



Data Analytics

Greenland warming last 140 years

Mario JANUS

September, 2022

Table of content

- 1.Introduction
2. Data collection
3. Data cleaning and EDA
- 4.Comparison of databases
5. ERD
- 6.Database creation
7. Insights

Introduction

Originally, I planned to analyze the risks of desertification in countries and I found a really good data source with a lot of atmospheric parameters (wind, rainfall, clouds, temperature, vegetation etc.) at <https://apps.ecmwf.int/datasets/data/era20c-daily/>. Unfortunately, the website is currently doing a database move till October, so it was not possible to request any data from this website. So therefore, my topic for the final project is the analyzing of Greenland warming over the last 140 years.

As everybody knows the global warming is big challenge for current and future generations. The warming of Greenland plays a big role in this context, as it is one of the biggest reservoirs of frozen water on land (except Antarctica). The melting of this water is already increasing the sea level and the sea level has already been rising for 21cm since 1900. The increase in temperature in Greenland could or will accelerate the sea level rise even further. Should all ice in Greenland melt, the global sea level will increase by another 3 meters from this ice alone. This will have a big impact on coastal areas and the infrastructure there. Especially Asian countries are affected a lot by this, because a lot of big cities and industrial centers are located in coastal areas. To fight the sea level rise it will need a lot of investments to harness and protect this city from the sea level rise as dams and flood doors will need to be built to protect this city. Also, small islands in the pacific are affected a lot by the sea level rise, as most of the small islands are just a few meters over the current sea floor and they are getting more prone to flooding by storms (especially hurricanes). The flooding of this islands causes also emigration issues, as this people lose their homes due the flooding and this is already happening on some islands there, who have to abandon their home forever due the sea level rising. My goal with this project is to analyze the warming in Greenland and also predict the temperature increase in future, as rising temperature will accelerate the melting of the ice mass.

Data and data sources

The data was taken from the website <https://crudata.uea.ac.uk/cru/data/greenland/>. The data provider is Danish Meteorological Institute. The data source locations for the temperatures are Nuuk and Ilulissat on the west coast and Qaqortoq on the south coast. This data contains 3 data sets from three locations in Greenland that reaches back till 1784 from historical year books and ends in 2013. The data comes as .dat file and the scientists used some regression model on the average monthly temperature. Therefore, the monthly average temperature in the data set is not represented by real temperature values, but the encoded temperature will still give insight of the climate change in Greenland. To estimate the true mean monthly temperature from the thrice daily observations carried out before the introduction of the synoptic stations, the scientist used a weighted average of the observations and they calculated the monthly mean temperatures (T_m) in the Yearbooks with the following formula:

$$T_m = (2 * T_8 + 2 * T_{14} + 5 * T_{21}) / 9$$

where T_8 , T_{14} and T_{21} are the monthly mean temperatures measured at 08:00, 14:00 and 21:00 o'clock local time respectively.

Each dataset consists of the following 13 columns in a text format:

Variable	Description
First column	Year of observation
Second column	Temperature from January
Third column	Temperature from February
Fourth column	Temperature from March
Fifth column	Temperature from April
Sixth column	Temperature from May
Seventh column	Temperature from June
Eight column	Temperature from July
Ninth column	Temperature from August
Tenth column	Temperature from September
Twelfth column	Temperature from October
Eleventh column	Temperature from November
Thirteenth column	Temperature from December

