DATA420-21S2 (C)

Assignment 1

GHCN Data Analysis using Spark

Xin Gao (43044879)

September 17, 2021

Foreword

In this assignment we will investigate the weather data collected by the Global Historical Climate Network (GHCN), an integrated database of climate summaries from weather stations around the world. This data covers the last 259 years, is collected from over 20 independent sources, and contains records from over 100,000 stations in 219 countries around the world.

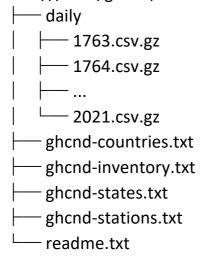
The code used to solve each of the questions is provided separately and is commented well. This report describes the approach used, answers any questions, and describes anything unexpected along the way.

Processing

Q1:

a) The data is in two parts: the collection of daily climate summaries, which are grouped by year into separate files and stored in a gzip compressed csv format, and the associated metadata providing additional data specific to each of the stations, states, countries, and inventory. And an explanatory readme text file is also provided.

The data is structured according to the directory tree below hdfs:///data/ghcnd/



where stations, states, countries, inventory, and readme are files and daily is a directory containing one file for each year.

b) There are 259 files in daily, one file for each year from 1763 to 2021. Note that by using -ls command to peek at the daily directory, it shows there are 259 items in the directory. However, by piping the file list to "wc -l" command, it returns 260. This is probably because the "wc -l" command

merely counts the number of newline characters(\n) in the text file which in this case is a list of file names. I assume there is a newline at end of the list, so it counts one line more. We should be careful with it.

The size of the data increased significantly over the years from only 3.3K in 1763 to a peak of 221.3M in 2010. After 2010, the size started decreasing and dropped to 142.4M in 2020. And 2021 has recorded 78M data so far.

c) The total size of the data is about 15.6G, most of which is daily. The other files only contributes 40.6M to the total. As the daily data is compressed, and the actual size of the uncompressed data will be significantly higher.

Q2:

- a) Although I defined schema for each of daily, stations, states, countries, and inventory, only the daily schema is used to load the daily data. Other schemas are not used because using read.text to load files and followed by selecting automatically created the schema.
- Note that to look at the head of daily, we had to pipe the file through gunzip and pipe the decompressed csv data to head or tail.
- b) The "DATE" column displays well because I specified the DateType in the daily schema. The "VALUE" column is defined as integer type as there are no decimal values. However, note that the temperature values are to tenths of degrees according to the readme file. That means, for example, the value 278 should be interpreted as 27.8 degrees.
- The "OBSERVATION TIME" column only contains the hour information, so it is not proper to define it as TimestampType. It would be handy if I just load it as string type since I do not need the data to do any calculation. It is the same that I loaded "LATITUDE", "LONGITUDE", "ELEVATION" etc as string type even though they are numeric.
- "MEASUREMENT_FALG", "QUALITY_FLAG" and "OBSERVATION_TIME" columns include many nulls.
- c) To parse the fixed width text formatting, I used read.text and then specified the number of characters for each column, gave a column name and then defined the data type. Note that read.text parses the data to string type one character by one character which means the white spaces are included exactly in the data frame. So I used F.trim to strip the white spaces. (see pyspark codes)

The rows in each metadata table are as below

metadata	stations	states	countries	inventory
rows	118493	74	219	704963

There are 110407 stations that do not have a WMO ID.

Q3:

a) Please refer to pyspark codes and the outcome is as below

LATITUDE	LONGITUDE	ELEVATION STATE	NAME	GSN_FLAG HCN/CRN_FLAG	WMO_ID COUNTRY_CO
17.1167	-61.7833	10.1	ST JOHNS COOLIDGE FLD		AC
17.1333	-61.7833	19.2	ST JOHNS	i i	AC
25.3330	55.5170	34.0	SHARJAH INTER. AIRP	GSN	41196 AE
25.2550	55.3640	10.4	DUBAI INTL		41194 AE
24.4330	54.6510	26.8	ABU DHABI INTL	l I	41217 AE
24.2620	55.6090	264.9	AL AIN INTL		41218 AE
35.3170	69.0170	3366.0	NORTH-SALANG	GSN	40930 AF
34.2100	62.2280	977.2	HERAT	i i	40938 AF
34.5660	69.2120	1791.3	KABUL INTL		40948 AF
31.5000	65.8500	1010.0	KANDAHAR AIRPORT	i i	40990 AF
	17.1167 17.1333 25.3330 25.2550 24.4330 24.2620 35.3170 34.2100 34.5660	17.1167 -61.7833 17.1333 -61.7833 25.3330 55.5170 25.2550 55.3640	17.1167 -61.7833 10.1 17.1333 -61.7833 19.2 25.3330 55.5170 34.0 25.2550 55.3640 10.4 24.4330 54.6510 26.8 24.2620 55.6090 264.9 35.3170 69.0170 3366.0 34.2100 62.2280 977.2 34.5660 69.2120 1791.3	17.1333 -61.7833 19.2 ST JOHNS 25.3330 55.5170 34.0 SHARJAH INTER. AIRP 25.2550 55.3640 10.4 DUBAI INTL 24.4330 54.6510 26.8 ABU DHABI INTL 24.2620 55.6690 264.9 AL AIN INTL 35.3170 69.0170 3366.0 NORTH-SALANG 34.2100 62.2280 977.2 HERAT 34.5660 69.2120 1791.3 KABUL INTL	17.1167 -61.7833 10.1

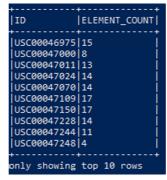
b) Please refer to pyspark codes and the outcome is as below

ID	LATITUDE	LONGITUDE	ELEVATION	STATE	NAME	GSN_FLAG	HCN/CRN_FLAG W	MO_ID	COUNTRY_CODE	COUNTRY	_NAME		STATE_NAME	COUNTRY	_NAME	
ACW00011604	17.1167	-61.7833	10.1		ST JOHNS COOLIDGE FLD				AC	Antigua	and	Barbuda	null	Antigua	and B	arbuda
ACW00011647	17.1333	-61.7833	19.2	ĺ	ST JOHNS	i i			AC	Antigua	and	Barbuda	null	Antigua	and B	arbuda
AE000041196	25.3330	55.5170	34.0	ĺ	SHARJAH INTER. AIRP	GSN	4	1196	AE	United	Arab	Emirates	null	United A	Arab E	mirates
AEM00041194	25.2550	55.3640	10.4	ĺ	DUBAI INTL	i i	4	1194	AE	United	Arab	Emirates	null	United	Arab E	mirates
AEM00041217	24.4330	54.6510	26.8	ĺ	ABU DHABI INTL	i i	4	1217	AE	United	Arab	Emirates	null	United	Arab E	mirates
AEM00041218	24.2620	55.6090	264.9	İ	AL AIN INTL	i i	4	1218	AE	United	Arab	Emirates	null	United A	Arab E	mirates
AF000040930	35.3170	69.0170	3366.0	ĺ	NORTH-SALANG	GSN	4	0930	AF	Afghani	stan		null	Afghani	stan	
AFM00040938	34.2100	62.2280	977.2	ĺ	HERAT	i i	4	0938	AF	Afghani	stan		null	Afghani	stan	
AFM00040948	34.5660	69.2120	1791.3	ı	KABUL INTL	1 1	4	0948	AF	Afghani	stan		null	Afghani	stan	
AFM00040990	31.5000	65.8500	1010.0	ĺ	KANDAHAR AIRPORT	i i	4	0990	AF	Afghani	stan		null	Afghani	stan	
									+	+			+			
nly showing	top 10 r	OWS														

