Assignment 1 Reading Rectangular/Tabular data

Not so simple!

STA141B Spring 2023

Professor Duncan Temple Lang

Due: April 17, 11pm

Submit via Canvas

The task is to read solar and climate-related data from a variety of locations in California. The data are related to solar performance for buildings and simulation models for understanding this solar performance.

We have data for all of the USA and also for the entire world.

We will focus on just 5 locations and 5 ZIP files:

```
USA_CA_Fairfield-San.Francisco.Bay.Reserve.998011_TMYx.2007-2021.zip
USA_CA_Marin.County.AP-Gnoss.Field.720406_TMYx.2007-2021.zip
USA_CA_Napa.County.AP.724955_TMYx.2007-2021.zip
USA_CA_Point.Reyes.Lighthouse.724959_TMYx.2007-2021.zip
USA_CA_UC-Davis-University.AP.720576_TMYx.2007-2021.zip
```

These are available in the Files section of the course's Canvas portal.

In each ZIP archive, we have files with different file extensions such as clm, ddy, epw, stat, e.g.,

```
USA_CA_Bodega.Bay.CG.Light.Station.724995_TMYx.clm
USA_CA_Bodega.Bay.CG.Light.Station.724995_TMYx.ddy
USA_CA_Bodega.Bay.CG.Light.Station.724995_TMYx.epw
USA_CA_Bodega.Bay.CG.Light.Station.724995_TMYx.pvsyst
USA_CA_Bodega.Bay.CG.Light.Station.724995_TMYx.rain
USA_CA_Bodega.Bay.CG.Light.Station.724995_TMYx.stat
USA_CA_Bodega.Bay.CG.Light.Station.724995_TMYx.wea
```

There is a basic description of the formats of these different files in the ZIP archive. However, you will have to explore the details of sample files to understand the general structure.

Tasks

For the 5 ZIP files provided, you are to read data from the

- .wea file
- .pvsyst file
- multiple rectangular tables in the .stat file.

Each of the .wea and .clm files correspond to a single rectangular data set. However, the .stat file contains multiple rectangular tables between other text.

Write functions (rather than one or more R commands) to read each of these tables, as you need to apply these to the contents of the 5 ZIP files. Also, you will most likely need to run the code multiple times to iteratively modify it and verify it is correct. Please use functions rather than repeating the same code for each file and table.

For the .stat file, read the tables for

- Monthly Statistics for Dry Bulb temperatures
- Monthly Statistics for Dew Point temperatures
- Average Hourly Statistics for Dry Bulb temperatures
- Average Hourly Statistics for Dew Point temperatures
- Average Hourly Relative Humidity
- Monthly Wind Direction {Interval 11.25 deg from displayed deg)
- Average Hourly Statistics for Direct Normal Solar Radiation
- Monthly Statistics for Wind Speed
- Average Hourly Statistics for Wind Speed

For each of the monthly data sets:

- Verify that the Max Hour and Min Hour are correct.
  - then omit these rows
- Convert the data so that the rows corresponding to measured variables and dates e.g. Maximum, Minimum, Daily Avg, . . . are columns and the columns corresponding to months are rows.
- Convert the Day:Hour values to a time (POSIXct). Use 2023 as the year.
- Convert the measurements for other variables to numbers.

For the hourly data tables,

- convert each to a data.frame with 3 columns:
  - converting the month-hour pairs to rows with the single variable as a column
  - one column for the month
  - one column for the hour - 0, 1, 2, 23

Finally,

- combine the average hourly tables into a single data frame with a column for each variable, i.e., dry bulb temperature, dew, relative radiation, wind speed. Ensure that the rows correspond to the same time, i.e., month, hour and day.
- for each variable, plot the values against hour for each month.

Do this for the 5 zip files.

Try to find a common structure for the monthly and then the hourly, or for both, so that you can write code to read these generally rather than code for each specific table.

Verifying Results

It is vital to verify that the results are correct. You need to check by

- manually comparing individual values in the files and the results,
- computing summary statistics from the results, and
- visualizing the results
- programmatically verifying the results,

to ensure they make sense and are correct.

Describe the approaches and processes by which you verified the results.

Identify Assumptions

State any assumptions you are making about the structure and order of the data, and show how you verified these were true.

Useful Functions

- strsplit()
- lapply(), sapply()
- list.files()
- readLines(), read.csv(), read.table()
- substring(), substr(),
- trimws()
- grep(), grepl(), gsub()
- which.min(), min(), max()
- data.frame(), as.data.frame()
- unlist()
- rep()
- strptime(), as.POSIXct(), as.Date()
- sprintf(), paste(), paste0()
- textConnection()
- close(), on.exit()
- %in%
- unzip()
- system(), system2()
- rbind(), do.call()
- by(), tapply(), aggregate()

The essential functions for checking results correspond to what you expect include and debugging code include:

- length(), names(), dim(), nrow(), ncol(), class(), typeof(), is.na()
- debug()
- browser()
- `options(error = recover)`
- summary(), plot()