2016

Analysis of Weather data

BIG DATA PROJECT | MSA-6500 | SPRING 2016

Submitted By-
Brian Keith Sigurdson, Sean Brigadier &
Pallavi Kalambe

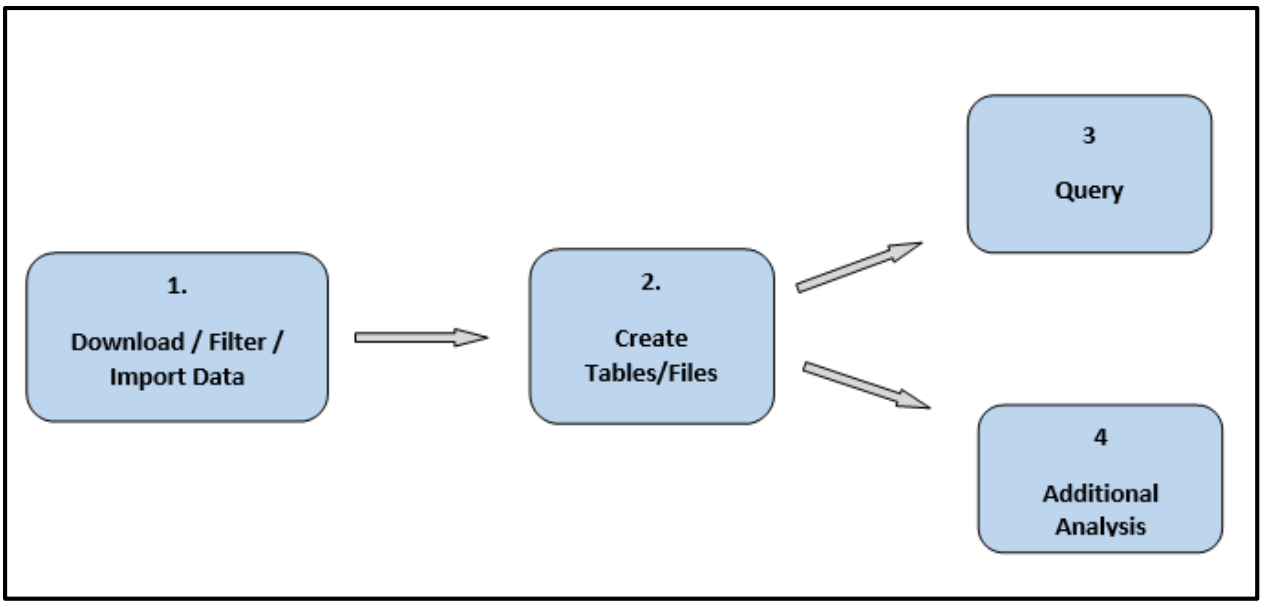
BGSU

Table of Contents

BGSU

PLANS FOR ANALYZING THE DATA

The following is a diagram that shows the process of this data project. For the two plans (A & B), we may use different tools or slightly different approaches, but should resemble this format. Our first choice is with plan A.



Given the constraints of time and our current skill level with all technologies involved with this project; we do not want to change certain steps, regardless of the chosen plan. For example, we feel that extracting and transforming of the data will be important under either plan; therefore step 1 is essentially the same under either approach. Likewise, given the time and our skill level with various tools, we have a finite number of tools from which to choose to perform the analysis in step 4, regardless of the plan followed. Therefore, steps 1 and 4 are very similar regardless of the chosen plan. Our plans, A and B, vary by which intermediate tool we use to query the data, and the tool's data repository requirements.

BGSU

Plans A and B

Step 1: Download / Filter/ Import Data

Overview: The initial data is an extract from the National Climatic Data Center. We can use the script given in this assignment to download the data by entering in the parameters (years) that we need. This step's primary purpose is to get the data from NOAA in the format ready to import into tables that we need for analysis.

Plan A and B Tools Used: Java is the preferred tools to transform and clean up the data, which will be used for the following tasks:

- Replace the symbol for missing values, '***', with an appropriate null value for the chosen repository in step 2 below.
- Filtering out values based on 12 month participation as given in the guidelines.
- Transforming the different value types, if and as needed.

Given the time frame of the project, Java is the desired tool, as a group member is already familiar with the language.

Plan A Potential Problems: There are several problems that we foresee possibly happening. The data may cause issues depending on the data format. We might find that there are many missing values to the available fields that will eventually cause issues when we do our data analysis and querying. There will also be data population issues since we are only supposed to use years that have data for all 12 months. Solving these issues may take time, but our initial try will be by using Java to get the data in the format we need before creating tables.

Plan B Potential Problems: The potential problems for Plan B are the same as A.

Step 2: Create Test/Prod Tables

Overview: We will import into HDFS, files with transformed data by year, and an accumulation file containing transformed data for all years.

Plan A Tools Used: We will use Hive to create the tables, as Hive appears to be the default table creation mechanism within the Hue technology provided in the Cloudera virtual environment. Hive tables can also be easily accessed by Impala within Hue. This is desirable, since we are all familiar with SQL, which should facilitate the use of HiveQL and or Impala, for questions that lend themselves to SQL tools.

We will create two structured tables: Test and Prod. The test data is a scaled down version of the full data set, such as for a single year, and may actually be several smaller tables for individual

years. These tables will be used for developing and testing queries before running them on the entire dataset. If the data extraction and transformation goes well in Step 1, we do not expect any issues creating tables in Hive.

Plan B Tools Used: Given the time constraints for the project, and our comfort level with SQL, we believe it is prudent to create a structured dataset on which to work. If Hive is unable to accommodate our needs, we may consider working with the files directly in HDFS, such as with Apache Pig discussed in the next step, or we may have to consider the use of some other repository, which would be a last resort and certainly not a first choice.

Plan A Potential Problems: The data formats may still be an issue even after the import. Although we've reviewed the field descriptions listed in the file ish-abbreviated.txt found on the NOAA/NCDC website, we still need to get a solid understanding of the meaning of each field, in order to facilitate writing the queries required in Step 3 below.

Another issue relates to the speed of the tools. In class, Hive and Impala had similar performance, but on our systems there is a dramatic difference in speed. As an example, Hive takes 30 seconds to run the same query, repeatedly, yet Impala runs the identical query in a second or two.

By consulting sources, such as TutorialsPoint.com's Hive tutorial (http://www.tutorialspoint.com/hive/hive_views_and_indexes.htm), we know that Hive supports the creation of indexes, but Impala under Cloudera does not, as found here (http://www.cloudera.com/documentation/archive/impala/2-x/2-1 x/topics/impala_faq.html#faq_features_unique_1__faq_unsupported_unique_1). Therefore, it is possible that with the use of indexes, Hive's speed could exceed Impala on larger tables, and therefore make Hive more desirable.

Concerning Impala and HiveQL, the SQL supported under Impala is not as feature rich as that of HiveQL, see (http://www.cloudera.com/documentation/enterprise/5-2-x/topics/impala_langref_unsupported.html#langref_hiveql_unsupported_unique_1). HiveQL also appears to support a more sophisticated data structures such as arrays, maps, and struct. A brief description can be found here (http://www.tutorialspoint.com/hive/hive_data_types.htm).

Lastly, however useful Cloudera's virtual environment is as a learning tool, each group member's laptop struggles under the weight of the memory requirements and administrative overhead when running the Cloudera virtual environment. Simple queries on test tables such as the zip codes table for PS4 are quite slow at times. This may require us to consider the limited use of third party tools, such as SQL Server Management Studio, to facilitate the development and frequent running of queries on test tables. We do not foresee any problems running the final version of queries on the production table residing on a system running the course's Cloudera virtual environment.

Plan B Potential Problems: This is the same as A, however, we may have more issues using other tools.

Step 3: Query

Overview: The purpose of this step is to solve questions that are being asked in tasks 2, 3, and 4. Since the data is cleaned and structured, Hive and or Impala should be a viable option.

Plan A Tools Used: We will be using Hive/HiveQL and or Impala to do all queries against the data.

Plan B Tools Used: Plan B will use Apache Pig over Hadoop, or potentially the limited use of a third party tool as mentioned in Step 2 above, as Apache Pig suffers from some of the same performance issues as found with Hive and are elaborated on below. Scala could possibly be used instead of Apache Pig, but the group is less comfortable with Scala than with more SQL like environments, such as HiveQL, Impala, and Apache Pig's Pig Latin.

Plan A Potential Problems: We may find data issues within the data, or realize that the data needs some sort of transformation to make sense of the data. We will not know if this is an issue until we start exploring with the data. Understanding of the data may also be an issue if we can't interpret what certain values mean. Finally, one potential issue that we may find is the slowness of the query tool. As mentioned in Step 2: Create Test/Prod Tables above, Hive is generally slower than Impala on our systems, but Hive allows for indexes. HiveQL is also a more feature rich language than the SQL supported under Impala.

Plan B Potential Problems: As with Plan A, we may find issues within the data, or realize that the data needs some sort of transformation to make sense of the data.

