DATA 420 – Scalable Data Science

ASSIGNMENT 1

GHCN Data Analysis using Spark

Aakanksha Kapoor (72261392)

PROCESSING

Q1

(a) How is the data structured? Draw a directory tree to represent this in a sensible way.
(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? How much of that is daily?

Data is structured mainly into two categories: - daily (which is a directory) and metadata (which contains different text files).

Daily directory contains different years in .csv format. Metadata further contains four different text files: - stations, countries, states and inventory.

```
/data/shared/ghcnd/
├─ countries
├─ daily
│  ├─ 1763.csv.gz
│  ├─ 1764.csv.gz
│  ├─ 1765.csv.gz
│  ├─ ...
│  └─ 2020.csv.gz
├─ inventory
├─ states
└─ stations
```

There are 258 years contained in daily. The size of the data changes as it increases as we go down as the number of years increases.

Total size of the data is 15.5 GB. The size of the data is majorly from daily directory as the other files combined is overall less than 0.5GB.

Q2

(a) Define schemas for each of daily, stations, states, countries, and inventory based on the descriptions in this assignment and the GHCN Daily README. These should use the data types defined in pyspark.sql.
(b) Load 1000 rows of hdfs:///data/ghcnd/daily/2020.csv.gz into Spark using the limit command immediately after the read command. Was the description of the data accurate? Was there anything unexpected?
(c) Load each of stations, states, countries, and inventory into Spark as well. You will need to find a way to parse the fixed width text formatting, as this format is not included in the standard spark.read library. You could try using spark.read.format('text') and pyspark.sql.functions.substring or finding an

existing open source library. How many rows are in each metadata table? How many stations do not have a WMO ID?

For the schemas in daily, the files were in .csv format so it easily readable. On the other hand, the other files station, states, countries and inventory were in .txt format, so they were read by the use of open source library – select and then substring was found which is loaded.

The data of top 1000 rows was loaded using the limit command.

The description was not accurate as column names were not defined. The date time format was casted in string type object and time stamp object was also in the string type.

Each metadata contained following rows.

Country – 219

State – 74

Station – 115081

Inventory – 687141

WMO ID has 106993 stations.

Q3

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

(b) LEFT JOIN stations with countries using your output from part (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 only collected precipitation?

(e) LEFT JOIN stations and your output from part (d). This enriched stations table will be useful. Save it to your output directory. Think carefully about the file format that you use (e.g. csv, csv.gz, parquet) with respect to consistency and efficiency. From now on assume that stations refers to this enriched table with all the new columns included.

(f) LEFT JOIN your 1000 rows subset of daily and your output from part (e). Are there any stations in your subset of daily that are not in stations at all? How expensive do you think it would be to LEFT JOIN all of daily and stations? Could you determine if there are any stations in daily that are not in stations without using LEFT JOIN?

Stations that contained code were extracted with the help of .withcolumn command. Left join was performed on the stations and the countries containing codes. After that, left join was performed on the stations and states in US with codes.

The two character country code was extractwed from each station code in stations and the output was stored as a new column using the withColumn command.

Left join of stations with countries using this output was as follows:

CODE	ID	LATITUDE	LONGITUDE	ELEVATION	STATE	NAME	GSN FLAG	HCN/CRN FLAG	WMO ID	NAME
AC	ACW00011604	17.1167	-61.7833	10.1		ST JOHNS COOLIDGE...				Antigua and Barbuda
AC	ACW00011647	17.1333	-61.7833	19.2		ST JOHNS				Antigua and Barbuda
AE	AE000041196	25.333	55.517	34.0		SHARJAH INTER. AIRP	GSN		41196	United Arab Emirates
AE	AEM00041194	25.255	55.364	10.4		DUBAI INTL			41194	United Arab Emirates
AE	AEM00041217	24.433	54.651	26.8		ABU DHABI INTL			41217	United Arab Emirates
AE	AEM00041218	24.262	55.609	264.9		AL AIN INTL			41218	United Arab Emirates
AF	AF000040930	35.317	69.017	3366.0		NORTH-SALANG	GSN		40930	Afghanistan
AF	AFM00040938	34.21	62.228	977.2		HERAT			40938	Afghanistan
AF	AFM00040948	34.566	69.212	1791.3		KABUL INTL			40948	Afghanistan
AF	AFM00040990	31.5	65.85	1010.0		KANDAHAR AIRPORT			40990	Afghanistan
AG	AG000060390	36.7167	3.25	24.0		ALGER-DAR EL BEIDA	GSN		60390	Algeria
AG	AG000060590	30.5667	2.8667	397.0		EL-GOLEA	GSN		60590	Algeria
AG	AG000060611	28.05	9.6331	561.0		IN-AMENAS	GSN		60611	Algeria
AG	AG000060680	22.8	5.4331	1362.0		TAMANRASSET	GSN		60680	Algeria
AG	AGE00135039	35.7297	0.65	50.0		ORAN-HOPITAL MILI...				Algeria
AG	AGE00147704	36.97	7.79	161.0		ANNABA-CAP DE GARDE				Algeria
AG	AGE00147705	36.78	3.07	59.0		ALGIERS-VILLE/UNI...				Algeria
AG	AGE00147706	36.8	3.03	344.0		ALGIERS-BOUZAREAH				Algeria
AG	AGE00147707	36.8	3.04	38.0		ALGIERS-CAP CAXINE				Algeria
AG	AGE00147708	36.72	4.05	222.0		TIZI OUZOU			60395	Algeria

