University of Victoria
Engineering & Computer Science Co-op
Work Term Report
Term (Spring/Summer/Fall) Year

The QA/QC Analysis Steps for Temperature Data

BC Ministry of Forests, Lands, Natural Resource Operations and Rural Development
Research Section, Resource Management Kootenay Boundary Region
Nelson, British Columbia, Canada

Zimeng Ming
V00844078
Summer 2020 work term
Combined Statistic and Computer Science
zming@uvic.ca
2020-08-24

In partial fulfillment of the academic requirements of this co-op term

Supervisor's Approval: To be completed by Co-op Employer

This report will be handled by UVic Co-op staff and will be read by one assigned report marker who may be a co-op staff member within the Engineering and Computer Science Co-operative Education Program, or a UVic faculty member or teaching assistant. The report will be retained and available to the student or, subject to the student's right to appeal a grade, held for one year after which it will be deleted.

I approve the release of this report to the University of Victoria for evaluation purposes only.

Signature:  Position: Research Hydrologist Date: 14 April 2020

Name (print): Dr, Natasha Neumann E-Mail: natasha.neumann@gov.bc.ca

For (Company Name) BC Ministry of Forests, Lands, Natural Resource Operations and Rural Development, Kootenay Boundary Region

Contents

Table of Figures

Abstract

Air temperature observation are always having the periods of missing data and bad data. Some of the bad data are caused by the technology issues. The report is discussing the procedure of the quality assurance and quality control of the West Arm Demonstration Forest (WADF) climate database. There are four stations are currently running since 1992, which located in the different evaluation. The report is mainly focusing on the method using for cleaning the dataset during the period of 1992 to 2020. The report is only dealing with the temperature related data and only the air temperature sensors data are loading for the QA/QC process. All the procedure is via the R language and running in the RStudio terminal.

The Description of the dataset

The WADF stations are in the west Kootenay region and belongs to Redfish Creek drainage that near Nelson BC. There are four stations are currently running which are Alpine, Burn, Cabin and Seedtree. The Alpine station are in an alpine meadow and the elevation are 2045(m) for the climate station. The Burn station are in a burned clear-cut and the elevation are 1290(m) for the climate station. The cabin station is a located in a cut-block near the snowmobile cabin. The Seedtree station are in a Seedtree treatment block with the elevation 830(m). (Ministry of Forest) All four datasets are measure the data for both daily and hourly.

Dataset format introduction

The datasets are stored in a excel file that written by the loggers and located in the different folder of Daily and Hourly. Each excel will has the unique column names for each attribute are measured. The details of the column names will be held on the separate sheet. As for the current process, only the attributes data that related to the air temperature are needed for the system. The columns that are using in the system: “Tair_Avg_C” (and may include “Tair2,3,4 _Avg_C if existed),” Tair_Max_C”,” Tair_Min_C”, “Date”,” JulianDay”, “Time”.

Dataset problems and the format restriction

There will be serval technical errors will occur for the WADF dataset. Some of errors are because of extremely cold weather and it will produce the gap of the data. Some of errors are because of the technical issues that will produce the bad data such as -6999(^{o}C) for the temperature.

The format for the dataset is not affect to the script. However, each dataset's name should follow the format as "(Station name) _Daily/Hourly_Year".

Script Manually Setting

Switch Button

The script can be both using for processing the QA/QC for only current year and the all previous year. The Switch Button is located on the top of each steps' scripts. If only need doing the process for current year, switch the button to "True" side, if need to do the process for whole year datasets, switch the button to "False" side.

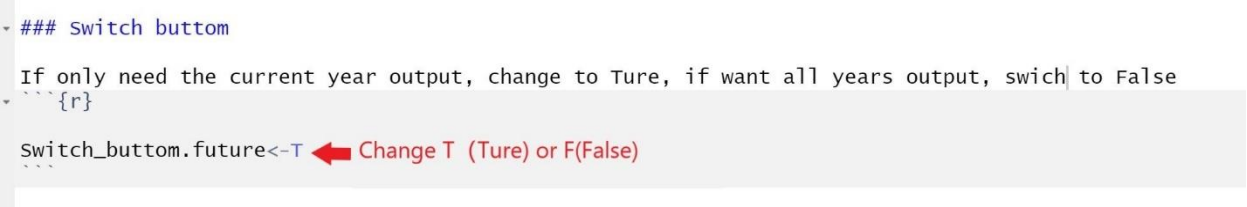

Figure 1

Load the dataset and Output the result

In general, there are serval path need to set manually by the user. The location of path setting is located on the first part of the script for each step. The following Section will show how to change the path and set the path.

Set the input path

The input path need set manually in the following part:

Figure 2

If the output only needs some particular year result, for example, the dataset contains the year range from 1992-2020, but you only need to see the result for 2004, so just change the second path(read_in_path.future) to the file of year 2004 and switch the button to T(true).

As for Step1, it contains an input path for the climate dataset that record but Environment Canada at the station located on Castlegar which will be used for the step1 Rate of Change test. The path needs to be set manually, and the dataset could be download from the official website of the Environment Canada.

The example above is only showing for the step1, so for the step2 and step3, the input path should be the path that after the manually review. The hourly data also will load to the step2 which also need to be manually set.

Set the output path

All the path MUST BE existed already. The R could not make up the result.

- **Output Path for the result CSV file**

The output path will also set manually which will output the graphs and result for it.

````
### For the output path

In this section will set the path that will use for the output the path file.