Although Pig Latin is SQL like, some of the steps to using Pig may be unfamiliar to our group, and we may incur a learning curve by using this tool. If we go outside to SQL Server Management Studio, data size and format may be problematic.

One concern with Apache Pig is its rather sluggish performance when compared to HiveQL and especially when compared to Impala. The Cloudera virtualization environment seems to run slow on everyone's laptops, and initial Pig queries in class ran exceedingly slow. Depending on the queries we'd like to perform, Pig's performance may not be acceptable. We will not know if this is an issue until we start exploring with the data.

It is our understanding that Apache Pig can run on the local file system, recommended mainly for testing, which is outside of the virtualization environment and Hadoop. This might allow us to circumvent some of the potential performance issues discussed above, but at the cost of taking on the responsibility of installing and administering Pig ourselves. Given the constraints of time and skill level on this project, it is unlikely that we will pursue using Pig outside of Cloudera, but it is an option, and may need to be pursued if needed.

Step 4: Additional Analysis

Overview: The purpose of this step is to expand the work done in prior steps with potentially novel and more sophisticated data analysis techniques. As such, this analysis may include the use of outside tools and will not be limited to only the big data tools learned in the course.

We do not yet know exactly which or what additional analytic tools and techniques will be employed, but some ideas generated so far are as follows:

- Spatial and graphical analysis making use of longitude and latitude information.
- Statistical inference on temperature based on multiple regression analysis
- K-means cluster analysis on groups of weather stations

Plan A Tools Used: In addition to the many tools we've been introduced to in this course, we may additionally use RStudio, SAS, Microsoft Excel, or SparkR for analysis, and Tableau for data visualization. Tableau is a valuable tool for data visualization, and we are lucky to have two experienced users of Tableau in our group.

Plan B Tools Used: Plan B is identical to Plan A, because with the exception of SparkR or other tools introduced in this course, these are the tools with which we have the most comfort and experience. We will make use of their various strengths as needed throughout the course of the project. Introducing additional would likely be detrimental to our overall performance.

Plan A Potential Problems: Several problems may occur in this step. Until we know exactly what's in the data, we may not know exactly where things will possibly go wrong. We may not have enough data for trending analysis or have clean data. There may be a learning curve for some of the less familiar tools as well. After completing step 1, data extraction and translation, we will have a better idea of the limitation of various analytical techniques and or tools listed in the **Plan A Tools Used** section above.

Plan B Potential Problems: The potential problems for this step in Plan B are the same ones as listed in Plan A, for the reasons listed at the beginning of this document.

BGSU

WHICH PLAN DID WE CHOSE TO USE. (BRIEF OVERVIEW)

We followed Plan A with a few adjustments.

We used Java to extract and transform the data into the format that we needed to import into a Hive table and analyze with Impala. We planned to make use of Java and Impala for tasks 1 – 3, but were unsuccessful in getting a JDBC connection to Impala, regardless of several approaches show on the Cloudera website. In addition to the difficulties of connecting to Impala via JDBC, we found the virtual environment to be prohibitively slow in which to work, so we imported the weather data to MySql and later Postgresql databases as detailed below. The Python programming language was also used in tasks 1 – 3.

We were then able to answer questions by exporting data from the new data repository and importing it into tableau / RStudio for EDA analysis/visualizations and SAS Data Miner for clustering. The idea is to do cleanup work on the backend so that there was less work on the analysis side for data cleansing. The analysis was decided on after we had a better understanding of the data.

DETAILED STEPS ON EXECUTION AND ANALYSIS

Task 1 – Filtering:

The task of extracting, transforming, and filtering the data was accomplished with a Java program. The program took roughly 10-15 hours to do the following:

- Map each column position of the file to an appropriate field for the data, such as USAF = positions 1-6, MAX = 114-116, etc…
- Write and debug the Java code, including the development and use of a WeatherStation class
- Run the program

The program took roughly five minutes to run and the output was one filtered file for each year and on file appended for all year's information.

Had we known that the run-time would be so brief, the program could have been written so as to run it once for extraction and transformation, and a second time to filter the data, possibly requiring less development time.

Problems related to this task were minimal, and mainly normal debugging issues inherent in any programming project.

After transforming the missing data symbol, '*', to null values, represented as a null string object in Java, the file of all the year's data was roughly 500 megabytes.

Task 2 – Analysis 1:

The total time related to this portion was roughly 10-15 hours, although it is difficult to estimate accurately as the work was not always done sequentially. For example, the first three parts of this task could be answered via SQL queries, but the last question concerning the top 50 stations that operated the most consecutive years was more easily accomplished using a programming language. We used Python to connect to the Postgresql database and help answer the question of the top 50 stations.

As mentioned above, we did not stick with Impala for the data source and instead imported the data into a MySql database initially, but we eventually moved the data to a Postgresql database. MySql was initially chosen for its enhanced workbench GUI, but queries were too slow at times, even with the use of indexes, and stopping a query caused the server to crash, requiring a full reboot of the MySql server. Thankfully, such issued were not a problem with the Postgresql database.

The first three activities of this section were accomplished by the use of SQL queries and views. The last question concerning the ranking of top 50 stations by continuous operations was facilitated by the use of Python to connect to the database and write the results out to a comma separated value file.

There were no weather stations found operable for all years 1901-1940. The top 100 stations, ranked by their years of operability, albeit not continuously operable, are listed in Appendix A, with the top 10 listed in **Table 1** below.

Table 1 Ten of the top 100 stations ranked by continusous years

Count	USAF	Years Active
1	29110	17
2	29750	17
3	103380	15
4	108650	15
5	29170	14
6	106370	13
7	228020	13
8	28360	12
9	104190	12

10	101200	11

The second question related to the top 100 stations ranked by years active is provided in its entirety in Appendix B, and has the first 10 stations for the first 10 years listed in **Table 2** below.

Table 2 The first 10 stations of the first 10 years showing the years that these stations were operable

Year	14030	23610	28360	28750	28970	29110	29170	29350	29440	29700
1906				1			1		1	
1907				1			1		1	
1908				1			1		1	
1909				1			1		1	
1910				1			1		1	
1911				1			1		1	
1912				1			1		1	
1913				1			1		1	
1914				1			1		1	
1915				1			1		1	

The last part of this task asks for the top 50 stations ranked by consecutive years operable. The full list is detailed in Appendix C , with the first 10 stations displayed in **Table 3** below.

Table 3 First 10 of the top 50 stations ranked by maximum continuous operations

USAF	Max Duration
29750	15
29110	14
103380	14
108650	14
29170	12
106370	11
228020	11
101200	10
101270	10
103840	10

Task 3 – Analysis 2:

The total time related to this portion was roughly 10-15 hours, although it too is difficult to estimate accurately as the work was not always done sequentially. For example, the descriptive statistics, excluding the median, were generated by the use of SQL queries, and are provided below.

The median calculation initially made use of Numpy's Median function, but became prohibitively slow as the data in the later years grew larger. The median was then altered to make use of a sorted database query, taking into account whether the resulting relation had an even or odd number of rows, using Python, and are provided below.

The mean Python was used to generate the mean temperature values, by year, for the top 100 and 50 stations from task 2. The results were written to a text file to facilitate a simple linear regression comparison with the mean temperature values for the years 1901-1940, using the r programming language and RStudio, as shown below.

The descriptive statistics by year are detailed in Appendix D, with the first 10 years shown in **Table 4** below.

Table 4 Descriptive statistics for the first 10 years

Year	Min	Max	Mean	Median
1901	-28	89	40.41018	40
1902	-27	76	35.89901	36
1903	-23	84	40.68208	38
1904	-21	78	37.9962	38
1905	-27	83	39.79713	38
1906	-13	85	40.47534	38
1907	-31	83	37.71818	38
1908	-36	84	37.18481	36
1909	-36	82	37.29496	36
1910	-35	85	38.39961	37

The simple linear regression analysis for the top 50 stations, the top 100 stations, and all stations is shown in **figure 1** below.

Each data set is displayed over all active year for the respective instances. The top regression plot is for all year, while the second and third plots, the top 50 and top 100 respectively, are plotted for almost all years.

