DATA420-25S1 (C)

Assignment 1

GHCN Data Analysis using Spark

Due on Friday, April 11 by 5:00 PM.

If you want to discuss the assignment material you can use the Discord server where the discussion will benefit all. If you have a question that requires an official answer you can use the forum on LEARN. If you have a more personal question you can email me or contact the class rep as needed.

A reminder that the Discord server is for discussion of concepts only, not for sharing code or answers to assignment questions.

Links

Report upload (pdf)
Supplementary material upload (zip, limited to 10 MB)

Discord server
Help forum for Assignment 1

Instructions

- Any assignments submitted after the deadline without obtaining an extension will receive a 20% penalty. You should aim to work on the assignment incrementally as you work through the first four modules.

- The forum is a great place to ask questions about the assignment! Any questions will be answered by a facilitator as soon as possible but students can also answer each other's questions so that you can all benefit from the answers.

- The data provided for the assignment is stored in Azure Blob Storage and outputs that you generate will be stored in Azure Blob Storage as well. Hadoop and Spark can both interact with Azure Blob Storage similar to how they interact with HDFS, but where the replication and distribution is handled by Azure instead. This makes it possible to read or write data in Azure over HTTPS where the path is prefixed by `wasbs://`. Follow the instructions in the notebook provided to load data from the data container and to save outputs that you generate to a separate user container where the path is prefixed by your username.

- Your report should be submitted as a single pdf file on LEARN. Any additional code, images, and supplementary material should be submitted separately as a single zip file on LEARN. You should **not** submit any outputs as part of your supplementary material, leave these in cloud storage.

- The body of your report should be between 3,000 and 5,000 words long, excluding your cover page, table of contents, references, appendices, and supplementary material. You need to be accurate and concise and you need to demonstrate depth of understanding.

- You should make sensible choices concerning margins, font size, spacing, and formatting. For example, margins between 0.5" and 1", a sans-serif font e.g. Arial with font size 11 or 12, line spacing 1 or 1.15, and sensible use of monospaced code blocks, tables, and images.

- You should reference any external resources using a citation format such as APA or MLA, including any online resources which you used to obtain snippets of code or examples. You must reference any use of Grammerly, ChatGPT, or any other generative AI tools to **improve** the quality of your own original work.

- **You must not use any content generated by ChatGPT or other generative AI tools directly in your work.** These are useful tools for aggregating existing knowledge but they lack the context and discernment to provide concise, specific answers to questions that require qualitative reasoning. We encourage **sensible** use of generative AI to improve the quality of your own original work.

- Your report should have the following sections and can also use question numbers as subheadings to group paragraphs, tables, and figures that you use to answer the questions that have been asked.

  - **Background**

    In this section you should give a brief overview of what you are doing in the assignment including any useful links or references to background material and a high level description of any difficulties that you had.

  - **Processing**

    *In this section you should describe the structure and content of the data and provide references for discussion in the sections below. You should describe the steps that you took to load, join, and check the data and discuss anything that you discovered along the way, but you should **not** include outputs other than answers to the questions that you have been asked.*

  - **Analysis**

    In this section you should answer the questions that you have been asked, give a high level summary of what you have done, and discuss any insights that you had. You should not list answers you have not explained. You should talk about any tasks that you were unable to complete and explain why.

  - **Visualizations**

    In this section you should answer the questions, present your visualizations, give a high level summary of what you have done, and discuss any insights that you had. You should talk about any visualizations that you were unable to generate and why.

  - **Conclusions**

    In this section you should give a brief overview of what you have achieved and what you have learned.

  - **References**

    In this section you should list all references that you have used or referenced.

DATA

In this assignment we will study the weather data in the Global Historical Climatology Network (GHCN), a database of climate summaries from land surface stations around the world. The data extends back over 250 years and is collected from more than 20 independent sources, each of which have passed quality assurance reviews.

- Global Historical Climatology Network (GHCN)

The daily climate summaries contain records from over 100,000 stations in 200 countries and territories around the world. There are several daily variables, including maximum and minimum temperature, total daily precipitation, snowfall, and snow depth; however, about half of the stations report precipitation only. The records vary by station and cover intervals ranging from less than a year to over 100 years in total.

The daily climate summaries are supplemented by metadata further identifying the stations, countries, states, and elements inventory specific to each station and time period. These provide human readable names, geographical coordinates, elevations, and date ranges for each station variable in the inventory.

