be given as a percentage of the year.

[4 points] (h) Perform a Fourier transform on the load time series and plot the resulting spectrum. Explain the cause of the distinct peaks.

Task 3: Solar Power in the World [24 points]

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

In this task, we are going to look at the *Global Solar Power Tracker* published by the *Global Energy Monitor* with a focus on the global utility-scale solar photovoltaic (PV) and solar thermal facilities:

Source:

https://globalenergymonitor.org/projects/global-solar-power-tracker/

File:

https://tubcloud.tu-berlin.de/s/rwygB8sbmzxSRk5/download/Global-Solar-Power-Tracker-May-2023.xlsx

[3 points] (a) Read the provided Excel file with pandas and convert it into a `geopandas.GeoDataFrame` with the appropriate coordinate reference system. *Hint:* The file contains multiple sheets and only one of them contains the data.

[2 points] (b) What are the shares of solar thermal and photovoltaics capacity in percent of the total capacity?

[2 points] (c) Plot the new global capacity installations per year (in GW) for the years 2000 to 2022.

[2 points] (d) Identify the oldest solar photovoltaics park(s) in the dataset by name, country, start year, status and capacity.

[3 points] (e) Identify the three largest operating solar parks in the world. List their names, countries, capacities, start years, state/province and technology type.

[2 points] (f) Identify the total capacity (in GW) per project status.

[4 points] (g) Create a global map *each* with all utility-scale solar parks (i) in operation and (ii) under (pre-)construction. The marker size should be proportional to the nominal capacity (appropriately scaled) and coloured according to the start year. Add coastlines and country borders for orientation. Include labels for the colouring and marker sizes. The colorbar extent should range from 2010-2022 for (i) and from 2022-2026 for (ii).

[2 points] (h) Remove all solar parks with a missing start year from the dataset. Compute the capacity-weighted mean age of all remaining solar parks in operation.

[4 points] (i) For each continent, compute the average age of its solar parks weighted by the parks' rated capacity. Hint: You can iterate over `GroupBy` objects with a loop. Which continent has the oldest and newest fleet of solar parks?

Task 4: European Industrial Sites [35 points]

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

In the CSV file below, you will find a dataset of georeferenced industrial sites of energy-intensive industry sectors in Europe.

`https://tubcloud.tu-berlin.de/s/JamLFf2Pjc6KHfp/download/Industrial_Database.csv`

This dataset was originally retrieved from the following website. Documentation on fields (e.g. descriptions and units of columns) can be found in the `datapackage.json` file there.

[2 points] (a) Read the CSV file as DataFrame. The separator in this dataset is a semicolon (";"). Use the first column as the index column.

[4 points] (b) When you expect the dataset, you will find that the geographic information is given in the "geom" column in a format similar to "POINT(x, y)". This format is called the Well-known text representation of geometry (or short: WKT). Such WKT strings can be used to convert the ordinary DataFrame into a GeoDataFrame using the function `gpd.GeoSeries.from_wkt()`. The points are given in the coordinate reference system EPSG:4326, which you should also specify when constructing the GeoDataFrame. After building the GeoDataFrame, set the column "SiteID" as its index.

[2 points] (c) For how many industrial sites is the data on its location missing?

[2 points] (d) Data on emissions is given from two sources, ETS (EU Emissions Trading System) and EPRTR (European Pollutant Release and Transfer Register). We want to use the ETS as our primary data source for emissions, but these values are missing for some industrial sites. For sites where ETS data is missing, but EPRTR data is available, use these to fill missing values. Finally, state the main difference after this step.

[4 points] (e) Visualise the data contained in the GeoDataFrame in a `matplotlib` plot such that the following criteria are fulfilled:

- figure size 13 by 13 inches
- axis projection is the *Albers Equal Area* projection
- circles are coloured by the industry sector
- a legend indicates which colour corresponds to which industry sector
- circle size is proportional to emissions and appropriately sized (a legend is not necessary)
- coastlines are shown in black and country borders in grey

Using filtering and grouping commands:

[2 points] (f) Create a `matplotlib` bar chart outlining the emissions per industry sector in units of $MtCO_2$/year. Sort the bars by size in ascending order. Make sure to label the axes appropriately.

[3 points] (g) Identify the two industry sectors with the highest total emissions. What's the share in percent of each of these sectors relative to the total emissions accounted for?

[2 points] (h) Identify the two countries with the highest ETS emissions in the chemical industry sector.

[2 points] (i) For each country, identify the sector with the highest level of emissions.

[3 points] (j) List the 10 Polish companies responsible for the most emissions and their respective emissions in $MtCO_2$/year in descending order. For each of the companies, list the sector they belong to.

In the following, we additionally want to aggregate data on industrial sites to various NUTS levels. The NUTS region data can be read-in directly in `geopandas` with the URL below:

`https://tubcloud.tu-berlin.de/s/5WJ6pGsBKR7a3sp/download/NUTS_RG_10M_2021_4326.geojson`

[5 points] (k) Plot a choropleth map of NUTS-2 regions with `matplotlib` which shows each region's industry emissions per area (in $tCO_2$2/year/km^2). Limit the colorbar to 1 kt/year/km^2 by passing the keyword argument `vmax=1000` to the `geopandas` plotting function. Focus the view on continental Europe by setting the appropriate bounds of the figure. Add a colorbar to the right-hand side of the plot.

[4 points] (l) Which Austrian NUTS-3 region has the highest level of industry emissions? In addition to the "NUTS_ID", also provide the common name of the region. Which companies are settled there and which sectors do they belong to?