SIT220/731 2024.T3: Task 4P

Working with **pandas** Data Frames (Heterogeneous Data)

Last updated: 15th November 2024

Contents

1 Introduction

This task is related to Module 1, 2 and 3; see the *Learning Resources* on the unit site.

This task is due on **Week 11 (Sunday)**. Start tackling it as early as possible. If we find your first solution incomplete or otherwise incorrect, you will still be able to amend it based on the generous feedback we will give you. In case of any problems/questions, do hot hesitate to attend our on-campus/online classes or use the Discussion Board on the unit site.

Submitting after the aforementioned due date will incur a late penalty. This task is part of the **hurdle requirements** in this unit. Not submitting the correct version on time results in failing the unit.

All submissions will be checked for plagiarism. You are expected to work independently on your task solutions. Never share/show parts of solutions with/to anyone.

2 Question 1 - 4

Download the `nycflights13_weather.csv.gz` data file from our unit site (*Learning Resources → Data*). It gives the hourly meteorological data for three airports in New York: LGA, JFK, and EWR for the whole year of 2013. The columns are:

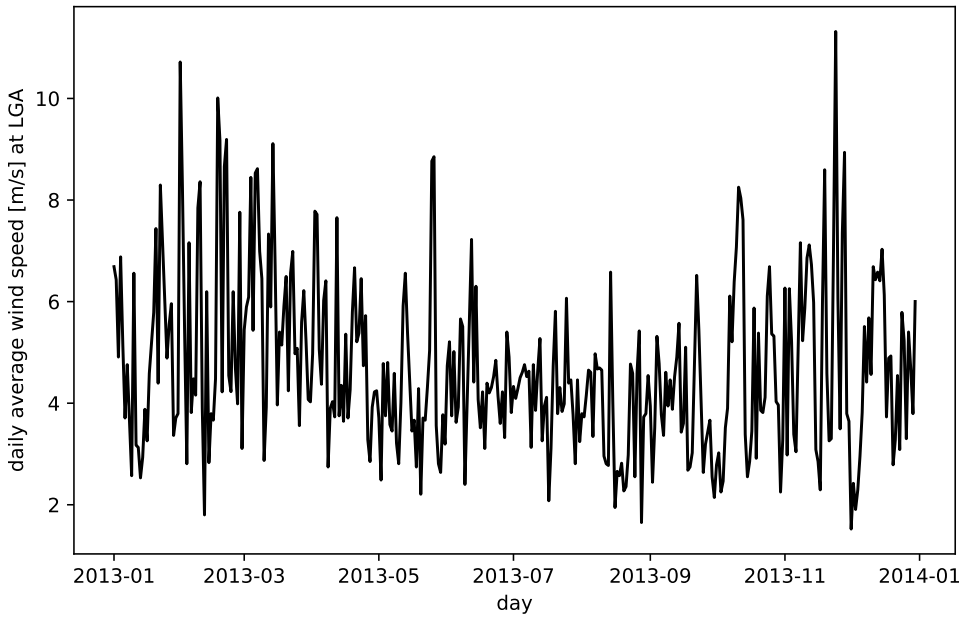
- `origin` – weather station: LGA, JFK, or EWR,
- `year`, `month`, `day`, `hour` – time of recording,
- `temp`, `dewp` – temperature and dew point in degrees Fahrenheit,
- `humid` – relative humidity,
- `wind_dir`, `wind_speed`, `wind_gust` – wind direction (in degrees), speed and gust speed (in mph),
- `precip` – precipitation, in inches,
- `pressure` – sea level pressure in millibars,
- `visib` – visibility in miles,
- `time_hour` – date and hour (based on the `year`, `month`, `day`, `hour` fields) formatted as `YYYY-mm-dd HH:MM:SS` (actually, `YYYY-mm-dd HH:00:00`). *However, due to a bug in the dataset, the data in this column are (incorrectly!) shifted by 1 hour. Do not rely on it unless you manually correct it.*

Then, create a single Jupyter/IPython notebook (see the *Artefacts* section below for all the requirements), where you perform what follows.