only showing top 20 rows

LEFT JOIN of stations and states, allowing for the fact that state codes are only provided for stations in the US was as follows: -

CODE	ID	LATITUDE	LONGITUDE	ELEVATION	STATE	NAME	GSN FLAG	HCN/CRN FLAG	WMO ID	NAME
AC	ACW00011604	17.1167	-61.7833	10.1		ST JOHNS COOLIDGE...				null
AC	ACW00011647	17.1333	-61.7833	19.2		ST JOHNS				null
AE	AE000041196	25.333	55.517	34.0		SHARJAH INTER. AIRP	GSN		41196	null
AE	AEM00041194	25.255	55.364	10.4		DUBAI INTL			41194	null
AE	AEM00041217	24.433	54.651	26.8		ABU DHABI INTL			41217	null
AE	AEM00041218	24.262	55.609	264.9		AL AIN INTL			41218	null
AF	AF000040930	35.317	69.017	3366.0		NORTH-SALANG	GSN		40930	null
AF	AFM00040938	34.21	62.228	977.2		HERAT			40938	null
AF	AFM00040948	34.566	69.212	1791.3		KABUL INTL			40948	null
AF	AFM00040990	31.5	65.85	1010.0		KANDAHAR AIRPORT			40990	null
AG	AG000060390	36.7167	3.25	24.0		ALGER-DAR EL BEIDA	GSN		60390	null
AG	AG000060590	30.5667	2.8667	397.0		EL-GOLEA	GSN		60590	null
AG	AG000060611	28.05	9.6331	561.0		IN-AMENAS	GSN		60611	null
AG	AG000060680	22.8	5.4331	1362.0		TAMANRASSET	GSN		60680	null
AG	AGE00135039	35.7297	0.65	50.0		ORAN-HOPITAL MILI...				null
AG	AGE00147704	36.97	7.79	161.0		ANNABA-CAP DE GARDE				null
AG	AGE00147705	36.78	3.07	59.0		ALGIERS-VILLE/UNI...				null
AG	AGE00147706	36.8	3.03	344.0		ALGIERS-BOUZAREAH				null
AG	AGE00147707	36.8	3.04	38.0		ALGIERS-CAP CAXINE				null
AG	AGE00147708	36.72	4.05	222.0		TIZI OUZOU			60395	null

only showing top 20 rows

Based on the inventory, the first and last year of each station for top 20 rows is shown below:-

ID	FirstYearActive	LastYearActive
CA006016970	2012	2018
CA006017400	2006	2012
CA006024010	1935	1951
CA00602K300	1978	2002
CA006032119	1970	2005
CA006037775	1938	2020
CA006037803	1933	1956
CA006040786	1968	1971
CA006040790	1957	1959
CA006045675	1973	1980
CA006046164	1952	1956
CA006046590	1959	1963
CA006050801	1926	1959
CA006051R65	1977	1979
CA006052258	1959	1973
CA006052563	1917	1960
CA006065015	1950	1957
CA006066873	1973	1983
CA006068980	1914	1990
CA006069165	1978	1989

