Project Assignment - Individual Report

3. **Sydney**:
The data of June **1858, 2010** is incomplete so the total amount cannot be computed, treat them as gaps.
Method A: Threshold value is 510.6 mm
A once in 20 years maximum rainfall for the month of June: 1950, 2007.
Method B: Threshold value is 373.2 mm
A once in 20 years maximum rainfall for the month of June: 1864, 1885, 1896, 1937, 1949, 1950, 1964, 2007.
Queanbeyan:
The data of June **1870** is incomplete so the total amount cannot be computed, treat it as a gap.
Method A: Threshold value is 136.9 mm
A once in 20 years maximum rainfall for the month of June: 1891, 1931, 1956.
Method B: Threshold value is 113.5 mm
A once in 20 years maximum rainfall for the month of June: 1891, 1925, 1931, 1956, 1975, 1997, 1998.

4. My program consists of four modules:
main.py: The entry of the program. The main function will interact with user, read user's input, call the aggregation functions according to user's choice and finally print the results.
aggregate.py: Based on pandas.py, including four aggregation functions for daily, monthly, yearly and a specific month respectively. The input is the path of csv file and the output are a time series list, an aggregate rainfall amount list corresponds to the time list and a time series list for those entries where the data is incomplete. Particularly, the output of the aggregation function for single day observation is a dataframe which is a data structure of pandas.py.
utils.py: - Method A and B. For method A, the input is time series list, rainfall amount list, frequency that user input and a parameter for determining whether to compute a threshold for exceptionally high or low values. For method B, the input is the total entries number, frequency that user input, time series list and rainfall amount list. The output is the threshold value and a map in which key for the exceptional time and value for corresponding rainfall amount.
- Some other utility functions, such as getting city name, getting month name, getting days of month or year, are related to time, which are used in printing and data validation.
- The last function is a decorator which is used to print the total running time of a function.
validate.py: This module is used for validating the integrity of the dataframe of daily, monthly, yearly and any specific month. The input is a dataframe read from csv file by pandas, the output is a bool list in which each element corresponds to a year/month/date, and a time series list for those entries where the data is incomplete.
- The last function is an invalid data filter, which could filter strange data. In fact, the only invalid row in three files is the period field of 2008-4-26, index 54902, in Rainfall_Sydney_066062.csv

5. The aggregation functions can be written without depending on any extra module. Take aggregation for specific month as an example, I can use a for loop to read and compute all lines. To begin with, filter invalid data, validate the integrity of the csv file, then skip the first line to get the initial value of year and month. Read the rows in which the month field is equal to the specific month after converting into integer. When it finished reading the whole month, append the sum value to a list and then continue.
In that way, I could exactly know what each statement is used for. Nevertheless, it might be low efficient and unreadable. By importing pandas, this function can be implemented in a few lines of code with high efficiency and readability. When analyzing the data, I notice that pandas provides a wide range of data analysis and manipulation functions as well as easy-to-use data structures. It is a powerful data analysis toolkit which helps me finish the assignment in an easy way.