c) Please refer to pyspark codes and the outcome is as below

ID	LATITUDE	LONGITUDE	ELEVATION	STATE	NAME	GSN_FLAG	HCN/CRN_FLAG	WMO_ID	COUNTRY_CODE	COUNTRY_	NAME	STATE_NAME	COUNTRY_NAME		STATE_NAM
ACW00011604	17.1167	-61.7833	10.1		ST JOHNS COOLIDGE FLD				IAC	Antigua	and Barbuda	null	Antigua and	Barbuda	null
ACW00011647	17.1333	-61.7833	19.2	i	ST JOHNS	i i	i i		AC	Antigua	and Barbuda	null	Antigua and	Barbuda	null
AE000041196	25.3330	55.5170	34.0	i	SHARJAH INTER. AIRP	GSN	į į	41196	AE	United A	rab Emirates	null	United Arab	Emirates	null
AEM00041194	25.2550	55.3640	10.4	İ	DUBAI INTL	i i	į į	41194	AE	United A	rab Emirates	null	United Arab	Emirates	null
AEM00041217	24.4330	54.6510	26.8	İ	ABU DHABI INTL	i i	į į	41217	AE	United A	rab Emirates	null	United Arab	Emirates	null
AEM00041218	24.2620	55.6090	264.9	i	AL AIN INTL		į į	41218	AE	United A	rab Emirates	null	United Arab	Emirates	null
AF000040930	35.3170	69.0170	3366.0	l .	NORTH-SALANG	GSN	l l	40930	AF	Afghanis	tan	null	Afghanistan		null
AFM00040938	34.2100	62.2280	977.2	İ	HERAT	i i	į į	40938	AF	Afghanis	tan	null	Afghanistan		null
AFM00040948	34.5660	69.2120	1791.3	i	KABUL INTL			40948	AF	Afghanis	tan	null	Afghanistan		null
AFM00040990	31.5000	65.8500	1010.0	l	KANDAHAR AIRPORT		l l	40990	AF	Afghanis	tan	null	Afghanistan		null
		+											 		
nly showing	top 10 re	OWS													

d) The different elements each station collected are shown as below



There are 20289 stations collect all five core elements. There are 16136 stations only collected precipitation.

- e) As the file size is about 11M, I choose to store the output as .csv files. Considering the size is small, it is not necessary to compress it in a distributed file system. Also, it is easy to output the file to a local computer and open it in Excel.
- f) There is no (0) stations in the subset of daily that is not in stations at all. I assume It will be very expensive to LEFT JOIN all of daily and stations. As the whole daily data include billions of rows (see A-Q4(a)) and the size is 15.6G, the data after LEFT JOIN will be huge because billions of station rows of station information will be joined. The size will probably be more than doubled.

We can do it without using LEFT JOIN. As we only need to compare the two ID columns and find the IDs appeared in daily but not in stations. Then anti join on ID column is the perfect choice for this purpose.

(see the pyspark code which returns the same result (0) as using LEFT JOIN)

Analysis

Q1:

a) Please refer to pyspark codes and the outcome is as below There are 118493 stations in total.

There are 41311 stations were active in 2020.

The numbers of stations that are in each of the GCOS Surface Network (GSN), the US Historical Climatology Network (HCN), and the US Climate Reference Network (CRN) are shown as below

FLAG	GSN	HCN	CRN
STATIONS	991	1218	0

There are 14 stations that are in more than one of these networks.

b) Please refer to pyspark codes Count the total number of stations in each country and each state as below

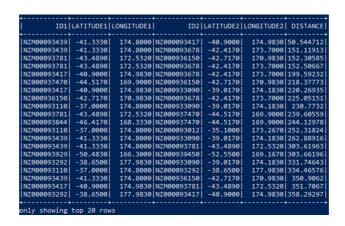
CODE	NAME	NUMBER OF STATIONS				
TI	Tajikistan	62				
MX	Mexico	5249				
NI	Nigeria	10				
SW	Sweden	1721				
UG	Uganda	8				
GM	Germany	1123				
HU	Hungary	10				
NH	Vanuatu	6				
TO	Togo	10				
MB	Martinique [France]	2				
+						
only	showing top 10 rows					

CODE	NAME	NUMBER OF STATIONS
INT	NORTHWEST TERRITORIES	137
ND	NORTH DAKOTA	545
NH	NEW HAMPSHIRE	431
AZ	ARIZONA	1534
МВ	MANITOBA	722
NM	NEW MEXICO	2033
AR	ARKANSAS	885
VI	VIRGIN ISLANDS	54
KS	KANSAS	1994
LA	LOUISIANA	734
+	+	+
only	showing top 10 rows	

c) Please refer to pyspark codes and the outcome is as below There are 25296 stations in the Southern Hemisphere. There are 339 stations in total that are in the territories of the United States excluding the United States itself.