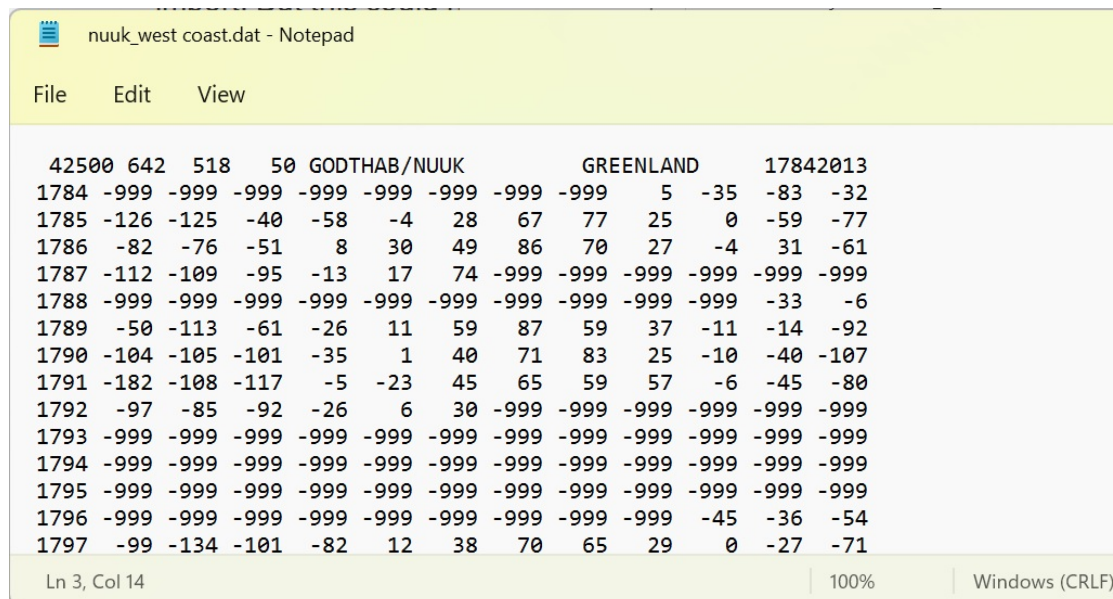
Data collection

The data was freely available and the data sets could be simply downloaded via a link. Therefore, no web-scraping or API requests were needed in this case to collect the data. The data sets were stored as a .dat file on the PC.

Data cleaning and Exploratory data analysis

After opening the files with a text editor (e.g. Notepad), I could see that the temperatures were not stored as real temperatures. After a little research I found out, that the scientists used some regression model on the monthly average temperatures to fill daily gaps to calculate the monthly average. But the numbers are still based on real temperatures and should show trends and patterns. I also tried to figure out how or if the regressed values could be reverted to normal temperatures, but unfortunately, I had no success to find any information about this. One challenge was, to import the data into a Pandas data-frame as the values were out of order after the import. But this could have been resolved after I removed the first line from each data set, as it was only a short description line and then I saved the file just as a simple .txt file. After the import into a Pandas data-frame, I added column names for the year and the 12 months in Python, because the datasets had only columns of numbers, but no headers and the column names make it easier to analyze the data.

Raw Data:



42500	642	518	50	GODTHAB/NUUK				GREENLAND				17842013
1784	-999	-999	-999	-999	-999	-999	-999	-999	5	-35	-83	-32
1785	-126	-125	-40	-58	-4	28	67	77	25	0	-59	-77
1786	-82	-76	-51	8	30	49	86	70	27	-4	31	-61
1787	-112	-109	-95	-13	17	74	-999	-999	-999	-999	-999	-999
1788	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-33	-6
1789	-50	-113	-61	-26	11	59	87	59	37	-11	-14	-92
1790	-104	-105	-101	-35	1	40	71	83	25	-10	-40	-107
1791	-182	-108	-117	-5	-23	45	65	59	57	-6	-45	-80
1792	-97	-85	-92	-26	6	30	-999	-999	-999	-999	-999	-999
1793	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
1794	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
1795	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999
1796	-999	-999	-999	-999	-999	-999	-999	-999	-999	-45	-36	-54
1797	-99	-134	-101	-82	12	38	70	65	29	0	-27	-71

Data-frame:

```
In [34]: #Adding column names to data set as the data sets had no column header (Year and Months)
west1.columns = ['Year', 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sept', 'Oct', 'Nov', 'Dez']
west2.columns = ['Year', 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sept', 'Oct', 'Nov', 'Dez']
south1.columns = ['Year', 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sept', 'Oct', 'Nov', 'Dez']
west1.head()
```

```
Out[34]:
```

	Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sept	Oct	Nov	Dez
0	1873	-121	-76	-98	-24	26	37	74	56	24	-21	-32	-96
1	1874	-131	-86	-78	-46	19	36	60	58	25	-32	-36	-48
2	1875	-66	-80	-121	-23	-8	30	63	54	33	-6	-13	-61
3	1876	-111	-92	-75	-31	7	42	59	61	38	0	-35	-46
4	1877	-145	-96	-68	-30	18	47	61	77	55	-14	-58	-85

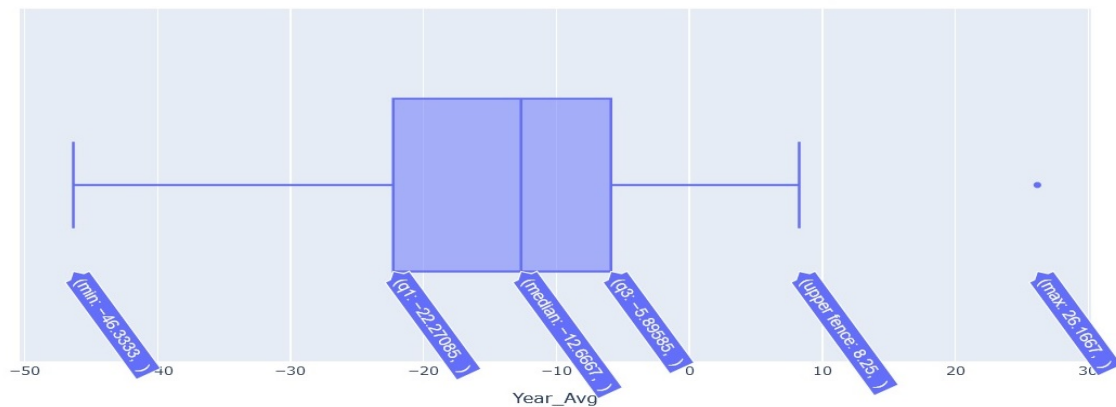
On further analysis I realized that the data from 1784 till 1872 contained a lot of “empty” values, which were represented by the value -999. This is probably attributed to the fact that data recording was only sporadically done in the 1700s and 1800s century and not periodically.

I therefore decided to split the data-frames into new frames, only taking the values from 1873 on-wards, as the data from 1873 to 2013 (140 years) is complete and contains no empty values at all. I also checked the data

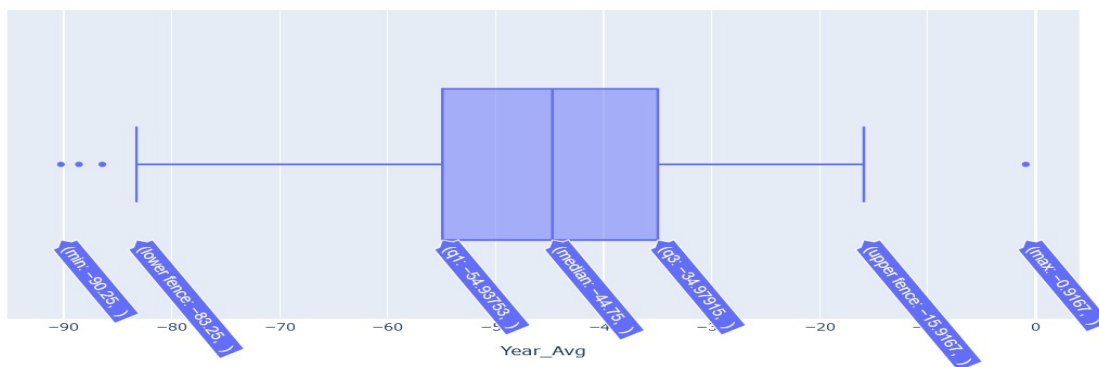
types and all columns were saved as integer values. For the beginning I also kept the year column as integer, as it was complicated to convert it into a date-time format, because the year column only contained the year value. This made it also easier to plot the frames and the visualization would be clearer. Later I converted the year to date-time format as it was needed for forecasting.