only showing top 20 rows

Number of different elements has each station collected overall: -

ID	DistinctElementCount
AGE00147719	4
AJ000037989	5
AQC00914873	12
ASN00004031	4
ASN00004087	1
ASN00006011	11
ASN00006080	3
ASN00007031	4
ASN00007166	1
ASN00007187	4
ASN00008010	4
ASN00008093	10
ASN00008154	4
ASN00009072	4
ASN00009111	10
ASN00009166	4
ASN00009532	4
ASN00009578	1
ASN00009579	4
ASN00009850	4

only showing top 20 rows

The number of core elements that each station has collected overall:-

ID	CoreElementsCount
AGE00147719	3
AGM00060445	4
AJ000037679	1
AJ000037831	1
AJ000037981	1
AJ000037989	4
ALE00100939	2
AM000037719	4
AM000037897	4
AQC00914873	5
AR000000002	1
AR000087007	4
AR000087374	4
AR000875850	4
ARM00087022	3
ARM00087480	4
ARM00087509	4
ARM00087532	4
ARM00087904	4
ASN00001003	1

only showing top 20 rows

The number of other elements that each station has collected overall:-

ID	OtherElementsCount
AGE00147719	1
AGM00060445	1
AJ000037679	0
AJ000037831	0
AJ000037981	0
AJ000037989	1
ALE00100939	0
AM000037719	1
AM000037897	1
AQC00914873	7
AR000000002	0
AR000087007	1
AR000087374	1
AR000875850	1
ARM00087022	1
ARM00087480	1
ARM00087509	1
ARM00087532	1
ARM00087904	1
ASN00001003	0

only showing top 20 rows

Number of stations which collected all five core elements was 115024 while number of stations which collected only precipitation is 113070.

Enriched station table was saved in .csv file format as it is read easily.

After assuming that stations refer to this enriched table with all the new columns included, there are stations in daily that are not in stations at all. This operation is done in the subset of daily which is 1000 rows. If had to LEFTJOIN all of daily and stations, then it would be very expensive as loading full data in daily is time consuming as well as cost taking, after loading data of daily and station, it would perform left join which will be indeed very expensive. One way to determine the stations in daily that are not in stations can be done by finding the stations that are present in both daily and stations by INNERJOIN and then returning all the stations in daily except for the result of INNERJOIN which would be less expensive than the previous approach.

ANALYSIS

(**Q1**

a) How many stations are there in total? How many stations were active in 2000? 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) Count the total number of stations in each country, and store the output in countries using the withColumnRenamed command. Do the same for states and save a copy of each table to your output directory.

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

Total number of stations are 115081.

Number of stations active in 2000 is 1138.

Number of stations in GCOS surface network (GSN) is 991.

Number of stations in US Historical Climatology Network (HCN) is 1218

Number of stations in US Climate Reference Network (CRN) is 233

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

Number of stations in each country and state are stored in the output directory: - hdfs:///user/aka233/outputs/ghcnd/states.csv.gz

code	NAME	CODECOUNT
AU	Austria	13
BA	Bahrain	1
BB	Barbados	1
CQ	Northern Mariana ...	11
DR	Dominican Republic	5
EU	Europa Island [Fr...	1
EZ	Czech Republic	12
FR	France	110
MY	Malaysia	16
TI	Tajikistan	62
AJ	Azerbaijan	66
BG	Bangladesh	10
BL	Bolivia	36
CA	Canada	8818
IN	India	3807
IZ	Iraq	1
JM	Jamaica	3
MX	Mexico	5249
MZ	Mozambique	19
NI	Nigeria	10

only showing top 20 rows

CODE	NAME	CODECOUNT
IL	ILLINOIS	null
NJ	NEW JERSEY	null
NT	NORTHWEST TERRITO...	null
PI	PACIFIC ISLANDS	null
CA	CALIFORNIA	8818
IN	INDIANA	3807
OK	OKLAHOMA	null
WY	WYOMING	null
CT	CONNECTICUT	17
MN	MINNESOTA	null
WV	WEST VIRGINIA	null
MT	MONTANA	1
ND	NORTH DAKOTA	null
NH	NEW HAMPSHIRE	6
OH	OHIO	null
WI	WISCONSIN	1
AZ	ARIZONA	null
ID	IDAHO	104
MB	MANITOBA	2
SD	SOUTH DAKOTA	null

only showing top 20 rows

Number of stations in the southern hemisphere only is 25336.

Number of stations in total in the territories of the United States around the world, excluding the United States itself is 316.