```{r}
#set the folder path where files will be saved
outPath <- "C:/Users/ZMING/Burn_Output/"

```
````

Figure 3

- **Output Path for Plots**

Each output path for the plots will located on the top of the plot section (not on the front of the script). Normally, the user does not need to change the path code unless the user wants it in the different location(folder) then the script does. The only things need to do is on the top of the script will show which folder need to create and the folder name. Note the folder name must exactly same with the restrict. If the user wants to change the folder name also need to change the path for the plot.

```
# Plots:

##Output path
```{r}
# These folder must already exist before you use it.
plot1.Sensor1_out<-"./Plot1/Sensor1"
plot1.Sensor2_out<-"./Plot1/Sensor2"
plot1.Sensor3_out<-"./Plot1/Sensor3"

plot2.Sensor1_out<-"./plot2/Sensor1"
plot2.Sensor2_out<-"./plot2/Sensor2"
plot2.Sensor3_out<-"./plot2/Sensor3"


plot3.Sensor1_out<-"./plot3/Sensor1"
plot3.Sensor2_out<-"./plot3/Sensor2"
plot3.Sensor3_out<-"./plot3/Sensor3"
```
```

Figure 4

The number of the sensor are different for different station, the script is containing at most 4 sensors. As for the plot path need to create the folder for each sensor for each types of the plot. For the example above, the user should create the folder for 3 plots, and for each plot need 3 subfolders for 3 sensors.

As for the step3, there is no needs to separately set for different sensors but need to be store differently by different types of the graph.

Step1

The purpose of the Step1

This is the first part of the quality assurance and quality control procedure. This part will main focusing on the first look of the dataset and clean the obviously bad data. The step 1 contains the graph about the Temperature of the different sensors versus time and the Minimum and Maximum temperature of the daily average and the test for the obviously bad data. According the research by Kenneth G.Hubbard, Nathaniel B.Guttman, the upper and lower threshold are applied to hold the data in a reasonable range. (Hubbard 2006) Once the obviously bad data coming out, the date of that data will flag to "bad data". The filtered data will export out to a csv file.

There are 5 tests are included in the step1: Detecting obviously bad data, Detecting zeros , Limited test, Rate of change test and Non of change test. For each test and each sensor will create a unique “IsBad” column to indicate the data is bad or not. The main method to find the bad data is to find whether there is an unreasonable data there or not. if we treat a data is unreasonable data, it will create a new column named " IsBad", the default value for that column is False, but it the unreasonable data showing up, it will turn to True.

The Dataset used in the Step1

There are two datasets are using in the Step1. The first one is the Daily climate dataset and the other is the Climate dataset provided by Environment Canada Castlegar station.

The Test used in the Step1

There are total of 5 tests currently run in the step1. Only Obviously Bad Value Test will contain the process of delete data.

Zeros Test

This method will not delete the data directly. Instead of deleting the data directly, the method will create a new column for each year dataset named “IsBad” as the flag of each test for each sensor. If the data are not pass the test, the flag will turn to “TRUE” value. The reason is that the method is using the algorithm to detect the suspect bad value which will also detect some true data as suspect bad value. After running the R script, the output file will go through with the manually review by the researchers in order to make sure the true data are not being deleted.

For the temperature data, it is not impossible for the Minimum and Maximum temperature data are exactly equal to zero. The data are always included to hundredth for each day data, thus if one data is exactly equal to zero it should be a technical issue for the minimum and maximum temperature data. Therefore, it is reasonable to define the value of maximum and minimum of a day data are equal to zero is a bad data. The method will go through each year each day to detect once the maximum and minimum data values are equal to zero will be flagged.

Obviously bad value test

There are serval obviously bad data are included in the dataset due to various technical issue. For example, there are lots of dates for the Burn station data are containing “-6999” degrees for the temperature which are obviously not a good data. For those obviously bad data, the method allows

the researcher to define the bad data value manually and automatically delete the bad data for the whole dataset. This method ensures the deleted data are recognized bad data, only the value looks extremely wired can be defined as obviously bad data. The data value that in the range of reasonable temperature data value should not be defined as an obviously bad data.

On the top of the script, there is variable named “Bad_Data_Values” are allowed user to set the bad data value by themselves.

Limited Test

The limited rules are using 10 years old datasets to make a range of the minimum and maximum temperature for each Julian day, and using 2 times Standard deviation to determine the range of the reasonable data. The test will flag out the outliers when comparing. The main consider of this method is using the limits of the temperature data to detect the data that looks suspect unreasonable temperature at that day. Normally, each Julian day will have similar temperature for each year, using two standard deviation and ten years average can make sure that the data are in a wide range of situation. If the data are out the boundary of two standard deviation, the temperature should consider as suspect unreasonable data.

In order to make the manually review more convenience, limited test is also including the plot of the limited range for each Julian day and use the red line to show the actual data for each year. The figure below is showing an example of the limited and the actual data.

