SIT220/731 2022.T3: Task 4P

Working with *pandas* Data Frames (Heterogeneous Data)

Last updated: 2022-12-12

Contents

1 Task 1

2 Additional Tasks for Postgraduate (SIT731) Students (*) 2

3 Optional Features (**) 2

4 Artefacts 2

5 Intended Learning Outcomes 3

1 Task

This task is related to Module 4 (see the *Learning Resources* on the unit site; see also Chapters 10, 11, 12, 16 of our book). Aim at submitting your solution by the end of Week 9. In case of any problems/questions, do hot hesitate to attend our on-campus/online classes. The hard deadline is Week 11 (Friday).

Download the following data file from our unit site (*Learning Resources* → *Data*):

- nycflights13_weather.csv.gz

It gives the hourly meteorological data for three airports in New York: LGA, JFK, and EWR for the whole year of 2013.

Columns are:

- `origin` – weather station: LGA, JFK, or EWR,
- `year`, `month`, `day`, `hour` – time of recording,
- `temp`, `dewp` – temperature and dewpoint 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 are shifted by 1 hour.

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

1. 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 kilometres, as well as wind_speed and wind_gust to km/h. Replace the data in-place (overwrite existing columns with new ones).
2. Convert the time_hour column (in-place) to the datetime64 type and then subtract one hour so that data match the information stored in the month, day, and hour fields.
3. Compute daily mean temperatures (360+ average temperatures for each day separately) for the JFK airport with missing hourly temperature measurements ignored (removed) whatsoever (e.g., mean of [10, NaN, 20] is simply 15).
4. Present the daily mean temperatures (360+ data points) in a single plot. The x-axis labels should be human-readable and intuitive (e.g., month names).
5. Find the five hottest days.

2 Additional Tasks for Postgraduate (SIT731) Students (*)

Postgraduate students, apart from the above tasks, are additionally **required** to solve/address/discuss what follows.

1. Compute the daily mean temperatures also for the EWR and LGA airports.
2. Draw the daily mean temperatures for the three airports in the same plot (three curves of different colours). Add a readable legend.

3 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. Mark the days with greater mean temperature than in the preceding day in red and those with smaller – in blue (in the plot).
2. 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.
3. Add the missing records to the dataset (just the date-time information, with all the remaining fields being set to NaN).
4. Re-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, ...].
5. Draw a plot of average daily temperatures comparing the missing value-omitted vs linearly interpolated cases.

4 Artefacts

The solution to the task must be included in a single Jupyter/IPython notebook (an .ipynb file) running against a Python 3 kernel.

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).

Imagine it is a report 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.).

Submit one file via OnTrack:

1. the version of the Jupyter/IPython notebook converted to a PDF file (e.g., via *File → Export Notebook As → PDF* or convert to HTML and from that to PDF with your web browser; any method will do).

You do not need to submit the .ipynb file via OnTrack, but you must store it for further reference – a marking tutor might ask for it later, e.g., at the end of the trimester.

5 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)	YES