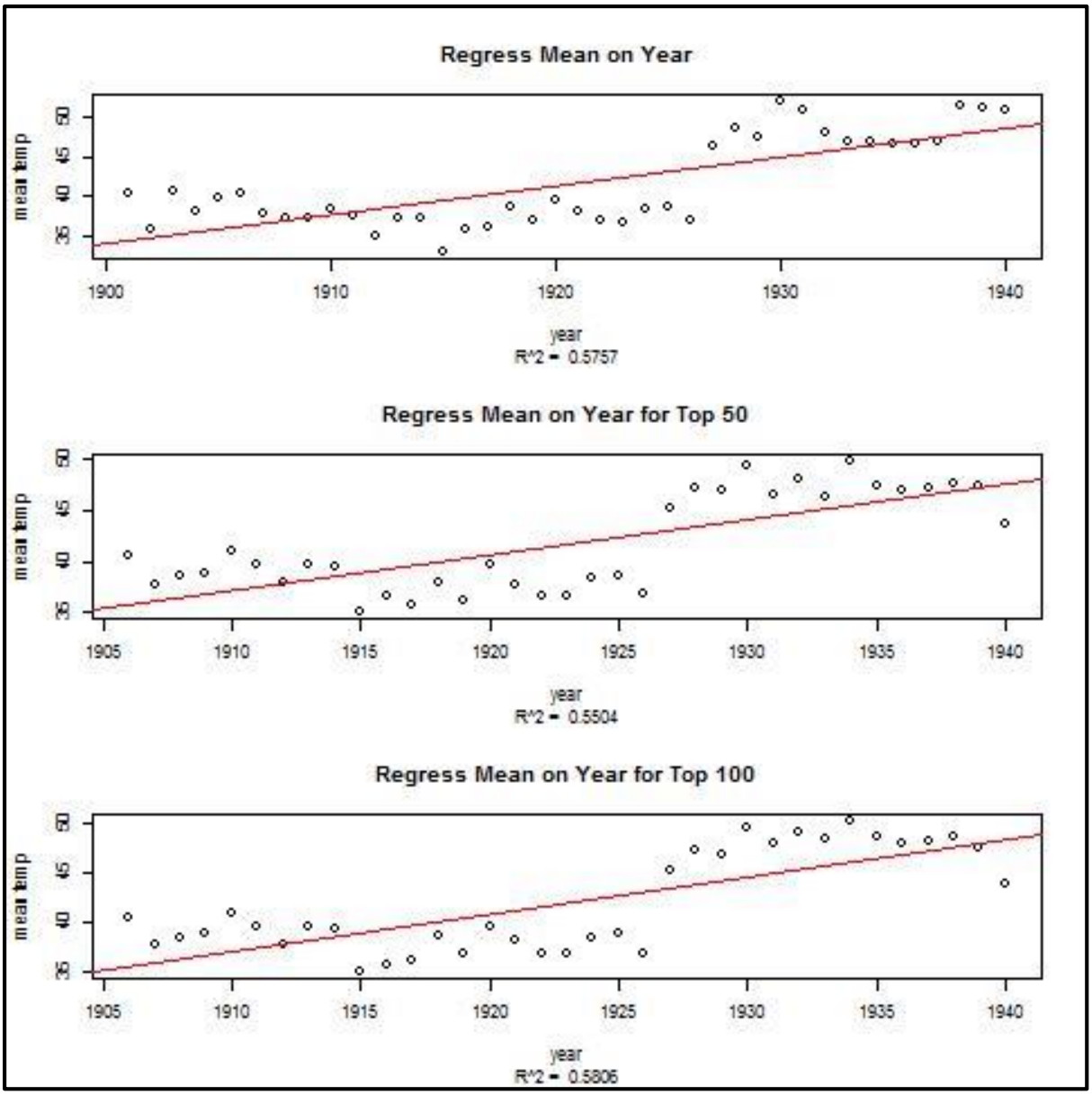


Figure 1 Simple linear regression for all years, top 50 stations, top 100 stations

The graph of Regression Mean on year has R^2 Value that explains 57% of variance of the temperature is explained by the model. Since there is only one predictor in the model, it shows a good fit for the data.

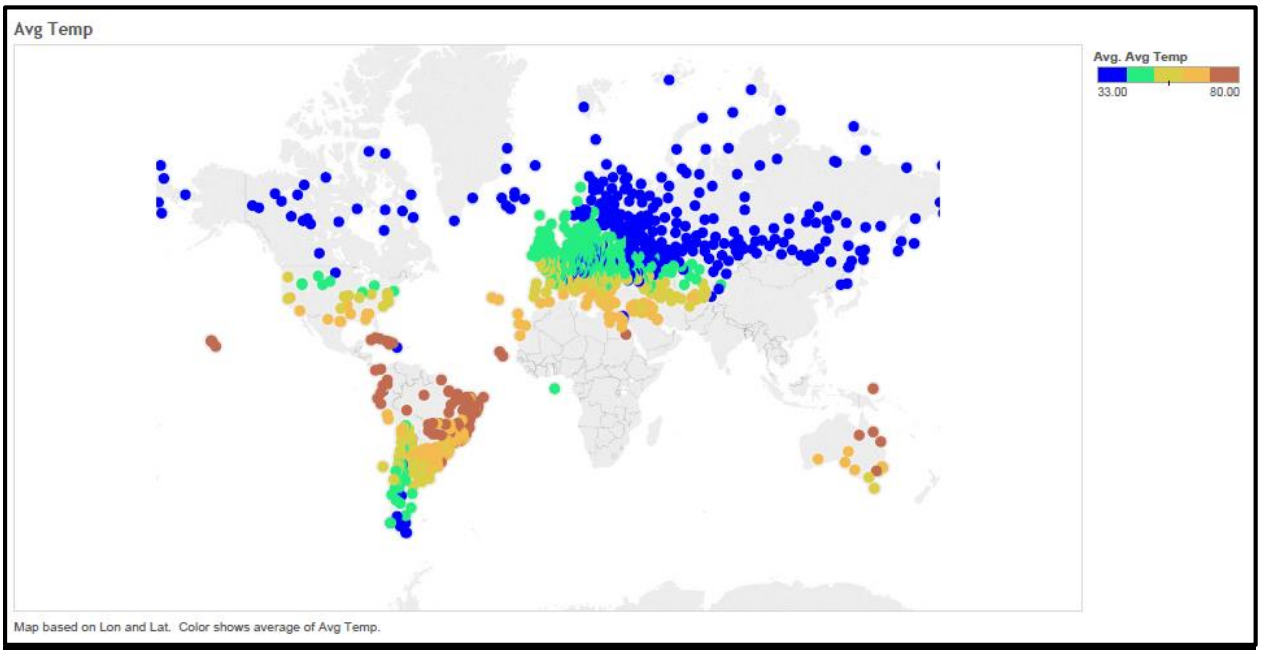
The graph of Regression Mean on year for top 50 shows R^2 Value that explains 55% of variance of the temperature is explained by the model. It shows a good fit for the data.

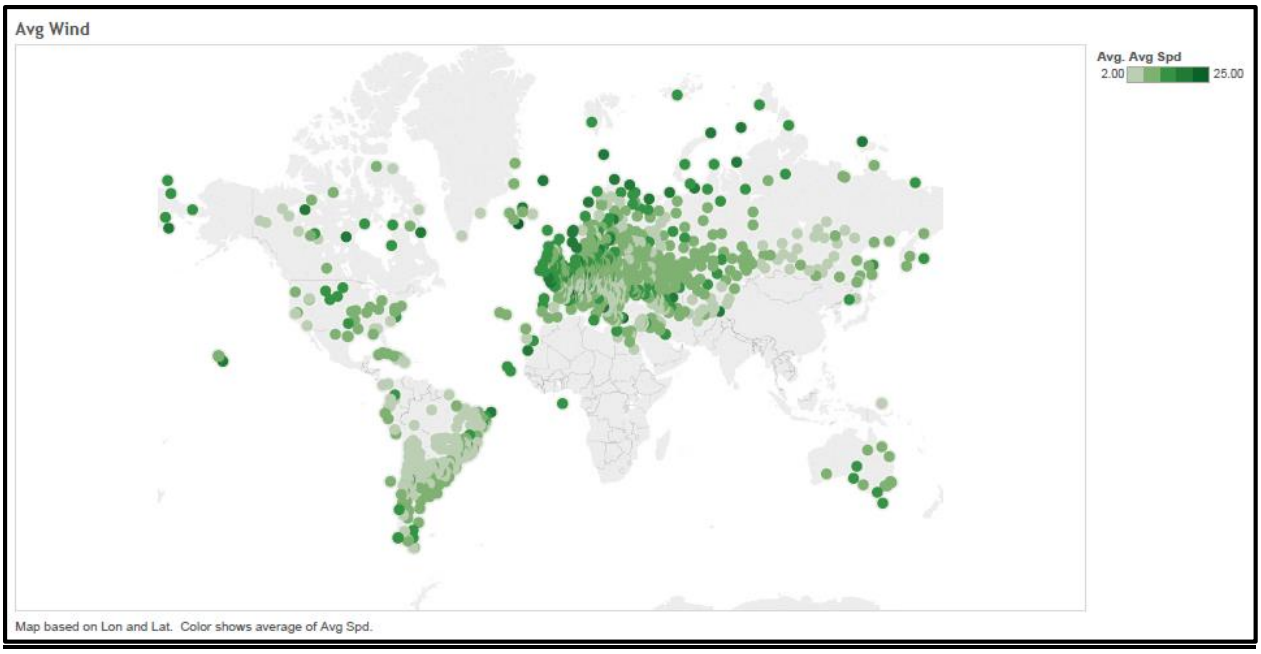
The graph of Regression Mean on year for top 100, shows R^2 Value that explains 58% of variance of the temperature is explained by the model. It also shows a good fit for the data.