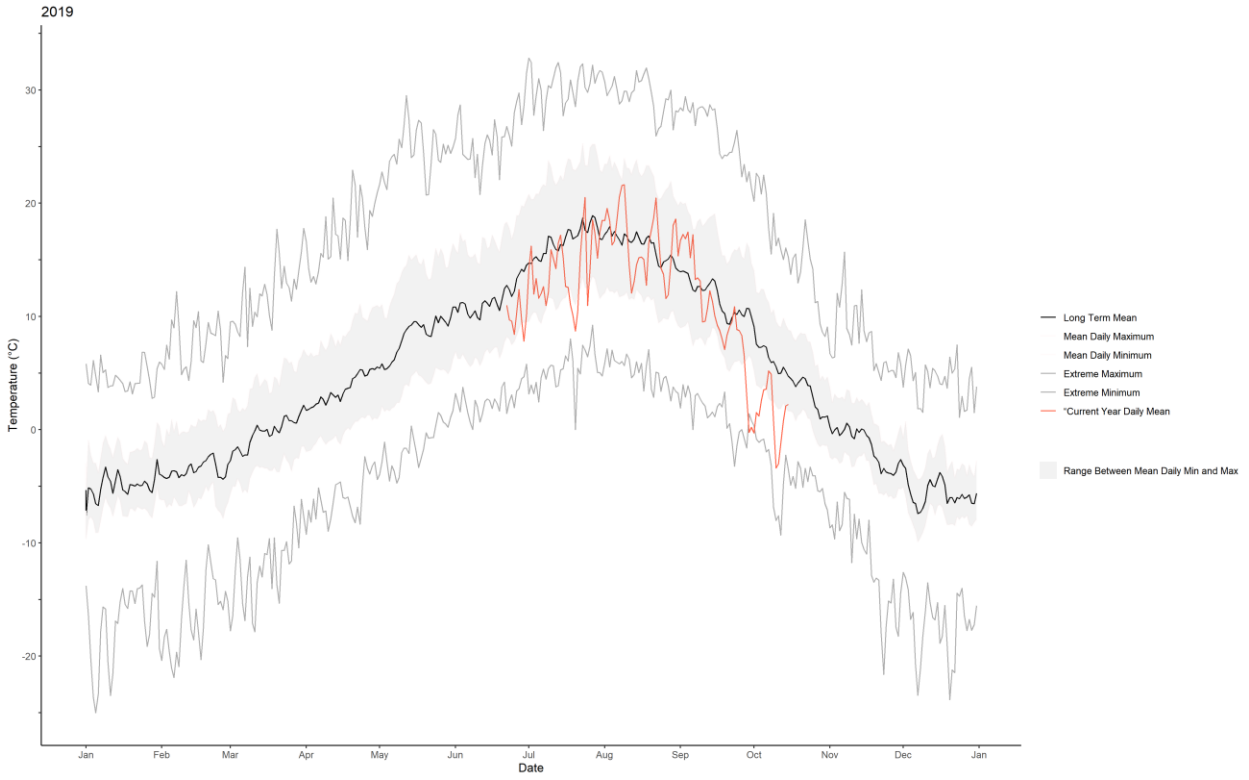

Figure 5

As the plot showing above, there is a suspect data that are showing on the end of the year, which out of the boundary. This date data is very suspect because it is unreasonable cold data for that day. The date will flag to "Ture" for the manually review but before future investigate, it is not possible to make sure whether it is a real extremely cold day in that year or technical issue. The plot for step1 will help the user to define the data that out of the range.

Rate of Change Test

The rate of change rule is using the dataset from Environment Canada dataset to determine the max rate of change for average temperature, maximum temperature and minimum temperature of one day. The Environment Canada has a valid, cleaned and reliable dataset for the station near the

WADF climate stations. The data are published to the public and free to download from the Environment Canada website.

The data are download from the 1993 to 2019. The method first combines all year's dataset to one dataset and use the difference between of temperature data of two adjacent dates to calculate the maximum daily temperature change for the average, minimum and maximum data. Once the result comes out, use the result as the rule to determine suspect date that the difference is bigger than the result. If two adjacent date data's difference in the WADF dataset is bigger than the rule results, the dates can be treated as a Bad data. The flag of this sensor and method will turn to "True".

None of Change Test

When brief looking at the data, it is obviously can see some unusual data which possibly because of the sensor are drift or have other technical issue. According to the pervious analysis, that kind of the problem are very common in the use of the sensors. Thus, the Non-change test are specific to detect the dataset for those problem.

The algorithm for the test is indicate the date n day in a year, if the difference of temperature for the day n-1 and n day, the difference of temperature for the day n and n+1 day and the difference of temperature for the day n-1 and n day minus the difference of temperature for the day n and n+1 day less than 0.1 (The equation are showing below), those three days are flag to suspect data. All the three condition should satisfy at the same time in order to make sure the data are suspect. When the researchers are doing the manually review, once the researchers confirm those

data are the bad data, it is very easy to delete the bad data from the first day to the end of the bad data.

$$abs(T(n-1)) - abs(T(n)) < 0.1;$$
$$abs(T(n)) - abs(T(n+1)) < 0.1;$$
$$abs(abs(T(n-1)) - abs(T(n))) - abs(abs(T(n)) - abs(T(n+1))) < 0.1;$$

Figure 6

The plots that used in Step1

The plot will be using for the limited test. The Variable names can be change in the plot section of the Step1 Script.