I also checked the data for outliers and fortunately the data had only 1 to 4 outliers per site. As the temperatures were taken with a methodical method by scientists, I decided to keep this outlier, as they are part of the natural climate fluctuation.

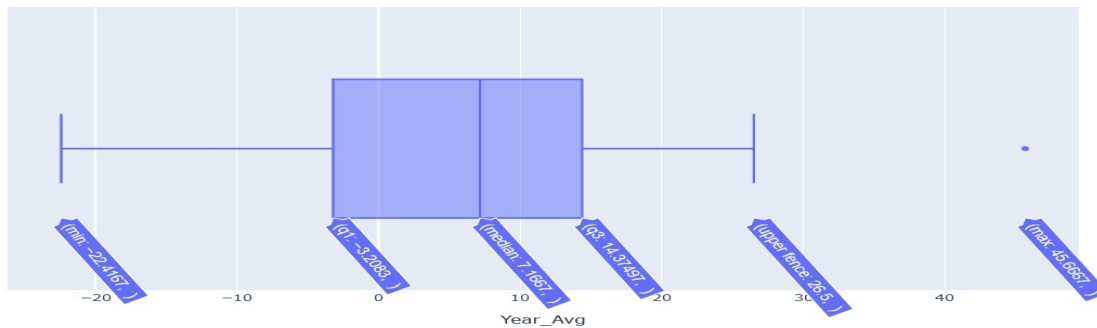
Nuuk - West Coast



Ilulissat - West Coast



Qaqortoq - South Coast



I also analyzed if the datasets are stationary or not and for this I used the Augmented Dickey-Fuller and KPSS test:

ADF Statistic: -7.212436879191608

p-value: 2.2168600582803282e-10

Critical Values:

1%, -3.4779446621720114

Critical Values:

5%, -2.8824156122448983

Critical Values:

10%, -2.577901887755102

KPSS Statistic: 0.637281

p-value: 0.019247

Critical Values:

10%, 0.347

Critical Values:

5%, 0.463

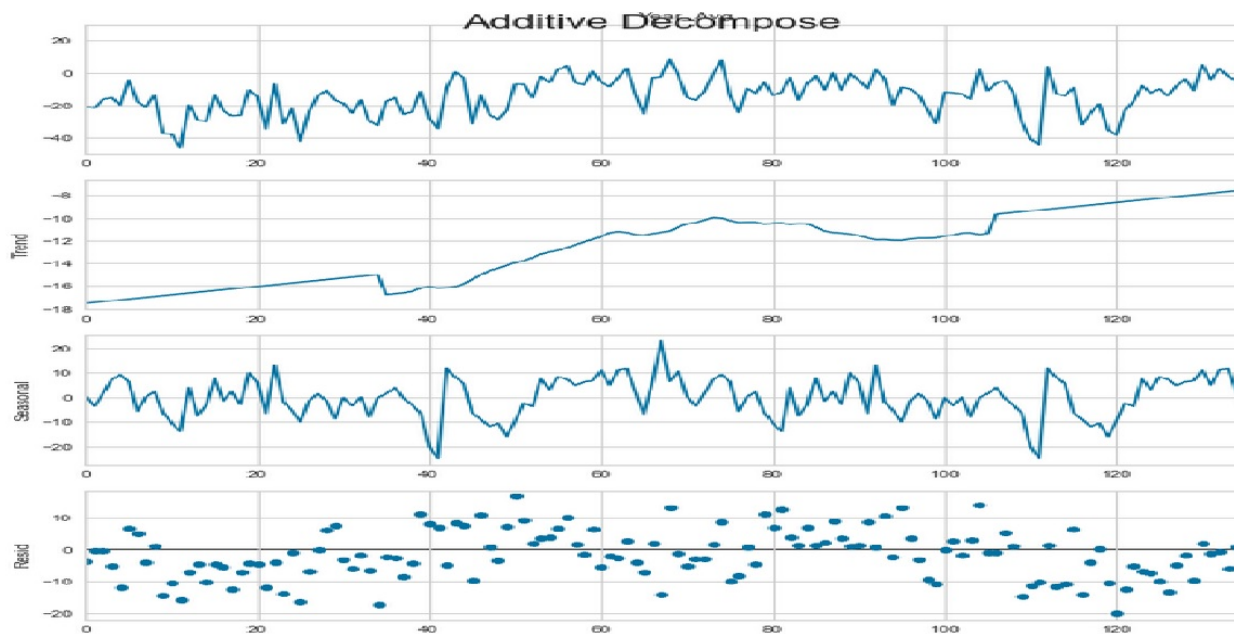
Critical Values:

2.5%, 0.574

Critical Values:

1%, 0.739

ADF test shows, that the data is stationary, because the p-value is very close to zero and far away from 0.05 and therefore the Null-Hypothesis of ADF can be rejected. The Null-Hypothesis here is, that the data is non-stationary. The KPSS test shows the contrary, because here the Null-Hypothesis is, that the data is stationary and the p-value is also under 0.05. But because the ADF test shows a very low p-value, we can assume that data is indeed stationary. I also performed a seasonal decompose:



As you can see, the seasonal section shows no seasonal pattern over the last 140 years, but there is a visible upward trend and this will probably continue in the future.

Database selection

To use the data for future use, I will store this data also in a database. There are two kinds of databases, flat file database and structured databases.

For the project I will choose a structured database, as I have 3 different datasets and flat file databases store data only in a single text file. A relational or structured database offers more flexibility, because queries can be done between different tables/datasets and it offers also the possibility to alternate data, as it uses a Structured Query Language (SQL). A relational database management system is one of four common types of systems you can use to manage your business data. The other three include:

- hierarchical database systems

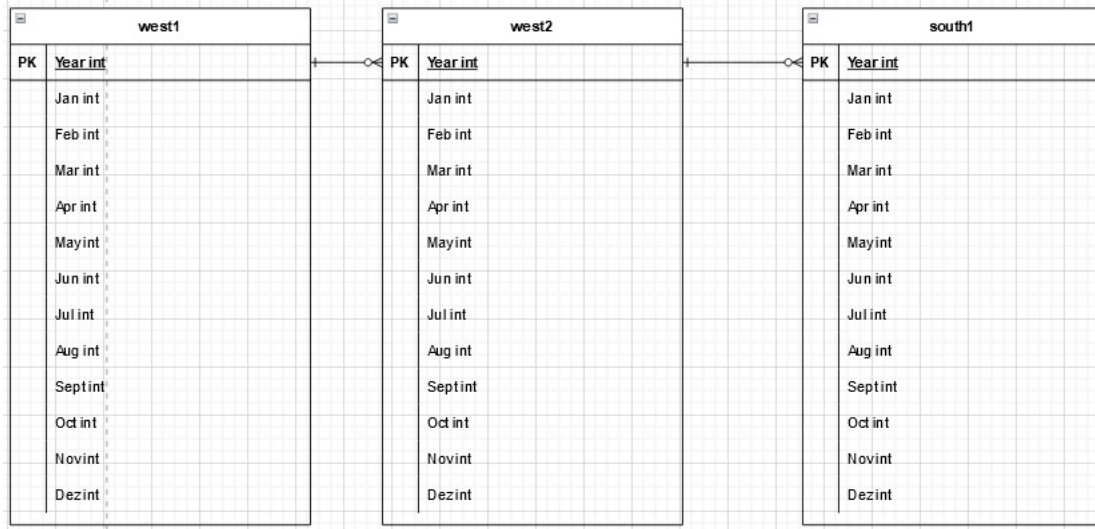
- rarely used anymore, was more used in the beginning of computing
- network database systems
 - A network database is a type of database model wherein multiple member records or files can be linked to multiple owner files and vice versa. The model can be viewed as an upside-down tree where each member information is the branch linked to the owner, which is the bottom of the tree. Essentially, relationships are in a net-like form where a single element can point to multiple data elements and can itself be pointed to by multiple data elements.
- object-oriented database systems
 - An object-oriented database is a database that subscribes to a model with information represented by objects. Object-oriented databases are a niche offering in the relational database management system (RDBMS) field and are not as successful or well-known as mainstream database engines

For this project I have chosen the relationship database MySQL as it is open source and free to use. Other well-known related databases are Oracle, R, SAP and Microsoft SQL

Entities. ERD

To describe the database I created an entity diagram. Because my dataset doesn't contain a lot of parameters (only year and temperatures), this entity diagram is very simple.