Task 4

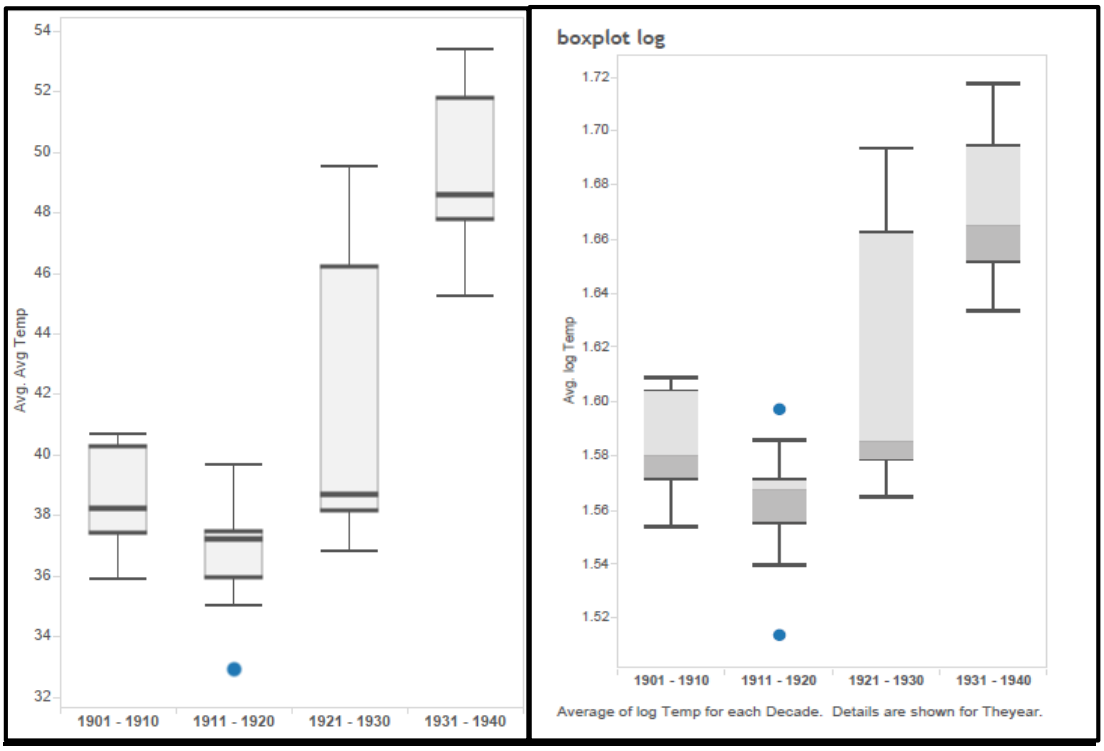
Exploratory Data Analysis

We did exploring with the data and wanted to do further research based on temperature and wind speed. We imported the data into Tableau to visually represent the average data temperature and wind speed. We adjusted the groups to better show the spread of warm/cold temperatures and hi/low wind speeds by location. The data is exactly as expected. The higher temperatures, in general, are near the equator and the wind speeds are near the coasts. There are exceptions of course, but most seem to follow this pattern (see graphs below).





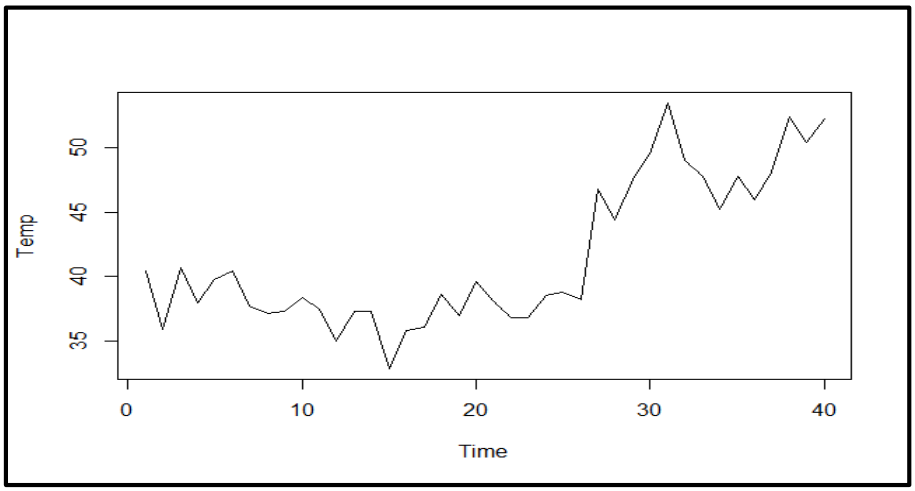
Since the data was as expected, the decision was made to explore the average temperatures by decade. We noticed a trend of increasing temperatures. To keep an even spread of temperatures by decade, we took the log of the data and replotted. With even spreads, the data does show increasing temperatures but at different rates (See graphs below).



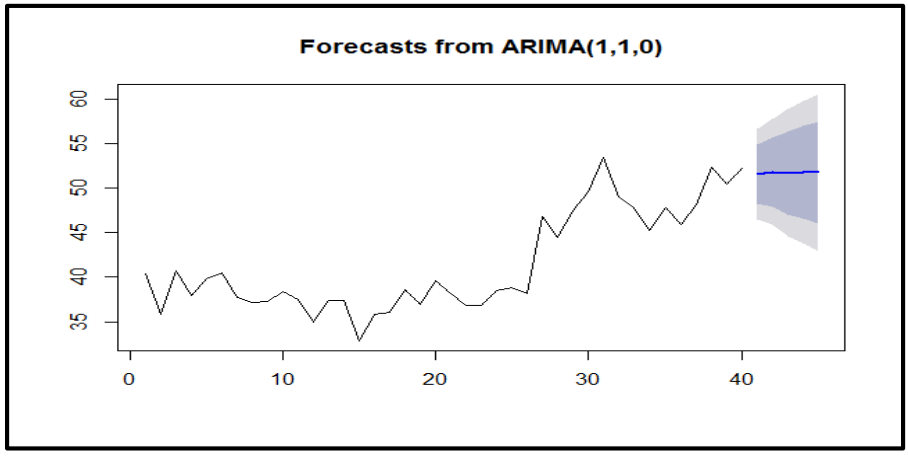
The data of increasing temperatures by decade does seem to show an interesting pattern. The data is increasing, but we do not know if we expect this trend to continue. We will do a Time Series prediction to forecast future temperatures.

Time Series Prediction over Global Temperature

We also did a time series prediction based on average yearly temperature by year. It was the decision to use average temperature due to the lack of data on many of the fields and the inconsistent availability of fields by year. The data was cleaned and plotted, where the Y-axis was temperature and the X-axis was time (by year).



Using the Dickey-Fuller Test, we determined that this is a non-stationary dataset. We graphed the ACF and PACF and found that the best fit(after a few trials) was ARIMA (1,1,0). The stationary condition was satisfied with both ARIMA (2,0,0) and ARIMA (1,1,0), and both passed assumptions. We compared the AIC for both models and found that ARIMA (1,1,0) was the better fit.



As you can see from the graph above, it is predicted that the temperature should remain relatively flat for the next 5 years. Further explanation of the codes used or the assumption tests are not explained in this paper since this not a Time Series project.

Data Mining

Since there are total 5237 observation with multiple USAF stations, which are operable for some specific period of time, analysis of the average temperature over all 40 year (1901 to 1940) might not give the accurate insight. For this purpose we divided the data in to parts on the basis of years.

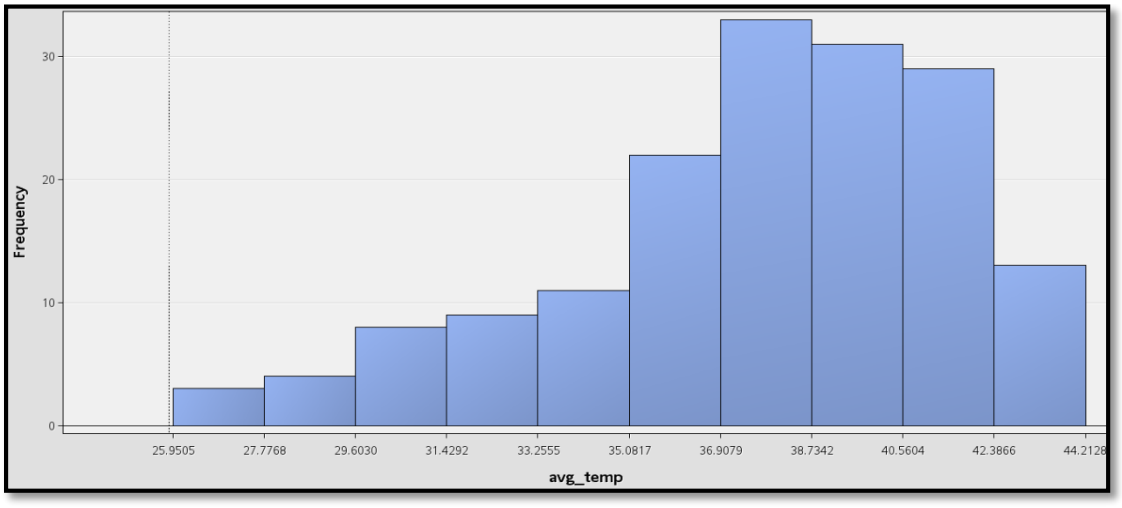
- First part-Years 1901-1925
- Second Part-Years 1926-1940

Now once we perform data exploration and clustering on both of the sets of years, we might get a well understanding of average temperature variation across the weather stations.