Q2

(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 the geographically closest together in New Zealand?

In built functions radians, cos, sin, asin, sqrt are imported from math library.

As earth is spherical, so area is calculated as

area = sin(distance_latitude/2)**2 + cos(latitude_first) * cos(latitude_second) * sin(distance_longitute/2)**2

where

distance_latitude is the latitudinal difference between the two stations whereas distance_longiitude is the longitudinal distance between the two stations.

Spark function geographic_distance was constructed which accepts the arguments as longitude and latitude of two stations.

def geographic_distance(longitute_first, latitude_first, longitute_second, latitude_second)

This was then verified by performing CROSSJOIN on subset of the distance.

The same function was applied to compute pairwise distances between stations in New Zealand.

Two stations which are closest together in New Zealand are Raoul and Campbell.

Q3

(a) How many blocks are required for the daily climate summaries for the year 2020? What about the year 2010? What are the individual block sizes for the year 2010?

Based on these results, is it possible for Spark to load and apply transformations in parallel for the year 2020? What about the year 2010?

(b) Load and count the number of observations in daily for each of the years 2015 and 2020. How many tasks were executed by each stage of each job? Did the number of tasks executed correspond. to the number of blocks in each input?

(c) Load and count the number of observations in daily between 2015 to 2020 (inclusive). Note that you can use any regular expressions in the path argument of the read command. Now how many tasks were executed by each stage, and how does this number correspond to your input? Explain how Spark partitions input files that are compressed.

(d) Based on parts (b) and (c), what level of parallelism can you achieve when loading and applying transformations to daily? Can you think of any way you could increase this level of parallelism either in Spark or by additional preprocessing?

Number of blocks required for the daily climate summaries for the year 2020 is 1.

Number of blocks required for the daily climate summaries for the year 2010 are 2.

Individual block sizes for the year 2010 are 134217728 and 97862871.

The transformations can't be applied parallel for year 2020 as there is 1 block. So, data will go on worker node, not more than 1. However, in the case of 2010, there are two blocks so parallel execution might be possible.

Number of observations for year 2015 is 34899014.

Number of observations for year 2020 is 5215365.

Number of tasks executed by each stage of each job is 1. The number of tasks executed doesn't corresponds to the number of blocks in each output. For the year 2015, there are 2 blocks, 2 stages and 2 tasks but for the year 2020, there is 1 block with 2 stages and 2 tasks.

Number of observations in daily between the year 2015 and 2020 are 178918901.There are 2 stages. In stage 1, there are 6 tasks and in stage 2, there is only 1 task. This is because there are six years 2015, 2016, 2017, 2018, 2019 and 2020. In stage one, there are 6 tasks and for stage two, to count the number of observations, there is 1 task. Spark gives one block to each compressed input file, it doesn't partition. However, if the compressed file is bigger than the block size (128MB), it gives and additional block. Even the compressed file, each block could not handle more than 128MB.

To perform parallism, 4 cores were used. So, the level of parallism is 4. It means that 4 blocks were executed at a time parallelly to perform the transformation. This parallism could be increased if we increase the cores in Spark. Also, pre- processing can be performed that propagate the RDDs partitions by using joins, reduceBy or groupBy functions.

Q4

(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 never try to cache or collect all of daily.

(b) Filter daily using the filter 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 TMAX and TMIN, but do not necessarily report them simultaneously due to issues with data collection or coverage. Determine how many observations of TMIN do not have a corresponding observation of TMAX. How many different stations contributed to these observations?

(d) 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? Use hdfs dfs -copyToLocal to copy the output from HDFS to your local home directory, and count the number of rows in the part files using the wc -l bash command. This should match the number of observations that you counted using Spark. Plot time series for TMIN and TMAX on the same axis for each station in New Zealand using Python, R, or any other programming language you know well. Also, plot the average time series for TMIN and TMAX for the entire country.

(e) Group the precipitation observations by year and country. Compute the average rainfall in each year for each country and save this result to your output directory. Which country has the highest average rainfall in a single year across the entire dataset? Is this result sensible? Is this result consistent? Find an elegant way to plot the average rainfall for each country using Python, R, or any other programming language you know well. There are many ways to do this in Python and R specifically, such as using a choropleth to color a map according to average rainfall.

Number of rows in daily are 2928664523.