Daily

The daily climate summaries are comma separated, where each field is separated by a comma (,) and where null fields are empty. A single row of data contains an observation for a specific station and day, and each variable collected by the station is on a separate row.

The following information defines each field in a single row of data covering one station day. Each field described below is separated by a comma (,) and follows the order below from left to right in each row.

Name	Type	Summary
ID	Character	Station code
DATE	Date	Observation date formatted as YYYYMMDD
ELEMENT	Character	Element type indicator
VALUE	Real	Data value for ELEMENT
MEASUREMENT FLAG	Character	Measurement Flag
QUALITY FLAG	Character	Quality Flag
SOURCE FLAG	Character	Source Flag
OBSERVATION TIME	Time	Observation time formatted as HHMM

The specific ELEMENT codes and their units are explained in Section III of the GHCN Daily README along with the MEASUREMENT FLAG, QUALITY FLAG, and SOURCE FLAG. The OBSERVATION TIME field is populated using the NOAA / NCDC Multinetwork Metadata System (MMS).

Metadata tables

The station, country, state, and variable inventory metadata tables are fixed width text formatted, where each column has a fixed width specified by a character range and where null fields are represented by whitespace instead.

Stations

The stations table contains geographical coordinates, elevation, country code, state code, station name, and columns indicating if the station is part of the GCOS Surface Network (GSN), the US Historical Climatology Network (HCN), or the US Climate Reference Network (CRN).

Name	Range	Type
ID	1 - 11	Character
LATITUDE	13 - 20	Real
LONGITUDE	22 - 30	Real
ELEVATION	32 - 37	Real
STATE	39 - 40	Character
NAME	42 - 71	Character
GSN FLAG	73 - 75	Character
HCN/CRN FLAG	77 - 79	Character
WMO ID	81 - 85	Character

Countries

The countries table contains country name only.

Name	Range	Type
CODE	1 - 2	Character
NAME	4 - 64	Character

States

The states table contains state name only.

Name	Range	Type
CODE	1 - 2	Character
NAME	4 - 50	Character

Inventory

The inventory table contains the set of elements recorded by each station, along with the time period each element was recorded.

Name	Range	Type
ID	1 - 11	Character
LATITUDE	13 - 20	Real
LONGITUDE	22 - 30	Real
ELEMENT	32 - 35	Character
FIRSTYEAR	37 - 40	Integer
LASTYEAR	42 - 45	Integer

TASKS

The assignment is separated into a number of sections, each of which explore the data in increasing detail from processing to analysis to visualizations. You will need to plan your time carefully to complete the assignment by the end of Module 4.

Each question is broken down into the following,

- **What you need to do**

  Tasks that you need to step through in order before you start your write up.

- **What should be included in your write up**

  Specific items that you should include in your write up or general comments on what you should talk about as you describe your method, present your results, and discuss your conclusions step by step.

- **Tips**

  Any other suggestions to help you if you get stuck.

These supplement the report requirements detailed on the cover page. In general, your write up should be accurate and concise and should demonstrate depth of understanding. The tasks are intentionally detailed to make it easier to work through the assignment step by step.

Processing

In this section you will explore datasets at a high level.

Q1 First you will investigate the `daily`, `stations`, `states`, `countries`, and `inventory` data provided in cloud storage in `wasbs://campus-data@madsstorage002.blob.core.windows.net/ghcnd/` using the `hdfs` command.

What you need to do

Follow the instructions in the notebook provided to explore each dataset using the `hdfs` command without loading any data into memory and answer the following questions:

(a) How is the data structured?

(b) How many years are contained in `daily`, and how does the size of the data change?

(c) What is the total size of all of the data, and how much of that is `daily`?

What you should include in your write up

(1) A visualization of the directory tree and a brief written summary.

(2) A visualization of the change in size in `daily` over time and a brief written summary.

(3) A table containing the names and sizes of the datasets.

(4) A written description of how the sizes of the other datasets compare to `daily`.

Q2 You will now load each dataset to ensure the descriptions are accurate and that you can apply the schema either as the data is loaded or by casting columns as they are extracted by manually processing the text records.

What you need to do

Extend the code example in the notebook provided by following the steps below.

(a) Define a schema for `daily` based on the description above or in the GHCN Daily README. This schema should use the data types defined in pyspark.sql.

(b) Modify the `spark.read.csv` command to load a subset of the most recent year of `daily` into Spark so that it uses the schema that you defined in step (a). Did anything go wrong when you tried to use the schema? What data types did you end up using and why?