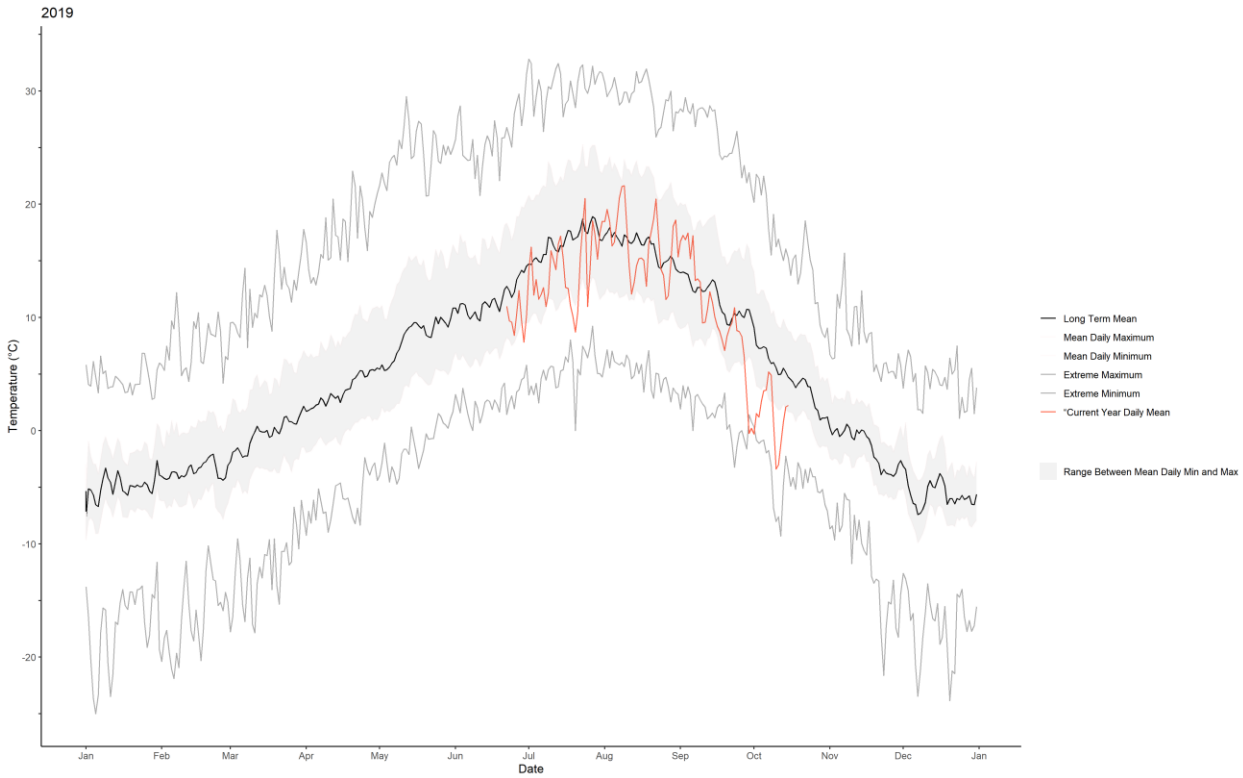

Figure 7

The Output of the Step1

The output of the step1 will be the analysis result dataset which stored in a CSV file for each year.

Manually review for the step1

The analysis result will be showing the flags for each test, the user can manually review for the flags and decided the data needed to be deleted.

Step2

The purpose of the Step2

The step2 is after finishing the manually reviewed of the Step1, the dataset for the daily will load from the files that reviewed. This step will use the hourly data for detecting if there is a day missing too many hours of the data.

The process is first load daily data from Step 1 into data frames (one per year), and then load hourly data into data frames (one per year). After loading the data to R, there is a function to detect how many hours are missing for that day and merge the count number to the daily dataset. It will create a new column for each sensor for the hours of missing. After all the process, the result will output to a CSV file and ready for the manually review.

The reason for doing this step is because the daily data are using the average of all the hourly data per day, if there is a day missing more than 4 hours a day, the daily average should not be correct and might have bias for the daily data. In order to avoid the dataset has the biased daily data, it is necessarily to delete the data with missing hourly data. Once the result is output, the researcher should detect which date data can be deleted after the investigations.

Also, the step2 will using hourly data to calculate the daily mean, which will compare to the daily data, if there is a difference bigger than 0.1 degree, it will be flag to a bad data.

The dataset used in the Step2

There are two datasets used in the Step2, the first one is the dataset after the step1 manually review, the other is the hourly dataset for that station.

The Test used in the Step2

There are two tests are using in the step2. The first one is using for detecting how many hours are missing for each day in order to decided that day data are good or not. The other one is using the hourly data to calculate the daily mean and compare to the daily dataset.

Sensor hour missing test

This test will only use the hourly dataset. The script will split the hourly data for each Julian day, and count how many hours are null for that. If there are more than 4 hours are missing for that Julian Day, it will be flag for the suspect bad data.

Hourly data average vs Daily data average

This test will use the hourly dataset and the Step1 Manually reviewed dataset. The script will split the hourly data for each Julian Day and calculate the average temperature for that Julian Day. Then it will compare to the Step1 Manually reviewed dataset, if the difference between them are bigger than 0.1 degree, it will flag to suspect data.

The plots of the step2

Daily Mean from Hourly Data vs. Logger Daily Mean

The y-axis stands for the average temperature calculated by using the hourly dataset, the x-axis stands for the average temperature which record on the daily dataset.