After filtering daily using FILTER command, command to obtain the subset of observations containing the five core elements described in inventory, number of observations for each of the five core elements are as follows: -

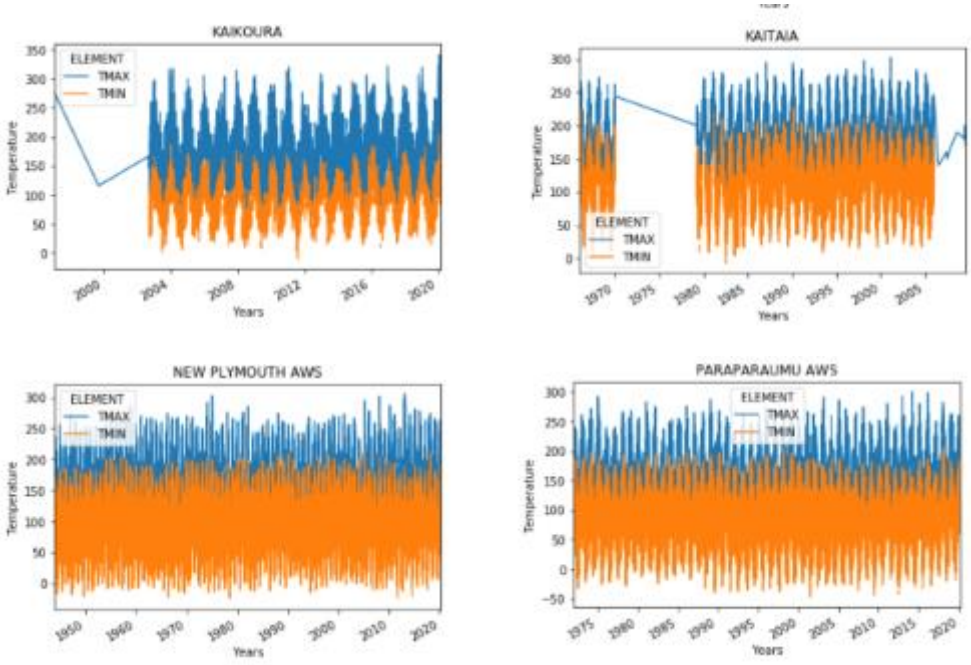
SNOW	332430532
SNWD	283572167
PRCP	1021682210
TMIX	435296249
TMAX	436709350

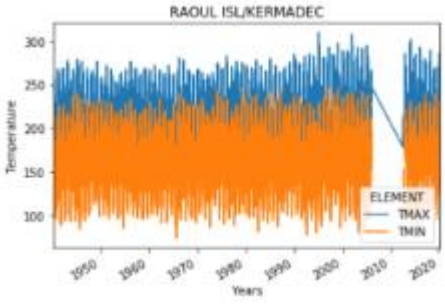
It is obvious that element having the most observations is PRCP.

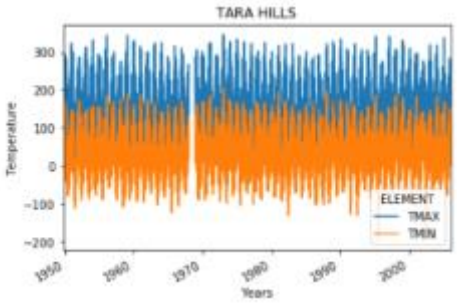
There can be many stations that collect TMAX and TMIN, but do not necessarily report them simultaneously due to issues with data collection or coverage. Number of observations of TMIN that do not having a corresponding observation of TMAX are 8428801 Number of stations contributing to these observations are 27526.

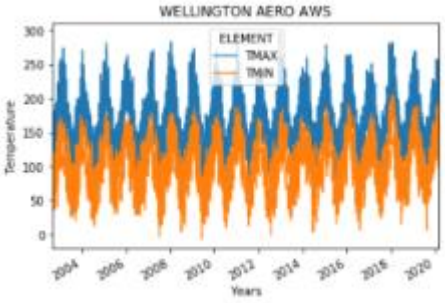
After filtering daily to obtain all observations of TMIN and TMAX for all stations in New Zealand and saving the result to the output directory, number of observations are 458892 and the number of years covered are 81.The same number is obtained by copying output from HDFS to local home directory.