1901 to 1925 Data

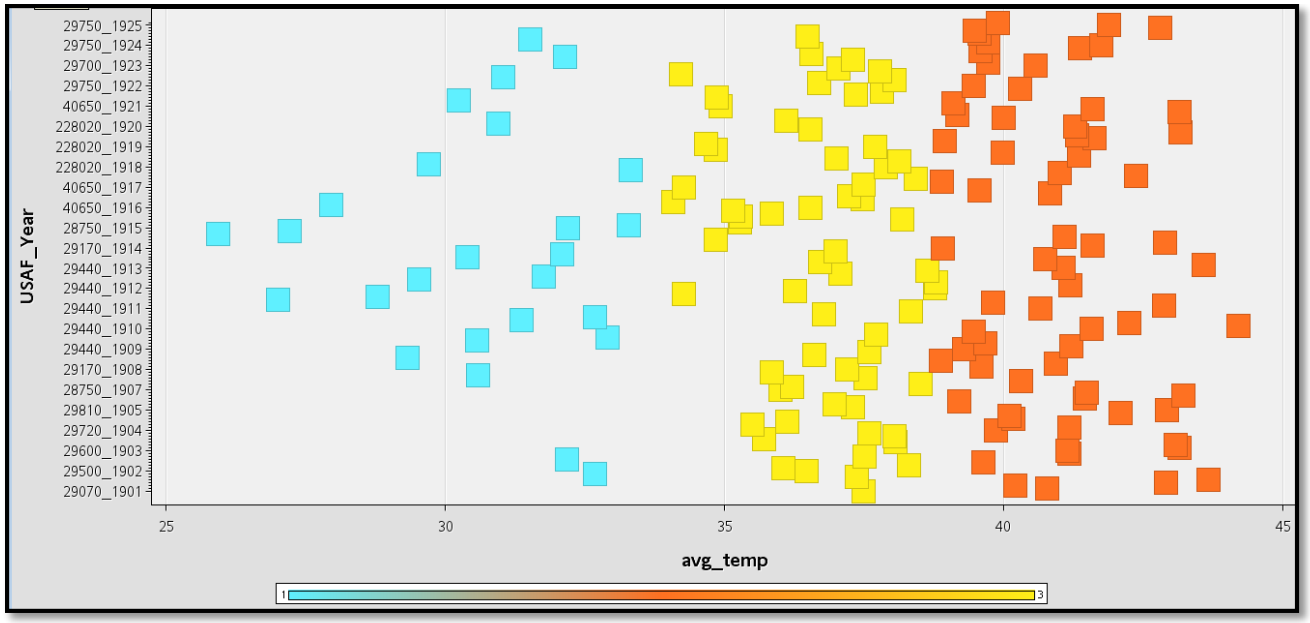
Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
avg_temp	INPUT	37.63231	3.879266	163	0	25.95055	38.06199	44.21284	-0.77878	0.221022

Descriptive statistics showing the spread of average temperature from 25.95F to 44.2128F .data is symmetric, as the mean and median are almost similar. Also data seem to be left skewed if we compare the mean with standard deviation. There are no missing values in the data.



Clustering on Period 1901-1925

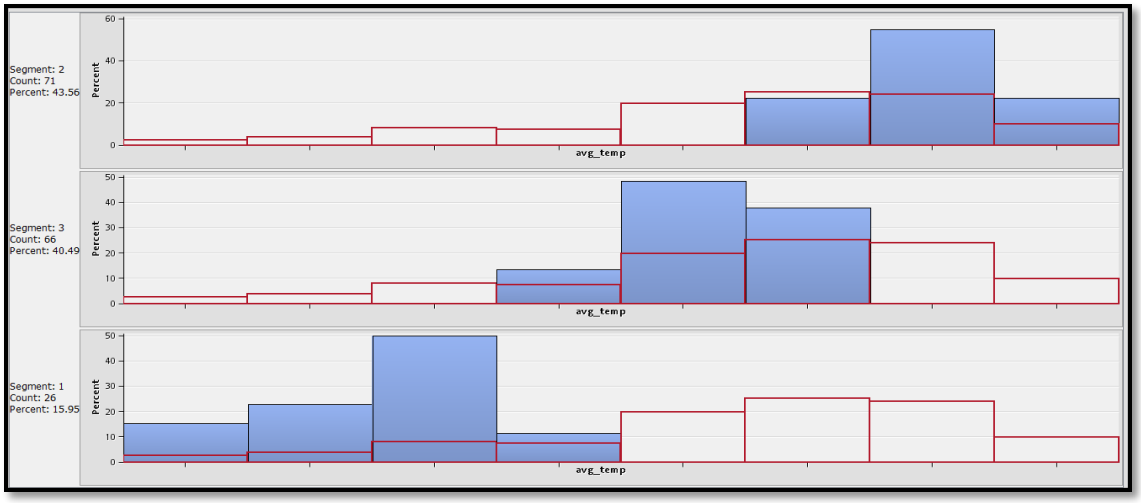
Here we are trying to cluster the weather stations based on average temperature recorded by the stations.



There were 3 clusters were formed:

- **Lowest Average Temperature size cluster with 26 USAF stations**
- **Medium Average Temperature size cluster with 66 USAF stations**
- **Highest Average Temperature cluster with 71 USAF stations**

Profile segment



From the profile segment we can get the detailed explanation about the cluster composition:

Cluster 1: Belongs to lowest average temperature (25.95F-32.79F)

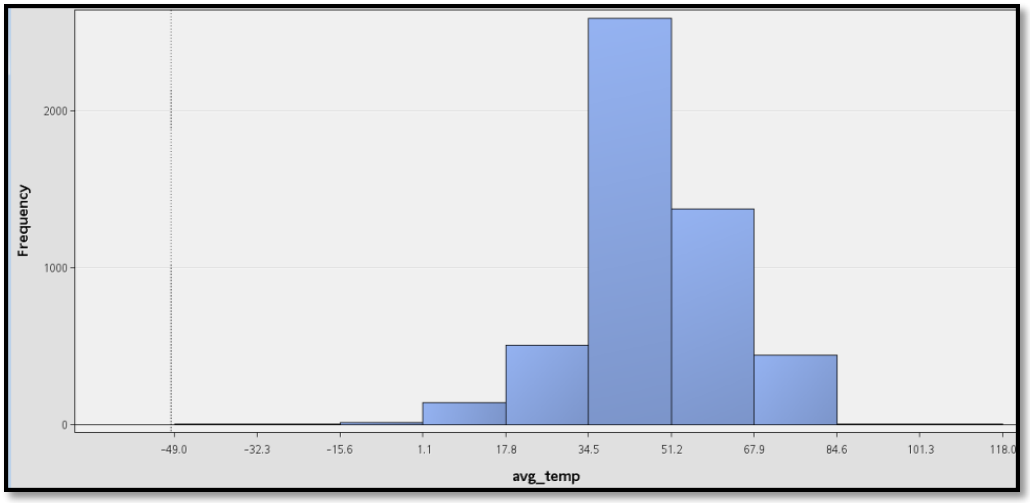
Cluster 2: Composed of Highest Average temperature (37.36F-44.21F)

Cluster 3: Contains medium Average temperature (32.79F-39.64F)