Q2:

- a) I choose the "haversine" formula which takes into account that the earth is spherical. (see pyspark code)
- b) The station ID(NZ000093417) and the station ID(NZM00093439) are geographically the closest in New Zealand. The distance is about 50 kms.



Q3:

a) The default blocksize of HDFS is 128 M. The size of daily 2021 is 78M, and the size of daily 2015 is 198M. As a result, 2021 requires only one block while 2015 requires two blocks.

I used the command "hdfs fsck hdfs:///data/ghcnd/daily/2015.csv.gz -files - blocks" to check the individual block size. The file 2015 has two blocks, one is 128M, the other is 70M. (look at the "len" information in bytes)

/data/ghcnd/daily/2015.csv.gz 207618101 bytes, replicated: replication=8, 2 block(s): OK 0. BP-700027894-132.181.129.68-1626517177804:blk_1073744657_3833 len=134217728 Live_repl=8 1. BP-700027894-132.181.129.68-1626517177804:blk_1073744658_3834 len=73400373 Live_repl=8 I think it is still possible to load and apply transformations in parallel for the year 2021, because the block is divided into partitions which are the unit of transformation. As such spark will be able to load and apply transformations to multiple partitions of 2021 in parallel as long as the number of partitions is not 1. For file 2015, there are two blocks which means at least two partitions will be created. It will be loaded and transformed in parallel.

b) Please refer to pyspark codes and the outcome is as below There are 34899014 rows in file 2015.

There are 19099479 rows in file 2021.

There is only one tasks executed by each stage of each job.

The number of tasks executed seems not corresponding to the number of blocks in each input.

c) The number of observations from 2015 to 2021 (inclusive) is 228659433.

7 tasks were executed in the first stage which is loading the 7 files into spark. 1 task was executed in the second stage which is to count the observations.

When loading the compressed files, Spark load them one file by one task. After loading, Spark combines the data into one RDD and does the transformation (count() in this case).

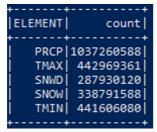
d) As there are 259 compressed files in daily, there will be 259 tasks when loading. When applying the transformation, however, based on the result of A-Q4, Spark will combine some small partitions together and then run in parallel which reduces the overall tasks.

If we want to increase the number of tasks, we can do repartition after loading all the files.

Q4:

- a) There are 2978405055 rows in daily (see pyspark codes).
- b) Refer to pyspark codes

 Number of observations for each of the five core elements as below



We can see PRCP(Precipitation) has the most observations.

c) Please refer to pyspark codes and the outcome is as below
There are 8689146 observations of TMIN do not have a corresponding observation of TMAX.

There are 27610 different stations contributed to these observations.

d) Please refer to pyspark codes and bash codes. It would be handy to output just one csv file to local directory for later plotting, so I repartitioned RDD to 1.

There are 468192 observations of TMIN and TMAX for all stations in New Zealand.

These observations covered 82 years.

Time series plot, please refer to the below links:
Time flow including all years for each station in New Zealand
https://public.tableau.com/views/MaxandmintemperatureinNZLongVersion/Dashboard1?:language=en-
GB&publish=yes&:display count=n&:origin=viz share link

Comparing the years for each station in New Zealand https://public.tableau.com/shared/7C6RMM23N?:display count=n&:origin=viz share link

e) Please refer to pyspark codes and bash codes

Equatorial Guinea has the highest average rainfall in a single year of 2000 across the entire dataset. The average rainfall is 4361mm. It makes sense because Equatorial Guinea is very close to the equator where water vapor massively condenses into rain.

The average rainfall for each country in a map as below link https://public.tableau.com/shared/Z3474NXFD?:display count=n&:origin=viz share link

When you choose year 2000, you will see the circle on Equatorial Guinea is standing out as it is extremely higher than others.