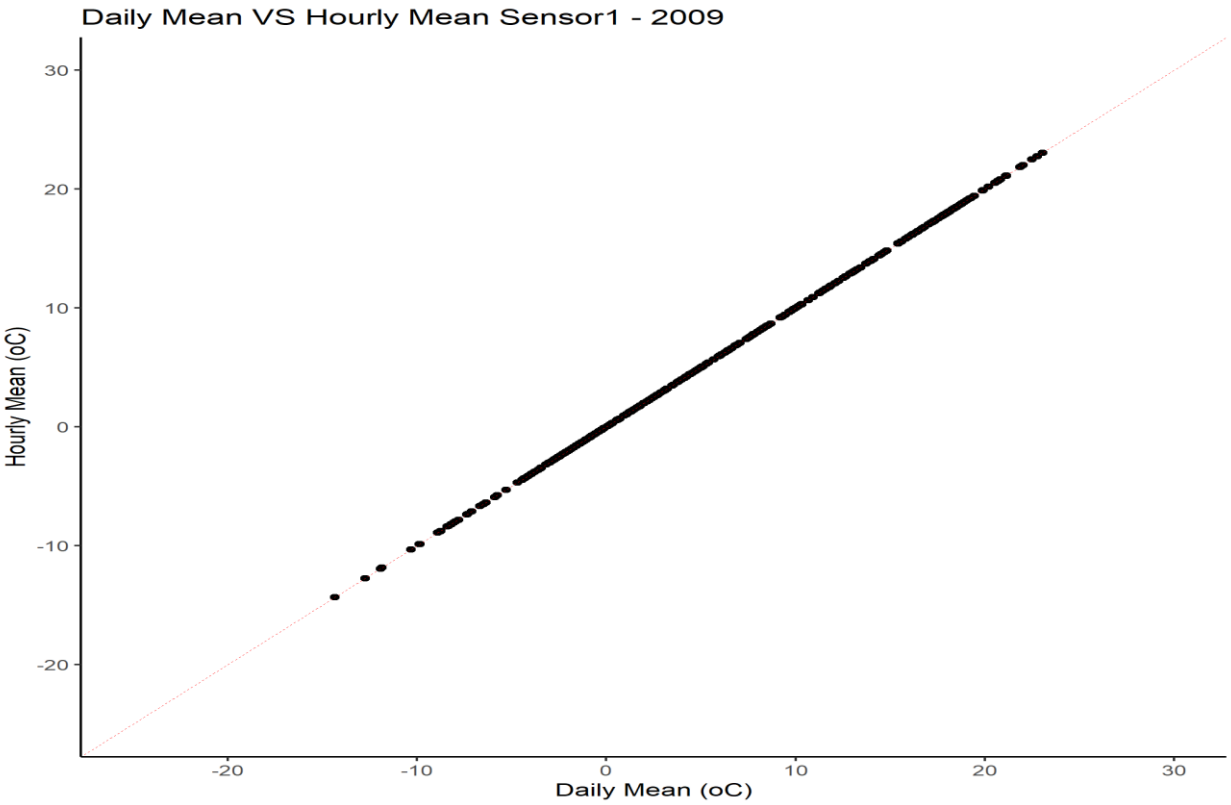

Figure 8

Temperature Difference vs. Day of Year

The y-axis stands for the difference of the average temperature which calculated by hourly and daily separately. The x-axis stands for the Julian Day though the year. The good data plot should be having a solid line lie on the y=0 line which stands for the difference are nearly zero.

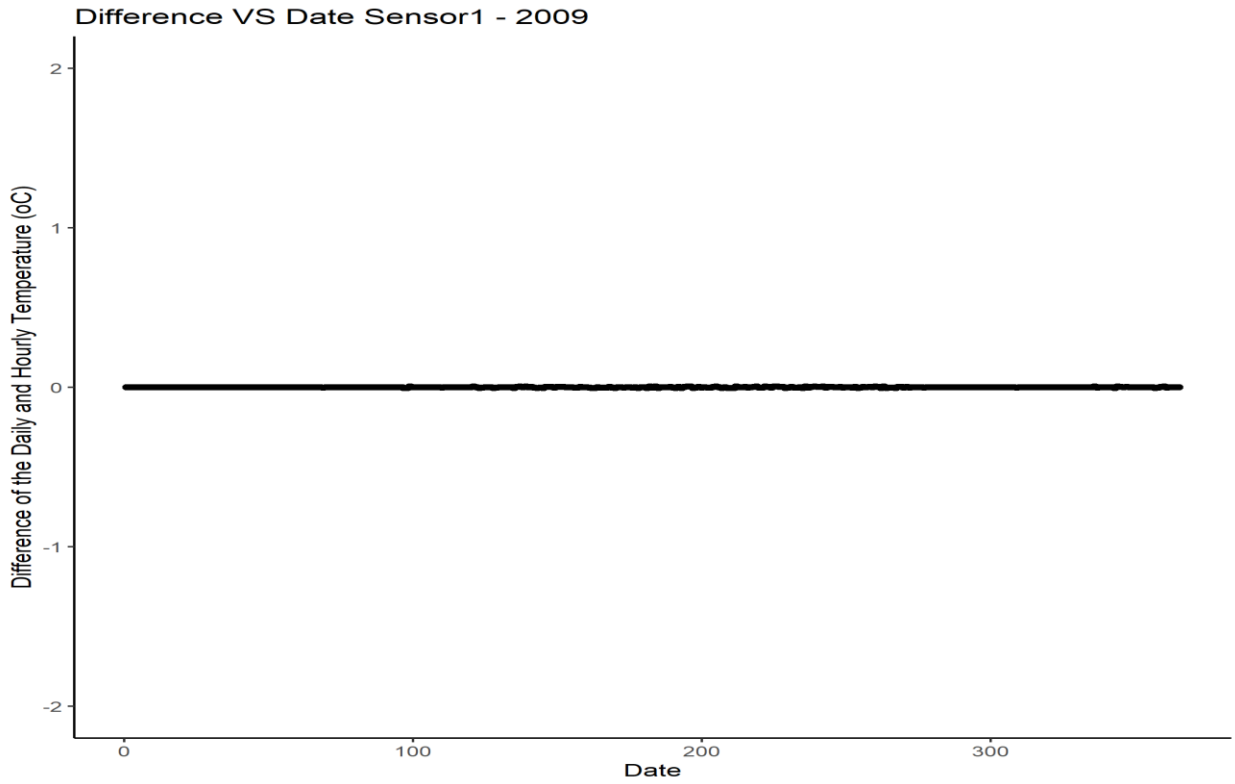

Figure 9

Temperature Difference vs. Logger Daily Mean

The y-axis stands for the difference of the average temperature which calculated by hourly and daily separately. The x-axis stands for the daily average temperature stored on the daily dataset.