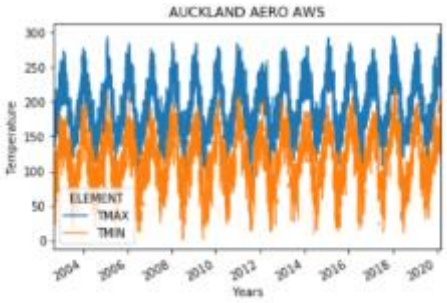
Time series plot for each station in New Zealand is as follows: -

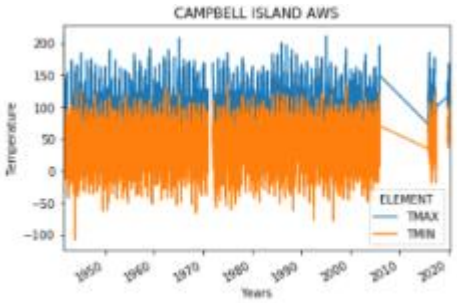


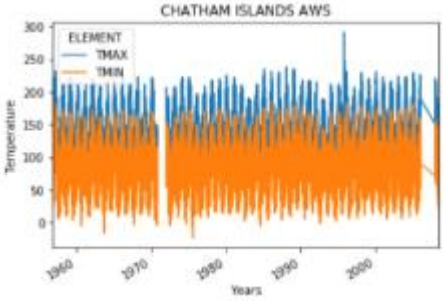

RAOUL ISL/KERMADEC
Temperature
Years
ELEMENT
TMAX
TMIN

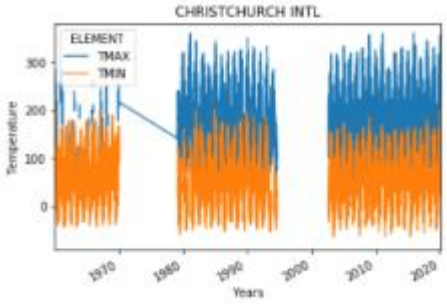

TARA HILLS
Temperature
Years
ELEMENT
TMAX
TMIN

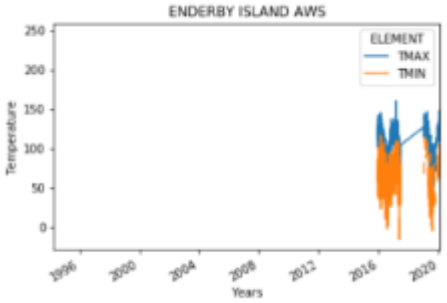

WELLINGTON AERO AWS
Temperature
Years
ELEMENT
TMAX
TMIN

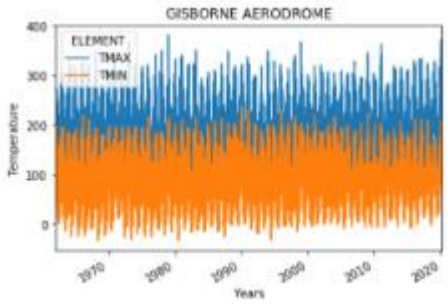

AUCKLAND AERO AWS
Temperature
Years
ELEMENT
TMAX
TMIN

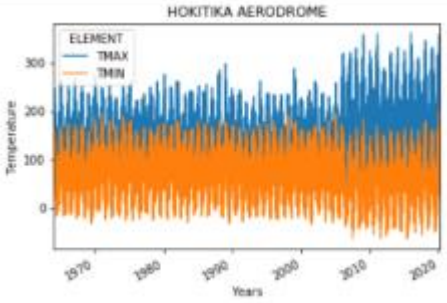

CAMPBELL ISLAND AWS
Temperature
Years
ELEMENT
TMAX
TMIN

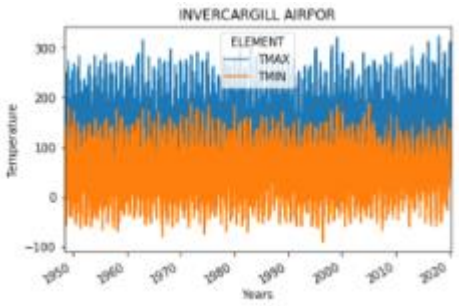

CHATHAM ISLANDS AWS
Temperature
Years
ELEMENT
TMAX
TMIN


CHRISTCHURCH INTL
Temperature
Years
ELEMENT
TMAX
TMIN

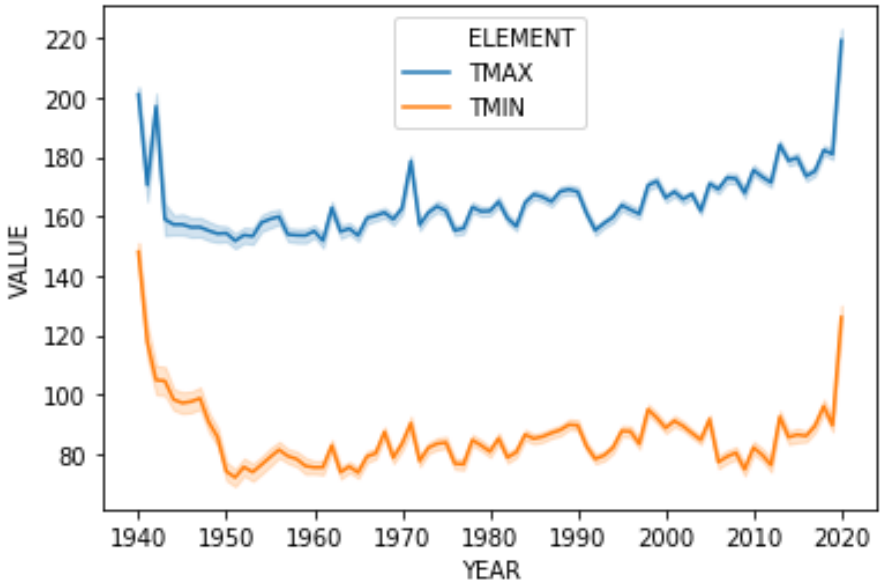

ENDERBY ISLAND AWS
Temperature
Years
ELEMENT
TMAX
TMIN


GISBORNE AERODROME
Temperature
Years
ELEMENT
TMAX
TMIN


HOKITIKA AERODROME
Temperature
Years
ELEMENT
TMAX
TMIN