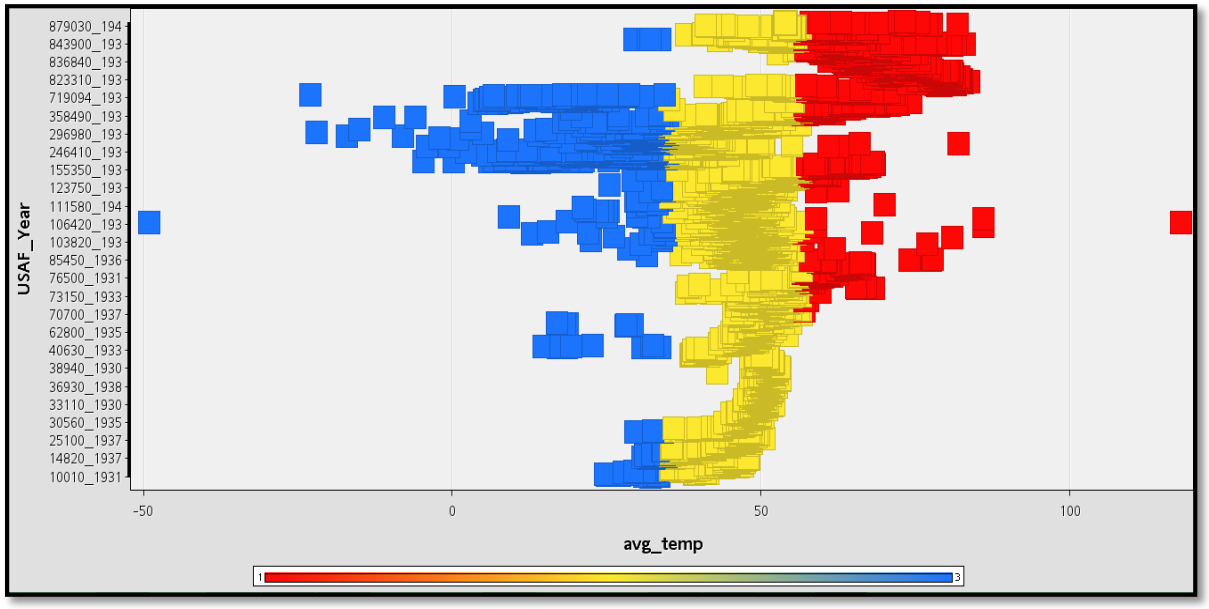
1926 to 1940 Data

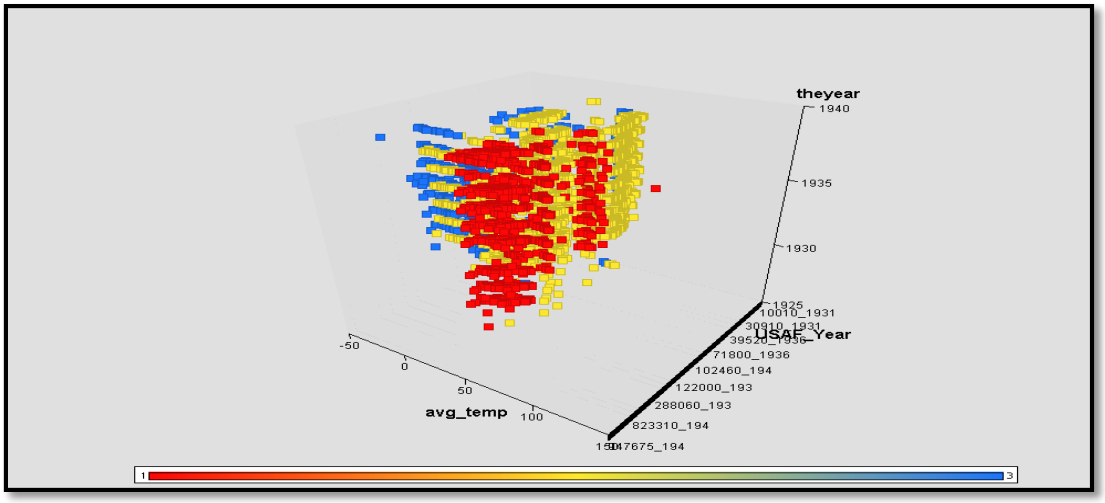
Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
avg_temp	INPUT	48.9087	14.1885	5074	0	-49	48.94194	118	-0.3721	1.700047

Descriptive statistics showing the spread of average temperature from -49F to 118F .data is symmetric, as the mean and median are almost equal. Also data seem to be symmetric and normally distributed if we compare the mean with standard deviation. There are no missing values in the data.



Clustering on period 1926-1940

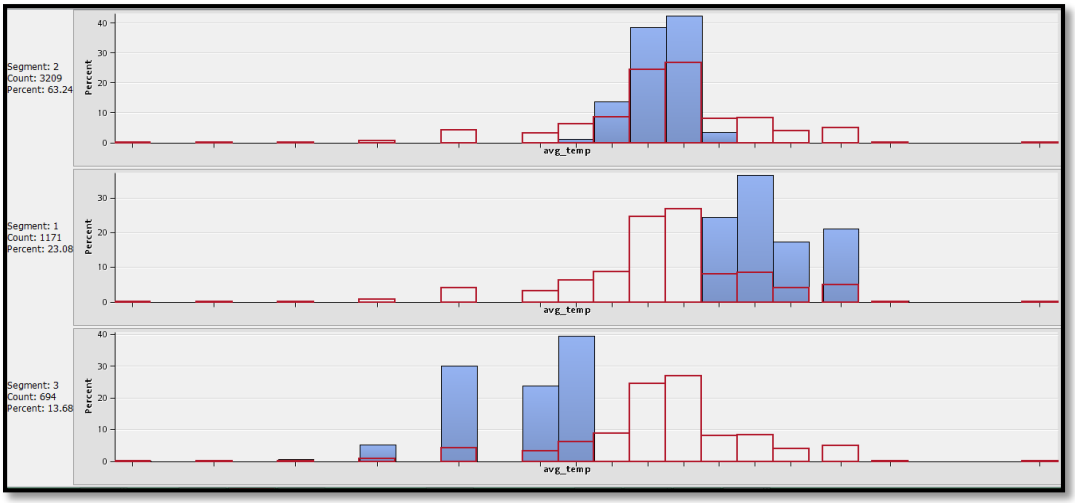




There were 3 clusters were formed:

- **Lowest Average Temperature size cluster with 694 USAF stations**
- **Medium Average Temperature size cluster with 3209 USAF stations**
- **Highest Average Temperature cluster with 1171 USAF stations**

Segment profile



From the profile segment we can get the detailed explanation about the cluster composition:

Cluster 1: Belongs to Highest average temperature (76.255F-118F)

Cluster 2: Composed of Medium Average temperature (34.5F-55.37F)

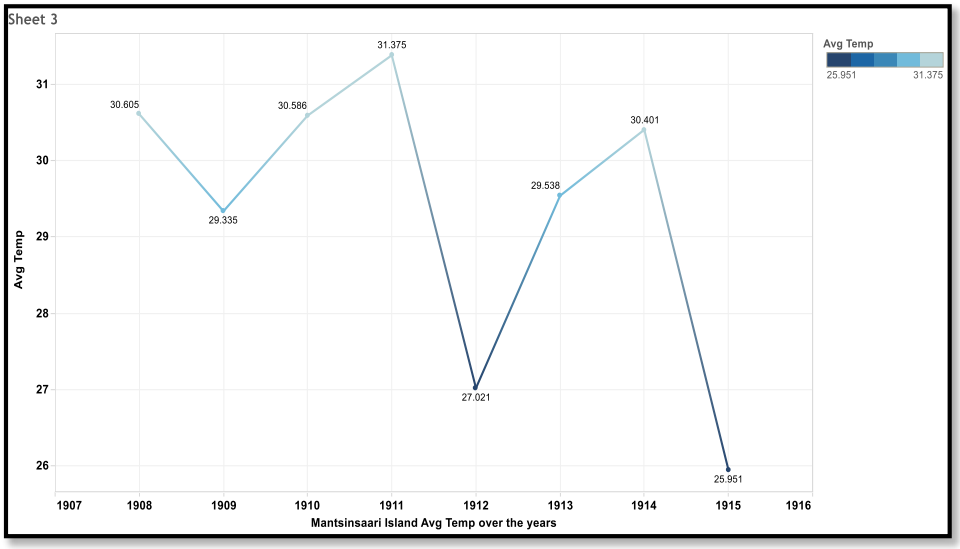
Cluster 3: Contains Lowest Average temperature (-7.25F-49.28F)

Analysis On Extreme Temperature Weather Station In Both Years

For years 1901-1925

Weather station 28060(Mantsinsaari Island,Russia) has been observed as the lowest Average temperature (25.95F) station.

So here we have analyzed the variation in average temperature during its years of operability.

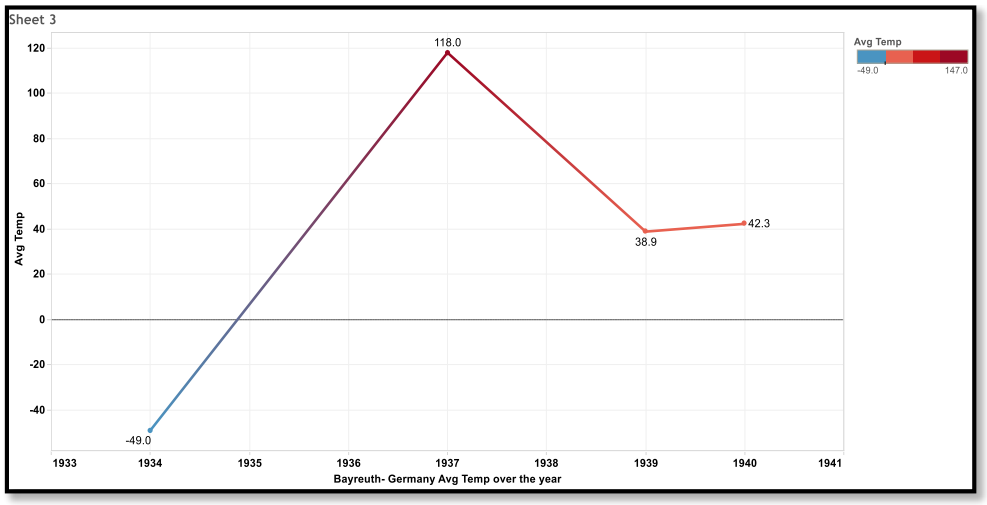


From the graph we can interpret that the USAF station(Mantsinsaari Island,Russia) had the Highest temperature 31.37F in 1911 and the lowest temperature 25.95F in 1915.There seem to be a lot of variation in the trend of average temperature for the years it was operable. Since it’s an Iceland, Variation in temperature is self-explanatory.