In order to compensate this a bit, I used Python to import my data set to MySQL and did some SQL queries within Python to plot some graphs with variables that have been pulled via SELECT queries in Python.



Creation of the database and data importation

After I created the entity diagram, I created the database in MySQL via Workbench which I named greenland. At first I tried to create the tables via Python, but I ran into some problems here, because the tables were not created and I always received some error messages. So I decided to create the tables also via Workbench. I named the tables west1 (Nuuk location), west2 (Ilulissat) and south1 (Qaqortoq) and I used this simple name for the tables, because the location names would be too cumbersome for SQL queries. When I tried to import the data-frames for the data sets via Python, I ran into a problem, because during importing it always tried to push an additional index column from the data-frames.

In order to solve this, I created a temporary column with the special format name `index`, because normally it is not possible to create a column with the name index, as this name is normally reserved by MySQL. After importing the data-frames, the temporary index columns have been deleted.

Creating tables:

```
use greenland;  
CREATE TABLE west1 (`index` int,Year int,Jan int,Feb int,Mar int,Apr int,May int,  
Jun int,Jul int,Aug int,Sept int,Oct int,Nov int,Dez int);
```

Importing data-frames with Python:

```
In [289]: from sqlalchemy import create_engine  
import mysql.connector as sqlc  
engine = create_engine("mysql+pymysql://{user}:{pw}@localhost/{db}"  
                        .format(user="root",  
                                pw="19225360",  
                                db="greenland"))  
west1.to_sql('west1',con = engine, if_exists = 'append', chunksize = 1000)  
west2.to_sql('west2',con = engine, if_exists = 'append', chunksize = 1000)  
south1.to_sql('south1',con = engine, if_exists = 'append', chunksize = 1000)
```

Deleting temporary index column in table with Python:

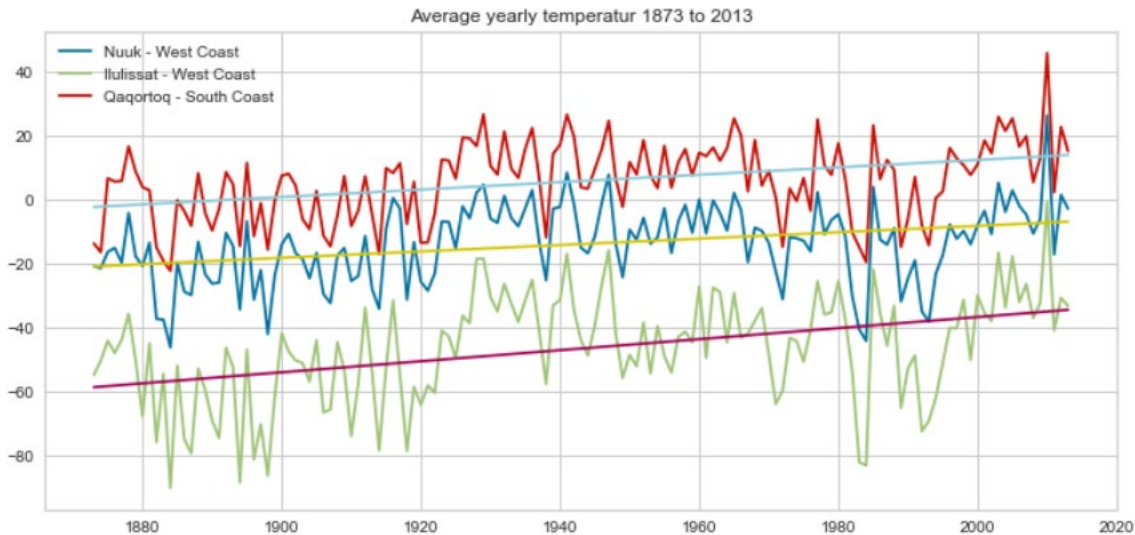
```
import MySQLdb  
  
db= MySQLdb.connect("localhost", "root", "19225360", "greenland")  
  
cursor= db.cursor()  
  
cursor.execute("ALTER TABLE west1 DROP `index`")  
cursor.execute("ALTER TABLE west2 DROP `index`")  
cursor.execute("ALTER TABLE south1 DROP `index`")  
  
db.close()
```