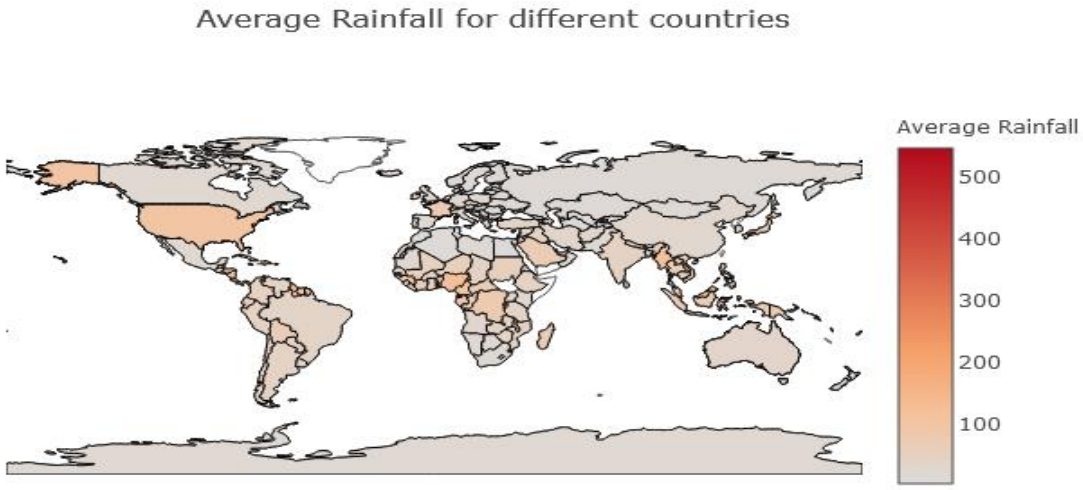


Time series plot for average time series for all stations is as follows: -



After grouping the precipitation observations by year and country, the average rainfall in each year for each country was calculated. Country having the highest average rainfall in a single year across the entire dataset is Equatorial Guinea which was 4361.0. This data is however not consistent as the data in daily is updated every day, so there are chances that the country with highest average rainfall may change is the future and also this is the average, it means the data may or may not be same throughout the year, there may be some outliers that is giving this type of result. Hence, we can't say that the data is sensible.

Choropleth map is chosen to map countries according to average rainfall as it is a convenient visualisation technique. The code is executed in python. Plotly is used to construct the map.


Average Rainfall for different countries
Average Rainfall
500
400
300
200
100