(c) Load each of `stations`, `states`, `countries`, and `inventory` datasets into Spark and find a way to extract the columns and data types in the descriptions above. You will need to parse the fixed width text formatting by hand, as there is no method to load this format implemented in the standard `spark.read` library. You should use `pyspark.sql.functions.substring` to extract the columns based on their character range.

(d) How many rows are there in each metadata table?

What you should include in your write up

(1) A brief summary of the data types that you used in your schema and why.

(2) Any extra processing that you needed to do to load the datasets successfully.

(3) A table containing the number of rows of each of the metadata tables.

Q3 Next you will combine relevant information from the metadata tables by joining on station, state, and country to give an enriched `stations` table that we can use for filtering based on attributes at a station level.

What you need to do

(a) Extract the two character country code from each station code in `stations` and store the output as a new column using the `withColumn` method.

(b) LEFT JOIN `stations` with `countries` using your output from step (a).

(c) LEFT JOIN `stations` and `states`, allowing for the fact that state codes are only provided for stations in the US.

(d) Based on `inventory`, what was the first and last year that each station was active and collected any element at all?

How many different elements has each station collected overall?

Further, count separately the number of core elements and the number of "other" elements that each station has collected overall. How many stations collect all five core elements? How many collect **only** precipitation and no other elements?

Note that we could also determine the set of elements that each station has collected and store this output as a new column using `pyspark.sql.functions.collect_set` but it will be more efficient to first filter `inventory` by element type using the element column and then to join against that output as necessary.

(e) LEFT JOIN your output from step (c) and your output from step (d).

This enriched table will be useful. Save it to the user container prefixed by your username, `wasbs://campus-user@madsstorage002.blob.core.windows.net/abc123/`. You should think carefully about the file format that you use e.g. `csv`, `csv.gz`, or `parquet` with respect to consistency and efficiency. For the rest of the assignment assume that `stations` refers to this enriched table with all of the new columns included.

What you should include in your write up

(1) A brief written summary of your method, your results, and any conclusions that you made as you went through the tasks step by step. You do **not** need to include screenshots of your output at each step but you should provide an overview of the schema that you have derived by step (d). You could list each column in table with a description of what it contains.

(2) Answers to the questions that you have been asked.

Q4 Next you will check for any missing stations in `daily`.

What you need to do

(a) LEFT JOIN a subset of `daily` and your `stations` table from Q3 step (e).

Are there any stations in your subset of `daily` that are not in `stations`?

How expensive do you think it would be to LEFT JOIN `daily` and `stations`? Can you think of a more efficient way to check if there are any stations in `daily` that are not in `stations` without using LEFT JOIN?

(b) Based on step (a) count the total number of stations in `daily` that are not in `stations` at all.

You may need to stop and restart your Spark application to increase the resources you have allocated. You may increase your resources up to 4 executors, 2 cores per executor, 4 GB of executor memory, and 4 GB of master memory.

What you should include in your write up

(1) A clear explanation of how expensive it would be to LEFT JOIN all of `daily` and `stations`.

(2) A clear explanation of at least one alternative method.

(3) An answer for the number of stations in `daily` that are not in `stations` at all.

Analysis

In the section you will answer specific questions about the data using the code that you have developed.

Q1 First it will be helpful to know more about the `stations` themselves before we study the `daily` climate summaries in more detail.

What you need to do

(a) How many stations are there in total? How many stations were active so far in 2025?

How many stations are in each of the GCOS Surface Network (GSN), the US Historical Climatology Network (HCN), and the US Climate Reference Network (CRN)? Are there any stations that are in more than one of these networks?

(b) How many stations are there in the Southern Hemisphere?

Some of the countries in the database are territories of the United States as indicated by the name of the country. How many stations are there in total in the territories of the United States around the world, excluding the United States itself?

(c) Count the total number of stations in each country, and join these counts onto `countries` so that we can use these counts later if desired.

Do the same for `states` and save a copy of each table to your output directory.

What you should include in your write up

(1) A concise summary of the counts that you have been asked to generate in steps (a) and (b).

(2) Answers to any other questions that you have been asked.

Q2 You can create user defined functions in Spark by taking native Python functions and wrapping them with `pyspark.sql.functions.udf` which allows you to apply a function to each row using columns as inputs. You may find this functionality useful.

What you need to do

(a) Write a Spark function that computes the geographical distance between two stations using their latitude and longitude as arguments. You can test this function by using CROSS JOIN on a small subset of `stations` to generate a table with two stations in each row.