Insights

I also performed some SQL queries in Python and plotted them to give some insight of the temperature over the last 140 years in Greenland. Here I calculated the yearly average over the last 140 year with SQL for all 3 sites and plotted them.

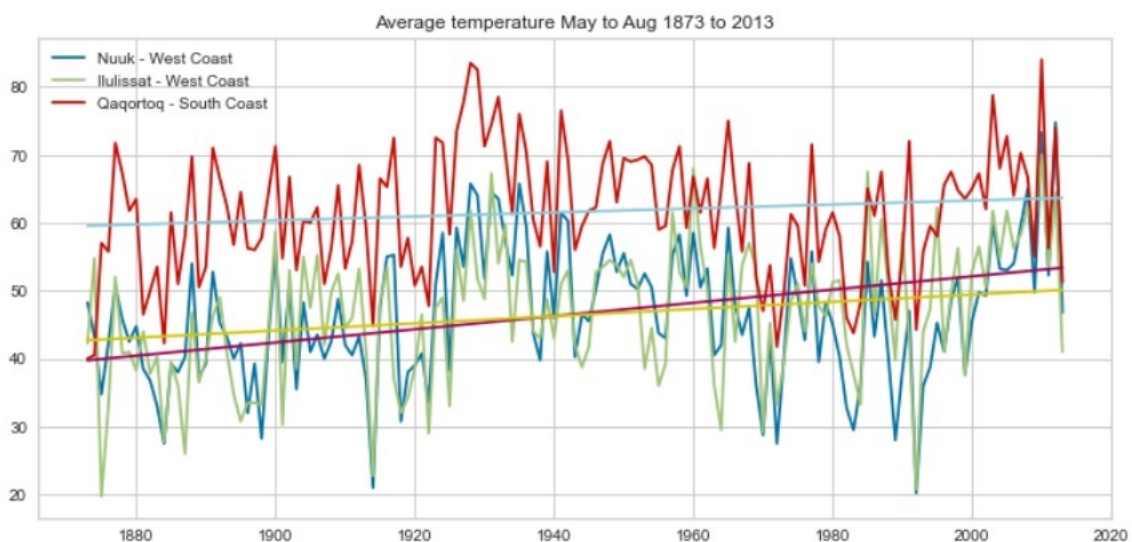
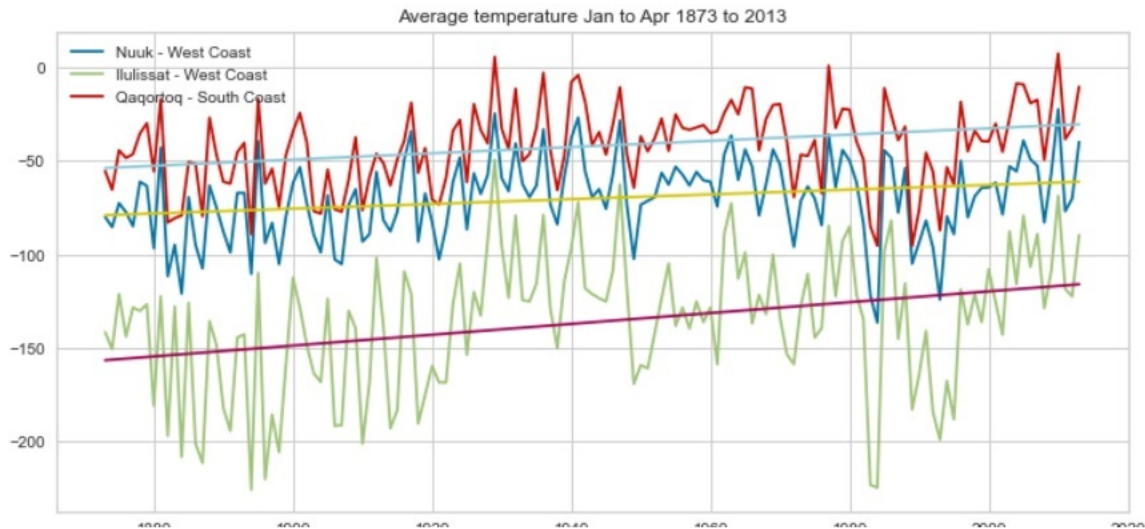
In [11]: *#Selecting average temperature per year from all 3 sites via SQL*

```
w1_year_avg_q="""SELECT Year, SUM(Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sept+Oct+Nov+Dez)/12 as Year_Avg
FROM west1 GROUP BY Year;"""
w2_year_avg_q="""SELECT Year, SUM(Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sept+Oct+Nov+Dez)/12 as Year_Avg
FROM west2 GROUP BY Year;"""
s1_year_avg_q="""SELECT Year, SUM(Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sept+Oct+Nov+Dez)/12 as Year_Avg
FROM south1 GROUP BY Year;"""
w1_year_avg=pd.read_sql(w1_year_avg_q,my_conn)
w2_year_avg=pd.read_sql(w2_year_avg_q,my_conn)
s1_year_avg=pd.read_sql(s1_year_avg_q,my_conn)
```