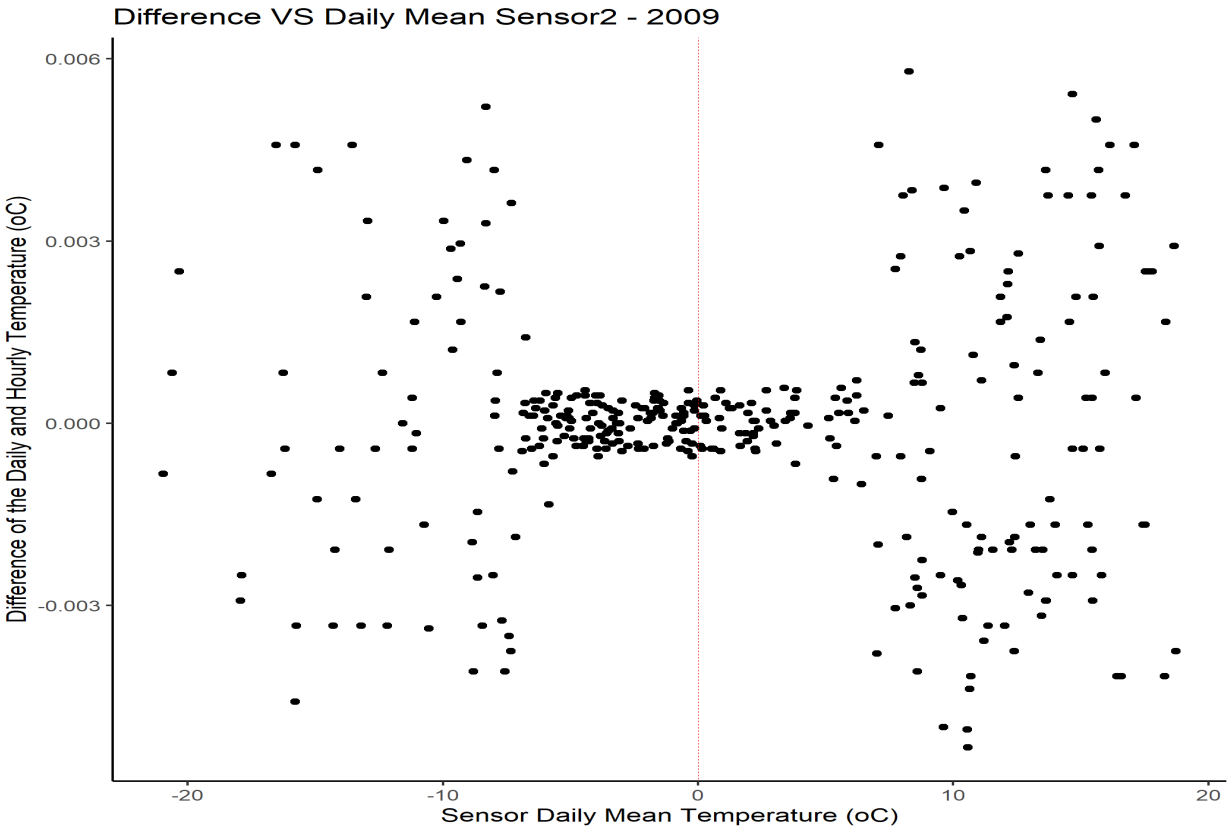

Figure 10

The Output of the step2

There is one csv file contain the analysis of the suspect data and three types of the plots will output from this step.

Manually review for the step2

The CSV file needs to be manually reviewed and the user can decide which suspect data are truly needs to be removed.

Step3

The purpose of the Step3

The step3 are using the data from step1 and step2 to create the plot in order to visualize the data. There are three plots will output from this step: the plot of Sensor1 vs Sensor2,3,4, the plot of the difference between Sensor1 and other sensors vs Date and the plot of the difference between Sensor1 and all other sensors vs Sensor1. For both plots, the plot will only plot the date that there are two or more sensor installed for the station. For example, the Burn station, the plot will used only in the period of 2000 to 2014. The plot should also combine the information of the sensor type and replaced information (located in Step1 part) in order to figure out the data is bad data or not.

There also will output the difference between the main sensor and other sensors. This analysis will help to find the main sensor are drift or not.

The dataset used in the Step3

The step2 manually reviewed data will load to this step.

