Report of Consulting Project

Jingyu Li
Pruthvi Bharadwaj
Yuyang Li
Peng Liu
12/14/2021

Introduction:

Our project is consultation on dissertation research related to building simulation modeling and weather file inputs. Our client is Kate Connolly from the Environmental Health Department of the School of Public Health. She is working on a project on a co-simulation housing model, which requires a type of weather data called a Typical Meteorological Year (TMY) file as input to her project. She hopes to use bootstrap analysis to randomly select months in historical periods (such as 1976-2005) to create output TMY files (30 times).

Data we used:

The data we obtained is the national solar radiation database of 1961-1990 and 1991-2005. The client hopes to extract the data from 1976 to 2005 from this data set and generate 30 TMY files.

When we extracted and cleaned the data, we found that the two data sets did not match completely, and there were many missing variables in the data set. After discussion and search, we finally finalized the data set as the National Solar Radiation Database of 1998-2020.

Github:

https://github.com/consulting-TMY3/bootstrap

User's manual:

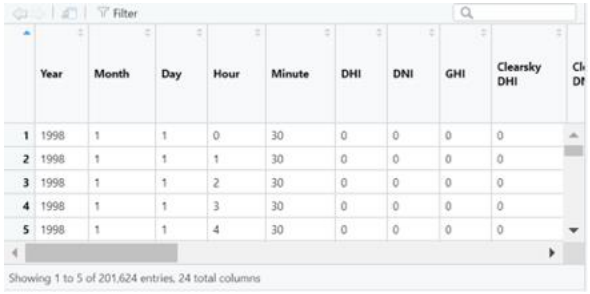
Our code is made of three main files: get_data.R, bootstrap_tmy.R , visualization.R

1. In get_data.R, we set a function named *get_data*(), the default input is all the TMY files in the folder *data*, and the output is a data frame contained all rows in 22 TMY flies.

Run the function:

```
df1 <- get_data()
```

The output data frame:



	Year	Month	Day	Hour	Minute	DHI	DNI	GHI	Clearsky DHI	Cl DI
1	1998	1	1	0	30	0	0	0	0	
2	1998	1	1	1	30	0	0	0	0	
3	1998	1	1	2	30	0	0	0	0	
4	1998	1	1	3	30	0	0	0	0	
5	1998	1	1	4	30	0	0	0	0	

Showing 1 to 5 of 201,624 entries, 24 total columns

2. The second file named bootstrap_tmy.R is the core of our procedure, consist of function *bootstrap_tmy*() and function *repeat_bootstrap*(), the first function is applied in the second function.

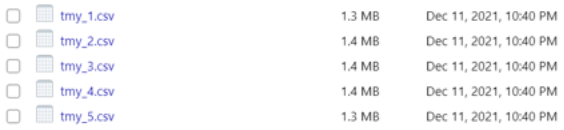
The input of bootstrap_tmy() is a data frame as the source to bootstrap, and in this project the source is the output df1 from former function df1 = get_data(), and it will output a new data frame produced by df1 as the result of bootstrap, that is a one-year new TMY file. This function is the basics for the second fuction repeat_bootsrap(n, df)

And in the function repeat_bootsrap(n, df), n is the repeat times of running bootstrap, if user input n=30, it will get 30 new one-year TMY files, and df is the data frame as the source of bootstrap. The output of this function will be n csv files, which are all one-year TMY files using bootstrap.

When n = 5:

```
repeat_bootstrap(n = 5,df1)
```

It will generate 5 csv files in folder automatically:



tmy_1.csv	1.3 MB	Dec 11, 2021, 10:40 PM
tmy_2.csv	1.4 MB	Dec 11, 2021, 10:40 PM
tmy_3.csv	1.4 MB	Dec 11, 2021, 10:40 PM
tmy_4.csv	1.4 MB	Dec 11, 2021, 10:40 PM
tmy_5.csv	1.3 MB	Dec 11, 2021, 10:40 PM

3. We used visualization.R to validate and visualize the outcome.

In this file, we wrote a function to check the output file in bootstrap_tmy.R. The user only needs to run the code of lines 1 to 8 first, enter the file name that he wants to check in the code of line 37, such as "tmy_1", and then run the line to get the data and the original

TMY file Comparison chart. The red line segment represents the data to be checked, and the blue line segment is the original TMY file.

visualization comparison with the example:

In our example, we used one of the results of bootstrap tmy_1.csv to compare with the first one-year TMY file in the original data. tmy_1 has 8760 observations (hours), which is equal to 24*365. Due to the randomness of bootstrap, this value may have slight differences.

We chose temperature as a factor to plot the comparison of the two data frames. This factor can be any variable that exists in the TMY file.

In this figure, we can see that the overall trends of the two data frames are similar.