For Years 1926-1940

Weather station 106770 (Bayreuth, Germany) has been observed as the extreme low and Extreme High Average temperature station.

So here we have analyzed the variation in average temperature during its years of operability.



Graph clearly shows the extremity in the average temperature for USAF station (Bayreuth, Germany) with the lowest temperature -49.0 in 1934 and the highest temperature 118F in 1937.There seem to be a significant increase and decrease in average temperature for the years it was operable.

Result/Output of Analysis

To start, we did some data exploration on both wind and temperature. After looking at graphical representations, we decided to pursue average temperatures on a global scale. After breaking down the averages by decade, we noticed a pattern of increasing temperatures at different rates. We decided a time series prediction would be a good next step thinking that we would predict increasing temperatures. Our prediction was wrong as the time series forecast predicted that the temperatures will remain relatively flat. We did notice a jump in the temperatures in the mid 1920's, so we wanted to do clustering to see if we can find patterns between these two date ranges.

Analysis of the average temperature over all 40 year (1901 to 1940) might not give the accurate insight of the actual temperature trend over the years since the weather stations were not operable for all the years. Considering dividing the data into parts might give a better insight, we went ahead with this strategy. First part contained data of Years 1901-1925 and second part

contained Years 1926-1940.Clusteing provided the detailed insight of which weather stations belong to Low, Medium or High average temperature. Profile segmentation provided explanation about the boundaries of cluster composition. Analysis of trend corresponding to lowest/highest temperature weather station gave us a clear picture of how the extreme temperature was varying over the years.

SUMMARY

We downloaded, cleaned, filtered, and analyzed the weather station data. The data suggested a possible upward trend in average weather temperature. The assumptions were met for the test we used, but it is predicted that the temperature should remain relatively flat for the next 5 years. Since we saw an increase in temperature, we did clustering to see the differences between the two date ranges that we specified above. We noticed, using clustering, that the later dataset was both more variable and contains a higher median. Further additional analysis could be done using statistical methods.

We have listed the anticipated and unanticipated issues below.

Anticipated Issues

We expected that the filtering logic using Java would be tedious and time consuming. The time took a little longer than expected, but nothing too out of the ordinary. Once all of the data was cleaned and imported to Postgres, querying was simple. I think the time and effort needed was well anticipated, so our group started early and finished on time.

Unanticipated Issues

One of the unanticipated problems that we ran into dealt with the data. There were multiple files that we had at our disposal, but linking them together seemed to be an issue. Two fields, USAF and WBAN, were used together to identify the stations in the dataset. The issue, however, deals with how these keys are used. In stations in the US, the USAF fields are all the same with different WBANs. Outside the US, all of the USAF fields are different. Linking these fields together turned out to be a challenge as we tried to combine tables together.

Another unanticipated problem was the number of null values and unavailable data. There was no consistency in the data available between years, location, or particular fields. It's difficult to decide on any type of analysis since there were so many gaps within the data. This was somewhat anticipated, but we didn't realize how much of an issue this could be if we were not very careful.

Effort by Each Team Member

We all took different sections of this project. We met once a week on Sunday and did many of the work on our own time throughout the week. We feel as a group that everyone contributed and everyone was willing to do their part. We are giving everyone max contribution to this project. See the breakdown below.

Brian Keith Sigurdson (100%): Task 1 – Import and transformation/filtering

Sean Brigadier (100%): Task 4 – EDA/Time Series

Pallavi Kalambe (100%): Task 4 – Data Mining/Clustering/Segment Profiling

Group work together: Project Plan, meetings/brainstorming, paper contribution, querying (various parts of Task 2&3)

BGSU

Appendix A: Task 2 - Top 100

Count	USAF	Years Active
1	29110	17
2	29750	17
3	103380	15
4	108650	15
5	29170	14
6	106370	13
7	228020	13
8	28360	12
9	104190	12
10	101200	11
11	101270	11
12	103840	11
13	104330	11
14	267020	11
15	28750	10
16	29350	10
17	29440	10
18	29820	10
19	40650	10
20	100190	10
21	100670	10
22	100910	10
23	101310	10
24	101470	10
25	101700	10
26	103120	10
27	103610	10
28	104000	10
29	104100	10
30	104160	10
31	104270	10
32	104380	10
33	104530	10
34	104680	10
35	104690	10
36	104880	10
37	105010	10
38	105130	10
39	105540	10

BGSU

40	105690	10
41	105770	10
42	105780	10
43	106770	10
44	106850	10
45	107270	10
46	107280	10
47	107630	10
48	107760	10
49	108030	10
50	108660	10
51	109350	10
52	110350	10
53	111200	10
54	112310	10
55	115180	10
56	116430	10
57	121140	10
58	121160	10
59	122050	10
60	124000	10
61	124250	10
62	228920	10
63	14030	9
64	23610	9
65	28970	9
66	29700	9
67	30050	9
68	30260	9
69	30750	9
70	30910	9
71	31000	9
72	31400	9
73	31620	9
74	31710	9
75	32620	9
76	33010	9
77	33110	9
78	33790	9
79	33960	9
80	34970	9
81	35310	9

82	36010	9
83	36270	9
84	36930	9
85	37660	9
86	37750	9
87	37770	9
88	37950	9
89	38040	9
90	38280	9
91	38560	9
92	38640	9
93	38940	9
94	39520	9
95	39530	9
96	39660	9
97	39730	9
98	39800	9
99	60110	9
100	60410	9

Appendix B: Task 2 - Top 100 Stations With Years Operable

Year	14030	23610	28360	28750	28970	29110	29170	29350	29440	29700
1906				1			1		1	
1907				1			1		1	
1908				1			1		1	
1909				1			1		1	
1910				1			1		1	
1911				1			1		1	
1912				1			1		1	
1913				1			1		1	
1914				1			1		1	
1915				1			1		1	
1916					1					
1917			1		1	1		1		1
1918						1		1		1
1919			1			1		1		1
1920			1		1	1		1		
1921			1		1	1		1		1
1922			1		1	1		1		1
1923			1		1	1		1		1
1924			1		1			1		1
1925			1		1	1		1		1
1926			1		1	1		1		1
1927										
1928										
1929										
1930	1	1								
1931	1	1				1				
1932	1	1				1				
1933	1	1				1				
1934	1	1				1				
1935	1	1				1	1			
1936	1	1	1			1	1			
1937	1	1	1			1	1			
1938	1	1	1			1	1			
1939										
1940										
Grand Total	9	9	12	10	9	17	14	10	10	9

BGSU

Year	29750	29820	30050	30260	30750	30910	31000	31400	31620	31710
1906		1								
1907		1								
1908		1								
1909		1								
1910		1								
1911		1								
1912		1								
1913		1								
1914		1								
1915		1								
1916										
1917	1									
1918	1									
1919	1									
1920	1									
1921	1									
1922	1									
1923	1									
1924	1									
1925	1									
1926	1									
1927										
1928										
1929										
1930			1	1	1	1	1	1	1	1
1931			1	1	1	1	1	1	1	1
1932	1		1	1	1	1	1	1	1	1
1933	1		1	1	1	1	1	1	1	1
1934	1		1	1	1	1	1	1	1	1
1935	1		1	1	1	1	1	1	1	1
1936	1		1	1	1	1	1	1	1	1
1937	1		1	1	1	1	1	1	1	1
1938	1		1	1	1	1	1	1	1	1
1939										
1940										
Grand Total	17	10	9	9	9	9	9	9	9	9

Year	32620	33010	33110	33790	33960	34970	35310	36010	36270	36930
1906										
1907										
1908										
1909										
1910										
1911										
1912										
1913										
1914										
1915										
1916										
1917										
1918										
1919										
1920										
1921										
1922										
1923										
1924										
1925										
1926										
1927										
1928										
1929										
1930	1	1	1	1	1	1	1	1	1	1
1931	1	1	1	1	1	1	1	1	1	1
1932	1	1	1	1	1	1	1	1	1	1
1933	1	1	1	1	1	1	1	1	1	1
1934	1	1	1	1	1	1	1	1	1	1
1935	1	1	1	1	1	1	1	1	1	1
1936	1	1	1	1	1	1	1	1	1	1
1937	1	1	1	1	1	1	1	1	1	1
1938	1	1	1	1	1	1	1	1	1	1
1939										
1940										
Grand Total	9	9	9	9	9	9	9	9	9	9