The plots of the step3

Annual Temperature Difference Graph

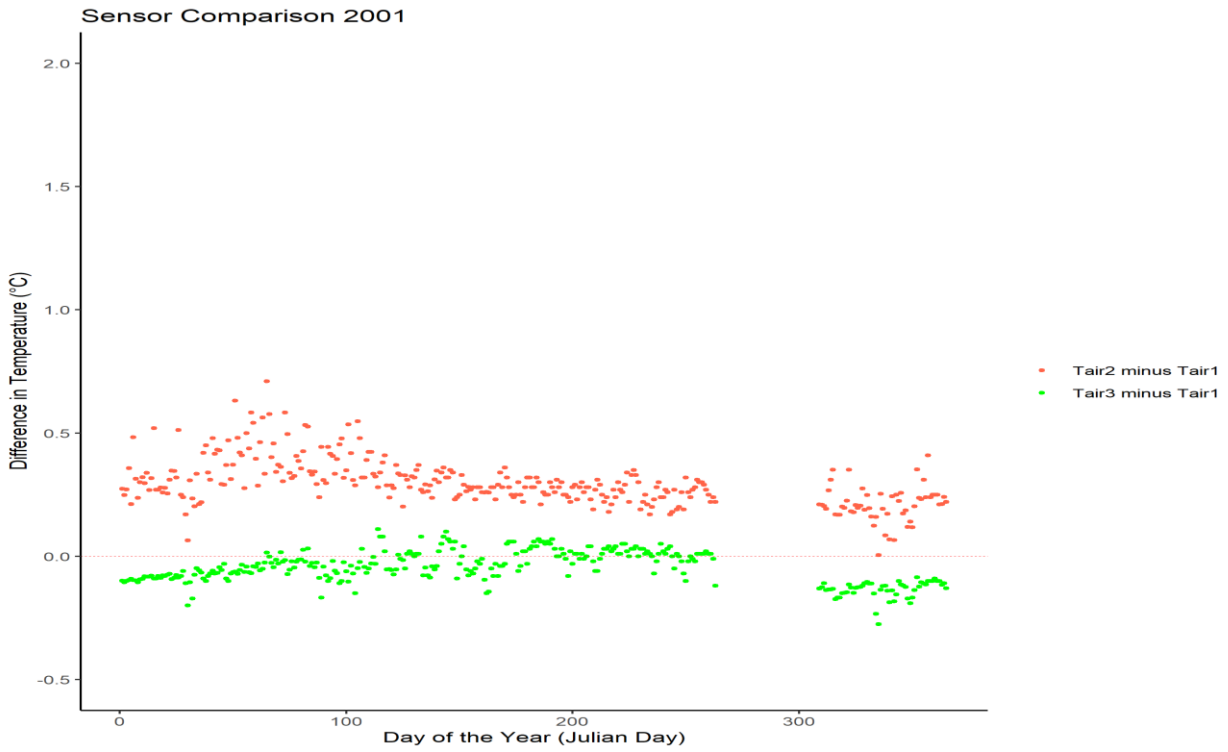

Figure 11

The y-axis stands for the difference between main sensor with other sensors. The x-axis stands for the Julian Day, the color will stand for different secondary sensor difference with main sensor. A good data graph should have the points around zero.

Primary Sensor vs Secondary Sensors

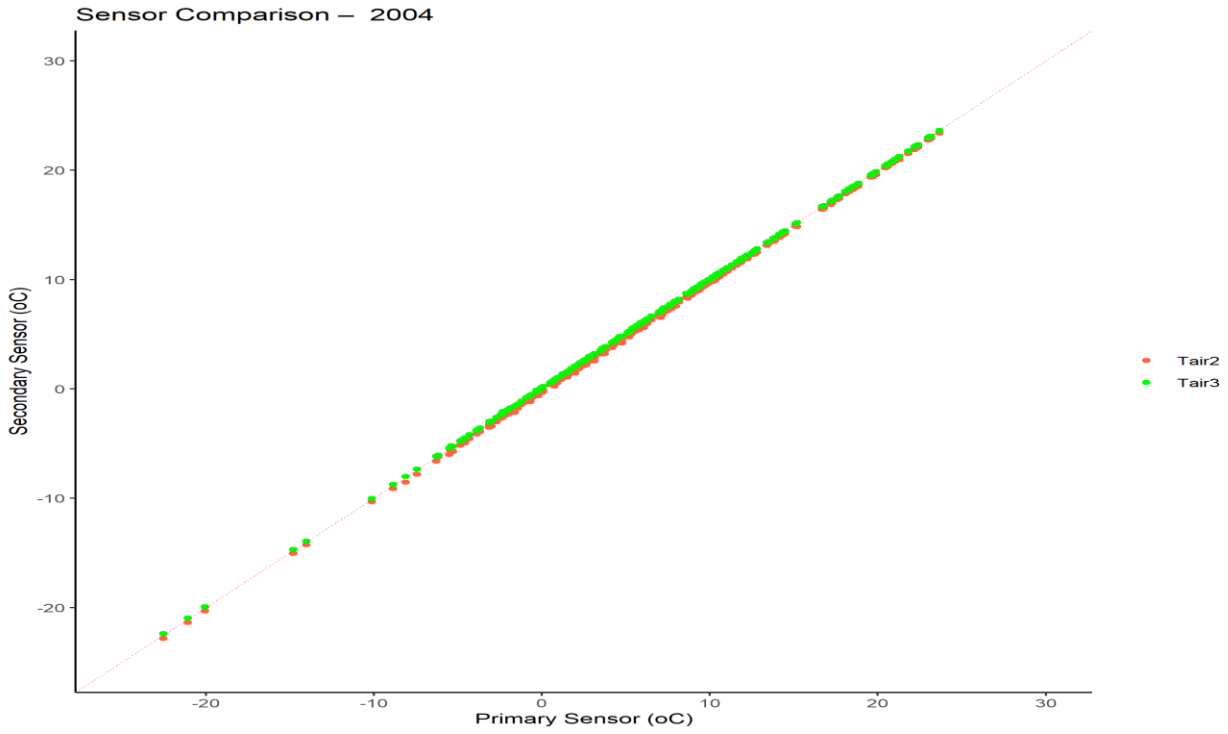

Figure 12

The y-axis stands for the temperature of the secondary sensor. The x-axis stands for the temperature of the main sensor. The different color stand for the different secondary sensor. A good data graph should look like all the points lie on the y=x line.