Note that there is more than one way to compute geographical distance, choose a method that at least takes into account that the earth is spherical.

(b) Apply this function to compute the pairwise distances between all stations in New Zealand, and save the result to your output directory.

What two stations are geographically closest together in New Zealand?

What you should include in your write up

(1) A brief summary of the distance function that you have implemented. You should explain how the function takes into account that the earth is spherical and you should clearly cite any sources that you have used.

(2) A brief summary of the two stations that are geographically closest together in New Zealand, including both their station name and ID. You should also include the distance between the two stations.

Q3 Next we will study the `daily` climate summaries in more detail.

What you need to do

(a) Count the number of rows in `daily`.

Note that this will take a while if you are only using 2 executors and 1 core per executor, and that the amount of driver and executor memory should not matter unless you actually try to cache or collect all of `daily`. You should **not** try to cache or collect all of `daily`.

(b) Filter `daily` using the `filter` or the `where` command to obtain the subset of observations containing the five core elements described in `inventory`.

How many observations are there for each of the five core elements?

Which element has the most observations?

(c) Many stations collect TMIN and TMAX, but do not necessarily report them simultaneously due to issues with data collection or coverage. Determine how many observations of TMAX do not have a corresponding observation of TMIN.

How many unique stations contributed to these observations?

What you should include in your write up

(1) A concise summary of the counts that you have been asked to generate in each of the parts.

(2) Any conclusions that you made as you went through the tasks step by step.

(3) A clear explanation of how you determined how many observations of TMAX have no TMIN.

(4) Answers to any other questions that you have been asked.

Visualizations

In the section you will develop time series and geospatial visualizations of the `daily` climate summaries which will require you to collect or copy some of your outputs to the master node in order to visualize them together. **This will be fine provided you collect or copy outputs that have been aggregated at a reasonably coarse level.**

Please be careful and do not accidentally collect or copy all of `daily` onto the master node or we will run out of memory very quickly.

Q1 First you will plot observations of TMIN and TMAX for stations in New Zealand.

What you need to do

(a) Filter `daily` to obtain all observations of TMIN and TMAX for all stations in New Zealand, and save the result to your output directory.

How many observations are there, and how many years are covered by the observations?

(b) Plot time series for TMIN and TMAX together for each station in New Zealand using Python, R, or any other programming language or data visualization tool that you know well. Also, plot the average time series for TMIN and TMAX together for the entire country.

What you should include in your write up

(1) A brief written summary of the number of observations and how many years are covered.

(2) A visualization of TMIN and TMAX for each station in New Zealand. You could use one grid of subplots covering an entire page or you could use separate plots where you include one in your report as an example and provide the rest in your appendices **and** your supplementary material. You should style your visualization appropriately.

(3) A separate visualization of the average TMIN and TMAX for all of New Zealand.

(4) A brief written summary of how the visualizations were generated. You should **not** include code in your report but you should describe any assumptions that you made along the way.

Tips

(1) You should consider the following

- Temporal smoothing
- Gaps
- How you can line up and compare each of the different time series consistently
- Choosing a title, labels, and a legend
- Choosing sensible layouts, colors, sizes, fonts, and other visualization styling

Q2 Next you will plot precipitation observations for stations around the world, grouping by year and by country to provide a sensible temporal and spatial resolution for our visualizations. You should look up sensible principles for visualization before continuing.

What you need to do

(a) Group the precipitation observations by year and by country. Compute the average daily rainfall in each year for each country, and save this result to your output directory.

Generate descriptive statistics for the average rainfall.

Which country has the highest average rainfall in a single year across the entire dataset?

Is this result sensible?

(b) Find an elegant way to plot the average rainfall in 2024 for each country. There are many ways that you could do this, such as using a choropleth to color a map based on the average rainfall.

Are there any gaps or missing values in your plot?

What you should include in your write up

(1) A brief written summary of the descriptive statistics for the average rainfall in each year for each country. You should comment on how sensible the statistics are and identify any outliers before continuing on to the final visualizations.

(2) A visualization of the average rainfall in 2024 for each country.

(3) A brief written summary of how the visualizations were generated. You should **not** include your code in your report but you should describe any decisions or assumptions that you made along the way.

(4) Answers to any other questions that you have been asked.

Tips

(1) You should consider the following

- Any outliers that might be skewing your results
- How the countries in this dataset line up with the countries in your plotting library
- Other gaps
- Choosing a title, labels, and a legend
- Choosing sensible colors, sizes, fonts, and other visualization styling