Assignment 1 – Renewables

Due Friday, 18 November 2022, 23:00 [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: Conversion of Wind Speeds to Power Output of a Wind Turbine [15 points]

Required Tools: `pandas`, `matplotlib`

You have been tasked with picking a wind turbine model for development at a particular site. 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 narrowed down the choice of turbines to two models. The power curves of two wind turbine model that are being considered are shown in the table below. Both wind turbine choices have a hub height of 80 meters and a nominal rating of 3 MW.

Wind speed time series:

`https://tubcloud.tu-berlin.de/s/DbDZ7KCtoRAFpyS/download/wind-speeds.csv`

Power curves:

Wind Speed [m/s]	Vestas V112 3MW [MW]	Enercon E82 3000kW [MW]
<4	0	0
4-6	0.2	0.1
6-8	0.5	0.3
8-10	1.4	0.8
10-12	2.6	1.6
12-14	3	2.2
14-16	3	2.7
16-	3	3

The cut-out speed beyond which the wind turbine is shut down is 25 m/s for the Vestas model and 28 m/s for the Enercon model.

[1 point] (a) Import `pandas` under the alias `pd`, 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.

[1 point] (c) Convert the wind speeds to a hub height of 80 meters using power law with $\alpha = 1/7$.

[2 points] (d) Write a 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.

[2 points] (e) Use this function convert from wind speeds to power production for the full year for each of the two turbine types.

[1 point] (f) Which wind turbine would yield the most electrical energy in the year? By what margin?

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

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

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

[1 point] (i) What share of time does the wind turbine produce nothing?

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

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

[1 point] (l) Plot the capacity factor time series for the full year using `matplotlib`. Include axis labels!

[1 point] (m) Looking at the plot, what would you improve about the conversion of wind speeds to capacity factors?

Task 2: Time Series Analysis [27 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

Use the function `pd.read_csv` with the keyword arguments `index_col=` and `parse_dates=` to ensure the timestamps are treated as `pd.DatetimeIndex`.

For each column of the DataFrame:

[2 points] (a) What are the average, minimum and maximum values?

[1 point] (b) Find the timestamps where data on prices is missing.

[2 points] (c) Fill up the missing data with the prices observed one week ahead.

[2 points] (d) Plot the time series for the full year.

[2 points] (e) Plot the time series for the month May.

[3 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.

[2 points] (h) Plot a histogram of the time series values. Use at least 20 bins.

[5 points] (i) Perform a Fourier transformation of the time series. What are the dominant frequencies? How can they be explained?

All requested plots above must be created with `matplotlib`. For all plots, choose suitable figure sizes, axis labels and line styles.

Full points indicated are awared if task was completed correctly for all columns.

[5 points] (j) Calculate the Pearson correlation coefficients between all given time series and identify the time series with strongest (positive/negative) and weakest correlations. Briefly write down your thoughts about what may be the reason for these correlations.

Task 3: European Industrial Sites [32 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.

[3 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.

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

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

- figure size 10 by 10 inches
- axis projection is Plate Carree projection
- circles are coloured by the industry sector
- a legend indicates which colour corresponds to which industry sector
- circle size is proportional to ETS 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] (e) Create a `matplotlib` bar chart outlining the ETS emissions per industry sector in units of MtCO2/year. Make sure to label the axes appropriately

[2 points] (f) Identify the industry sector with the highest ETS emissions. What's the share in percent of this sector relative to the total emissions accounted for?

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

[3 points] (h) For each sector, identify the country with the highest level of ETS emissions.

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

[3 points] (j) List the 20 German companies responsible for the most ETS emissions and their respective emissions in MtCO2/year in descending order.

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`

[4 points] (k) Plot a choropleth map of NUTS-2 regions with `matplotlib` which shows each region's industry ETS emissions per area (in tCO2/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.

[3 points] (l) Which NUTS-3 region has the highest level of industry ETS 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?

Task 4: Land Eligibility for Wind Turbines [26 points]

Required Tools: `atlite`, `geopandas`, `matplotlib`, `rasterio` (to be introduced Nov 9, 2022)

Perform an eligibility analysis for the land area available for the development of wind parks in the Grand Duchy of Luxembourg.

The following areas are to be excluded or included:

1. exclusion of natural protection areas
(Natura2000 dataset: `https://tubcloud.tu-berlin.de/s/zPMqJFPD8tKq2Ss/download/Natura2000_end2021-LU.gpkg`)
2. exclusion of second-tier land protection areas
(CDDA dataset: `https://tubcloud.tu-berlin.de/s/fZfFMim8MYNTsgA/download/CDDA_2022_v01_public-LU.gpkg`)
3. exclusion of a radius of 5 km around airports
(NaturalEarth dataset: `https://tubcloud.tu-berlin.de/s/TbEJ9Lsy9EpTcQS/download/ne_10m_airports.gpkg`)
4. exclusion of a buffer of 300m next to major roads
(NaturalEarth dataset: `https://tubcloud.tu-berlin.de/s/zcKQ95TgpyMJb8E/download/ne_10m_roads.gpkg`)
5. exclusion of a distance of 1200m around the following CORINE land cover classes: 111, 112, 121, 122, 123, 124
(CORINE CLC 2018 dataset: `https://tubcloud.tu-berlin.de/s/z7aY8HNCdETQMT7/download/U2018_CLC2018_V2020_20u1-LU.tif`)

6. inclusion only of the following CORINE land cover classes, which are deemed suitable for constructing wind turbines: 211, 212, 213, 231, 241, 242, 243, 321, 323, 324, 333
(CORINE CLC 2018: `https://tubcloud.tu-berlin.de/s/z7aY8HNCdETQMT7/download/U2018_CLC2018_V2020_20u1-LU.tif`)

The geometric shapes of countries can be found under the following link:

`https://tubcloud.tu-berlin.de/s/7bpHrAkjMT3ADSr/download/country_shapes.geojson`

The CORINE land cover dataset is given in coordinate reference system EPSG:3035 and stores classes as numbers as listed on the following website:

`https://collections.sentinel-hub.com/corine-land-cover/readme.html`

[2 points] (a) Import the relevant packages needed to perform this analysis.

[2 points] (b) Translate the CORINE land classes listed above to code values as stored in the dataset using the table above. Also state the descriptive names of the classes. Distinguish between (a) classes to which a distance should be kept and (b) classes deemed eligible for wind development.

[2 points] (c) Load the country shapes as GeoDataFrame, reduce it to a single entry for Luxembourg, and reproject it to EPSG:3035.

[6 points] (d) Plot and calculate (in %) the area excluded by each of the exclusion constraints (1.-6.) individually using the `atlite.gis.ExclusionContainer` and `atlite.gis.shape_availability`. Refer to the `atlite` tutorial for information on how to plot the output of the latter function with `rasterio` and `matplotlib`.

[2 points] (e) Plot and calculate (in %) the area available for the development of wind parks considering all exclusion zones together?

[2 points] (f) Why does the sum of individual exclusion constraints (1.-6.) not add up to the total area excluded?

[6 points] (g) Perform a sensitivity analysis on the distance criterion to CORINE land classes by repeating the land eligibility calculation with distance requirements of 0m, 200m, 400m, 600m, 800m and 1000m. Plot the available area as a function of the distance requirements and describe the curve. Make sure to label your figure appropriately.

[4 points] (h) Assume that for the expansion of wind power in Luxembourg, a capacity density of 2 MW/km^2 of available land can be achieved and that the wind conditions result in an average capacity factor of 33%. What is the total energy in units of TWh that could be produced from wind power in Luxembourg each year if the distance criterion were reduced to 800m?