As you can see from the linear regression line, the temperature increased slowly over the decades. The south is warmer as the temperatures are generally warmer in the south and also the Gulf stream plays maybe a role. The Ilulissat site is the coldest site of the 3 sites, as it lays in a protected bay which is more inland. The other sites are directly at the coast. If you look closely, then you will see two big drops in the early 1980s and 1990s. These drops are probably related to the eruption of Mount St. Helen (in the north of USA) in 1980 and Pinatubo (Philippines) in 1991. These big eruptions blew a huge amount of sulfur dioxide in the atmosphere, which has a cooling effect on the climate. I also looked at the season average over the last 140 years with a SQL queries. For winter I took the months Jan to Apr as this are the coldest months and the winter is longer in the arctic circle. For summer I chose May till Aug. This is the query and the results:

```
In [35]: w1_JanApr_avg_q="""SELECT Year, SUM(Jan+Feb+Mar+Apr)/4 as JanApr_Avg
        FROM west1 GROUP BY Year;"""
        w2_JanApr_avg_q="""SELECT Year, SUM(Jan+Feb+Mar+Apr)/4 as JanApr_Avg
        FROM west2 GROUP BY Year;"""
        s1_JanApr_avg_q="""SELECT Year, SUM(Jan+Feb+Mar+Apr)/4 as JanApr_Avg
        FROM south1 GROUP BY Year;"""
        w1_JanApr_avg=pd.read_sql(w1_JanApr_avg_q,my_conn)
        w2_JanApr_avg=pd.read_sql(w2_JanApr_avg_q,my_conn)|
        s1_JanApr_avg=pd.read_sql(s1_JanApr_avg_q,my_conn)
```



As

you can see the winter (Jan to Apr) shows nearly the same pattern as the yearly average for the 3 sites over the last 140 years.

The interesting thing is, that summer temperature in all 3 sites do move more closely together, especially since 1960 as seen in the second plot. Also the temperatures in the south increases more in summer then at the other 2 sites, so the glaciers there are at risk to melt faster.

Conclusion

The analysis shows that Greenland gets indeed more and more warm since the last 140 years. Especially the south is heating more than rest of Greenland and the ice melting there will probably accelerate in the future. The analysis also shows that big volcanic events can influence the temperature in Greenland, even when the eruptions happen 1000ths of kilometers away from Greenland.