Q1. Convert all columns so that they use metric (International System of Units, SI) or derived units: `temp` and `dewp` to Celsius, `precip` to millimetres, `visib` to metres, as well as `wind_speed` and `wind_gust` to metres per second. Replace the data in-place (overwrite existing columns with new ones).

Q2. Compute *daily* mean wind speeds for the `LGA` airport (~365 total speed values, for each day separately; you can, for example, group the data by year, month, and day at the same time).

Q3. Present the daily mean wind speeds at `LGA` (~365 aforementioned data points) in a single plot, e.g., using the `matplotlib.pyplot.plot` function. The x-axis labels should be human-readable and intuitive (e.g., month names or dates). Reference result:



Q4. Identify the ten windiest days at `LGA` (dates and the corresponding mean *daily* wind speeds).

Reference result:

```
##             wind_speed
## date
## 2013-11-24       11.32
## 2013-01-31       10.72
## 2013-02-17       10.01
## 2013-02-21        9.19
## 2013-02-18        9.17
## 2013-03-14        9.11
## 2013-11-28        8.94
## 2013-05-26        8.85
## 2013-05-25        8.77
## 2013-02-20        8.66
```

Important. *All packages must be imported and data must be loaded at the beginning of the file (only once!).*

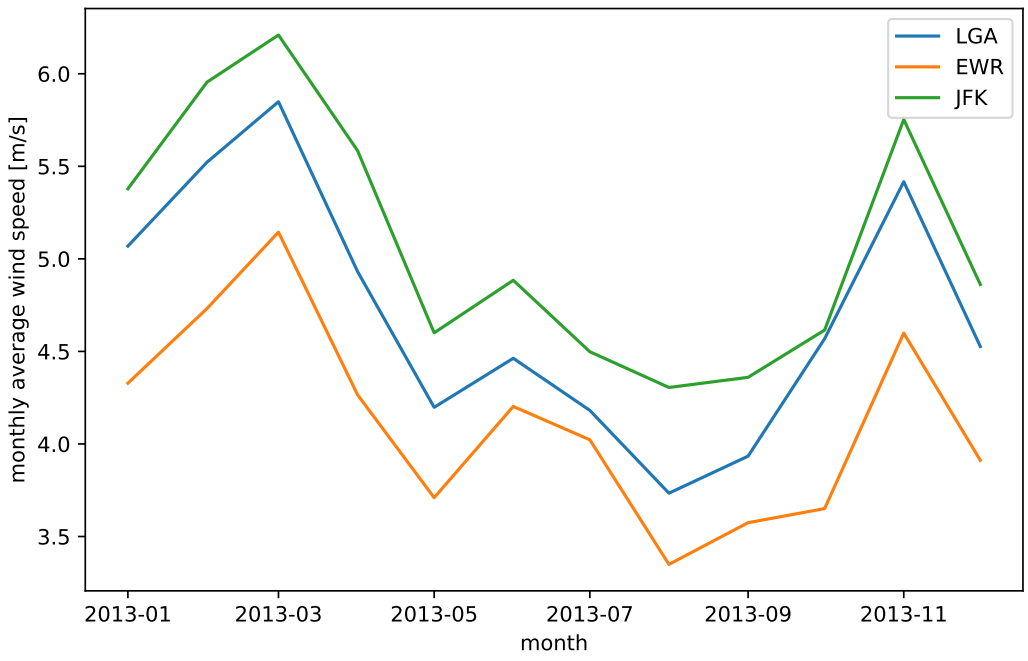
3 Question 5 - 6 for Postgraduate (SIT731) (*)

Postgraduate students, apart from the above tasks, are additionally **required** to solve/address/discuss what follows. Integrate these new requirements into the above subtasks (do not create a separate section of the report).

Q5. Compute the monthly mean wind speeds for all the three airports.

There is one obvious outlier amongst the observed wind speeds. Locate it (programmatically, do not hardcode the date/day/row number) and replace it with `np.nan` (NaN) before computing the means.