Year	37660	37750	37770	37950	38040	38280	38560	38640	38940	39520
1906										
1907										
1908										
1909										
1910										
1911										
1912										
1913										
1914										
1915										
1916										
1917										
1918										
1919										
1920										
1921										
1922										
1923										
1924										
1925										
1926										
1927										
1928										
1929										
1930	1	1	1	1	1	1	1	1	1	1
1931	1	1	1	1	1	1	1	1	1	1
1932	1	1	1	1	1	1	1	1	1	1
1933	1	1	1	1	1	1	1	1	1	1
1934	1	1	1	1	1	1	1	1	1	1
1935	1	1	1	1	1	1	1	1	1	1
1936	1	1	1	1	1	1	1	1	1	1
1937	1	1	1	1	1	1	1	1	1	1
1938	1	1	1	1	1	1	1	1	1	1
1939										
1940										
Grand Total	9	9	9	9	9	9	9	9	9	9

Year	39530	39660	39730	39800	40650	60110	60410	100190	100670	100910
1906										
1907										
1908										
1909										
1910										
1911										
1912										
1913					1					
1914					1					
1915					1					
1916					1					
1917					1					
1918					1					
1919					1					
1920					1					
1921					1					
1922					1					
1923										
1924										
1925										
1926										
1927										
1928										
1929										
1930	1	1	1	1			1			
1931	1	1	1	1		1	1	1	1	1
1932	1	1	1	1		1	1	1	1	1
1933	1	1	1	1		1	1	1	1	1
1934	1	1	1	1		1	1	1	1	1
1935	1	1	1	1		1	1	1	1	1
1936	1	1	1	1		1	1	1	1	1
1937	1	1	1	1		1	1	1	1	1
1938	1	1	1	1		1	1	1	1	1
1939								1	1	1
1940						1		1	1	1
Grand Total	9	9	9	9	10	9	9	10	10	10

Year	101200	101270	101310	101470	101700	103120	103380	103610	103840	104000
1906										
1907										
1908										
1909										
1910										
1911										
1912										
1913										
1914										
1915										
1916										
1917										
1918										
1919										
1920										
1921										
1922										
1923										
1924										
1925										
1926							1			
1927							1			
1928							1			
1929							1			
1930	1	1					1		1	
1931	1	1	1	1	1	1	1	1	1	1
1932	1	1	1	1	1	1	1	1	1	1
1933	1	1	1	1	1	1	1	1	1	1
1934	1	1	1	1	1	1	1	1	1	1
1935	1	1	1	1	1	1	1	1	1	1
1936	1	1	1	1	1	1	1	1	1	1
1937	1	1	1	1	1	1	1	1	1	1
1938	1	1	1	1	1	1	1	1	1	1
1939	1	1	1	1	1	1	1	1	1	1
1940	1	1	1	1	1	1	1	1	1	1
Grand Total	11	11	10	10	10	10	15	10	11	10

Year	104100	104160	104190	104270	104330	104380	104530	104680	104690	104880
1906										
1907										
1908										
1909										
1910										
1911										
1912										
1913										
1914										
1915										
1916										
1917										
1918										
1919										
1920										
1921										
1922										
1923										
1924										
1925										
1926										
1927										
1928			1							
1929										
1930			1		1					
1931	1	1	1	1	1	1	1	1	1	1
1932	1	1	1	1	1	1	1	1	1	1
1933	1	1	1	1	1	1	1	1	1	1
1934	1	1	1	1	1	1	1	1	1	1
1935	1	1	1	1	1	1	1	1	1	1
1936	1	1	1	1	1	1	1	1	1	1
1937	1	1	1	1	1	1	1	1	1	1
1938	1	1	1	1	1	1	1	1	1	1
1939	1	1	1	1	1	1	1	1	1	1
1940	1	1	1	1	1	1	1	1	1	1
Grand Total	10	10	12	10	11	10	10	10	10	10

Year	105010	105130	105540	105690	105770	105780	106370	106770	106850	107270
1906										
1907										
1908										
1909										
1910										
1911										
1912										
1913										
1914										
1915										
1916										
1917										
1918										
1919										
1920										
1921										
1922										
1923										
1924										
1925										
1926										
1927							1			
1928							1			
1929							1			
1930										
1931	1	1	1	1	1	1	1	1	1	1
1932	1	1	1	1	1	1	1	1	1	1
1933	1	1	1	1	1	1	1	1	1	1
1934	1	1	1	1	1	1	1	1	1	1
1935	1	1	1	1	1	1	1	1	1	1
1936	1	1	1	1	1	1	1	1	1	1
1937	1	1	1	1	1	1	1	1	1	1
1938	1	1	1	1	1	1	1	1	1	1
1939	1	1	1	1	1	1	1	1	1	1
1940	1	1	1	1	1	1	1	1	1	1
Grand Total	10	10	10	10	10	10	13	10	10	10

Year	107280	107630	107760	108030	108650	108660	109350	110350	111200	112310
1906										
1907										
1908										
1909										
1910										
1911										
1912										
1913										
1914										
1915										
1916										
1917										
1918										
1919										
1920										
1921										
1922										
1923										
1924										
1925										
1926					1					
1927					1					
1928					1					
1929					1					
1930					1	1				
1931	1	1	1	1	1	1	1	1	1	1
1932	1	1	1	1	1	1	1	1	1	1
1933	1	1	1	1	1	1	1	1	1	1
1934	1	1	1	1	1	1	1	1	1	1
1935	1	1	1	1	1	1	1	1	1	1
1936	1	1	1	1	1	1	1	1	1	1
1937	1	1	1	1	1	1	1	1	1	1
1938	1	1	1	1	1	1	1	1	1	1
1939	1	1	1	1	1		1	1	1	1
1940	1	1	1	1	1	1	1	1	1	1
Grand Total	10	10	10	10	15	10	10	10	10	10

Year	115180	116430	121140	121160	122050	124000	124250	228020	228920	267020
1906									1	
1907									1	
1908									1	
1909									1	
1910									1	
1911									1	
1912									1	
1913									1	
1914									1	
1915									1	
1916										
1917								1		
1918								1		
1919								1		
1920								1		
1921								1		
1922								1		
1923								1		
1924								1		
1925								1		
1926								1		
1927										
1928										
1929										
1930	1									1
1931	1	1	1	1	1	1	1			1
1932	1	1	1	1	1	1	1			1
1933	1	1	1	1	1	1	1			1
1934	1	1	1	1	1	1	1			1
1935	1	1	1	1	1	1	1			1
1936	1	1	1	1	1	1	1	1		1
1937	1	1	1	1	1	1	1	1		1
1938	1	1	1	1	1	1	1	1		1
1939		1	1	1	1	1	1			1
1940	1	1	1	1	1	1	1			1
Grand Total	10	10	10	10	10	10	10	13	10	11

BGSU®

Appendix C: Task 2 – Top 50 stations ranked by consecutive years operable

USAF	Max Duration
29750	15
29110	14
103380	14
108650	14
29170	12
106370	11
228020	11
101200	10
101270	10
103840	10
104190	10
104330	10
267020	10
28360	9
28750	9
29350	9
29440	9
29820	9
40650	9
100910	9
101310	9
101470	9
103120	9
103610	9
104000	9
104530	9
104680	9
104690	9
104880	9
105010	9
105540	9
105690	9
105780	9
106770	9
106850	9
107270	9
107280	9

BGSU

107630	9
107760	9
108030	9
109350	9
111200	9
112310	9
116430	9
121140	9
121160	9
122050	9
124000	9
124250	9
228920	9

Appendix D: Task 3 – Descriptive Statistics By Year

Year	Min	Max	Mean	Median
1901	-28	89	40.41018	40
1902	-27	76	35.89901	36
1903	-23	84	40.68208	38
1904	-21	78	37.9962	38
1905	-27	83	39.79713	38
1906	-13	85	40.47534	38
1907	-31	83	37.71818	38
1908	-36	84	37.18481	36
1909	-36	82	37.29496	36
1910	-35	85	38.39961	37
1911	-36	87	37.52899	36
1912	-42	90	35.02317	35
1913	-35	86	37.39244	37
1914	-36	92	37.36608	37
1915	-42	85	32.91849	35
1916	-20	82	35.85519	36
1917	-54	89	36.12053	37
1918	-32	90	38.66225	39
1919	-45	100	36.96824	36
1920	-30	85	39.65716	39
1921	-43	83	38.1449	38
1922	-40	82	36.8679	37
1923	-39	85	36.82199	38
1924	-50	85	38.52756	38
1925	-36	89	38.82299	37
1926	-42	81	37.00807	37
1927	-4	86	46.38209	45
1928	1	95	48.51071	48
1929	-22	91	47.4483	48
1930	-24	104	52.00533	51
1931	-49	115	50.78759	50
1932	-58	120	48.1341	48
1933	-56	120	46.87777	49
1934	-68	120	46.90411	48
1935	-61	120	46.69396	48
1936	-66	131	46.59832	48
1937	-60	120	46.88777	48
1938	-60	120	51.5432	52
1939	-60	120	51.32	50
1940	-56	113	50.7807	53

BGSU