Temperature Difference vs Primary Sensor

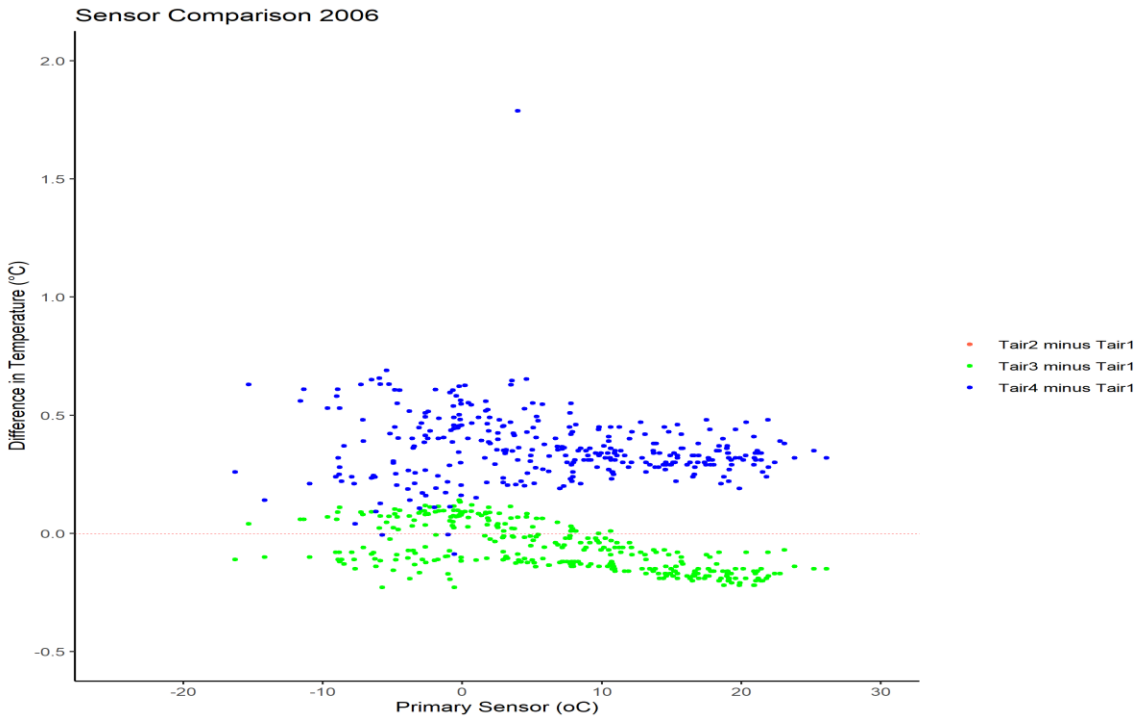

Figure 13

The x-axis stands for the difference between secondary sensors and the main sensor. The y-axis stands for the temperature recoded for primary sensor. The colors are different by different sensors. A graph treated to be a good data graph should have the points around zeros.

The Output of the step3

There are three types plots will output for the step3. Some years are not showing because of the empty of the graph. i.e., if there is only one sensor work for some years, those years graph will not showing up.

Also, the CSV file will contain the difference between other sensors and main sensor. Stored by years.

Common Errors

This section will include the common errors for the script and the dataset.

Errors for the Script

1. Sometimes the script will have the error when it runs the method. As showing the graph below. This is because the path of the dataset is not recognized. The user should check the path setting and make sure the dataset is load to the R.

```
> df.test<-mapply(add_MissingNumber_to_Daily_Tair1, df.test,missingHours.Tair1)
Error in mapply(add_MissingNumber_to_Daily_Tair1, df.test, missingHours.Tair1) :
  zero-length inputs cannot be mixed with those of non-zero length
```

Figure 14

2. If there is any warning, normally, it will not matter to the result.
3. For the graphs, it might need to change legends by the user self, but the order needs to be exactly same with the one before editing.

Errors for the Dataset

1. It may contain some errors for the obviously bad data, such as “-6999”, “-53” or other numbers. The obviously bad data need to be designated manually by the user.
2. The hourly data may have lots of NA’s from the first day of the year, if the missing hours are more than 1000 rows, the data will automatically change to type of Boolean and which will show a “1” and “0”. The solution to this error is that adding a significantly wrong number on the first row in order to avoid the data change to Boolean.

Errors for the Rmarkdown

1. This is only the user needs to output the script to a pdf file. If the Rmarkdown fail to output the pdf file, check the white area (not include the code chunk) whether it has some special symbols in it.

2. If the user need to change the features of the pdf file, can view this website for more information: https://rmarkdown.rstudio.com/lesson-3.html

Build-in package versions

Package Name	Version	Use of the package
Readxl	1.3.1	Read the excel files in
ggplot2	3.2.1	Plot the graph
dplyr	0.8.3	Data operation
scales	1.1.0	Data operation
reshape2	3.6.2	Re-formate the data for plot
plyr	1.8.5	Data operation
naniar	0.5.0	Using for the missing value
formatR	1.3.1	Formating the R Script and Rmarkdown

Conclusion

After these steps, if all the plot looks normal the dataset can be consider cleaned. The step1 are using the test to determine a wide range of suspect data, after the manually reviewing, the suspect data can be investigated carefully and cleaned for the obviously bad data. The step2 are using for deleting the data that collecting with the bias when there exist too many missing hours. Step3 are the insurance step that can ensure there is no other biased or unreasonable data in the dataset. The visualization of the dataset will directly show how the dataset quality is.

Although there are lots of method could detecting the bad data, there still not a common test for all types of the dataset. The test is very specific for the unique situation of the dataset. For example, step1 Non-change test are very specific test to use for the WADF dataset, this is not very common issue for the other dataset. However, the limited test and the rate of change test can be considered for the future climate dataset. The algorithms are not specific for the WADF dataset. Step2 and Step3 are very efficient way to find the bias data or the sensor drift data. Using the visualization of the data is the most common and efficient way to have a big picture of the dataset that working with.

Once the three steps are complete, the dataset can be published to the public without any unreasonable data in the dataset. The steps are very efficient and use for the test of Alpine and Burn station which both stations are satisfied with good quality.

References

1. The West Arm Demonstration Forest. (n.d.). Retrieved from https://www.for.gov.bc.ca/RSI/research/WADF/WADF.htm
2. Hubbard, K. G., Guttman, N. B., You, J., & Chen, Z. (2007). An Improved QC Process for Temperature in the Daily Cooperative Weather Observations. Journal of Atmospheric and Oceanic Technology, 24(2), 206–213. doi: 10.1175/jtech1963.1
3. Feng, S., Hu, Q., & Qian, W. (2004). Quality control of daily meteorological data in China, 1951–2000: a new dataset. International Journal of Climatology, 24(7), 853–870. doi: 10.1002/joc.1047