Q6. Draw the monthly mean wind speeds for the three airports on the same plot (three curves of different colours). Add a readable legend. Reference result:



4 Optional Features (**)

The following suggestions are not part of the requirements for a pass grade, therefore you can skip them. Nevertheless, you might still want to tackle them, as only practice makes perfect.

1. For the JFK airport, list all missing temperature readings. This should include not only the temperatures explicitly marked as missing values, but also the records that were completely omitted, for instance 2013-02-21 06:00:00.
2. Add the missing records to the dataset (just the date-time information, with all the remaining fields being set to `NaN`).
3. Compute the daily average temperatures, this time by linearly interpolating between the preceding and following non-missing data, e.g., a temperature sequence of `[..., 10, NaN, NaN, 40, ...]` should be transformed to `[..., 10, 20, 30, 40, ...]`.
4. Draw a plot of average daily temperatures comparing the missing value-omitted vs linearly interpolated cases.

5 Artefacts

The solution to the task must be included in a single Jupyter/IPython notebook (an .ipynb file) running against a Python 3 kernel. The use of G**gle Colab is discouraged. Nothing beats a locally-installed version where you have full control over the environment. Do not become dependent on third-party middlemen/distributors. Choose freedom instead.

Make sure that your notebook has a **readable structure**; in particular, that it is divided into sections. Use rich Markdown formatting (text in dedicated Markdown chunks – not just Python comments).

Do not include the questions/tasks from the task specification. Your notebook should read nicely and smoothly – like a report from data analysis that you designed yourself. Make the flow read natural (e.g., *First, let us load the data on… Then, let us determine… etc.*). Imagine it is a piece of work that you would like to show to your manager or clients — you certainly want to make a good impression. Check your spelling and grammar. Also, use formal language.

At the start of the notebook, you need to provide: the **title** of the report (e.g., *Task 42: How Much I Love This Unit*), your **name**, **student number**, **email address**, and whether you are an **undergraduate (SIT220) or postgraduate (SIT731)** student.

Then, add 1–2 introductory paragraphs (an introduction/abstract – what the task is about).

Before each nontrivial code chunk, briefly **explain** what its purpose is. After each code chunk, **summarise and discuss the obtained results** (in a few sentences).

Conclude the report with 1–2 paragraphs (summary/discussion/possible extensions of the analysis etc.).

Limitations of the OnTrack ipynb-to-pdf renderer:

Ensure that your report as seen in OnTrack is aesthetic (see *Download submision PDF* after uploading the .ipynb file). The OnTrack ipynb-to-pdf renderer is imperfect. We work with what we have. Here are the most common Markdown-related errors.

- Do not include any externally loaded images (via the `![label](href)` Markdown command), for they lead to upload errors.
- Do not input HTML code in Markdown.

- Make sure you leave one blank line before and after each paragraph and bullet list. Do not use backslashes at the end of the line.
- Currently, also *LaTeX formulae* and Markdown tables are not recognised. However, they do not lead to any errors.

Checklist:

1. Header, introduction, conclusion (Markdown chunks).
2. Text divided into sections, all major code chunks commented and discussed in your own words (Markdown chunks).
3. Every subtask addressed/solved. In particular, all reference results that are part of the task specification have been reproduced (plots, computed aggregates, etc.).
4. The report is readable and neat. In particular:
    - all code lines are visible in their entirety (they are not too long),
    - code chunks use consecutive numbering (select *Kernel - Restart and Run All* from the Jupyter menu),
    - rich Markdown formatting is used (`# Section Title`, `* bullet list`, `1. enumerated list`, `| table |`, `*italic*`, etc.),
    - the printing of unnecessary/intermediate objects is minimised (focus on reporting the results specifically requested in the task specification).

Submissions which do not *fully* (100%) conform to the task specification *on* the cut-off date will be marked as FAIL.

Good luck!

6 Intended Learning Outcomes

ULO	Is Related?
ULO1 (Data Processing/Wrangling)	YES
ULO2 (Data Discovery/Extraction)	YES
ULO3 (Requirement Analysis/Data Sources)	YES
ULO4 (Exploratory Data Analysis)	YES
ULO5 (Data Privacy and Ethics)	NO