DATA ANALYTICS GLASGOW

R Programming

Project

This project is assessed and contributes 25% to your overall grade for this course.

Please upload your project code (Shiny app and additional scripts) before Monday, January 25th, 10am (UK time).

To upload your coded solutions, log on to Moodle and go to R Programming. Under Week 11, there is a link Upload code for project (due January 25th). Click on this link to upload a zip file containing your Shiny app and any additional scripts.

It is recommended that you also try to deploy your Shiny app on `shinyapps.io`*. This way you can make sure that the app does not just work on your computer. If you deploy your app on* `shinyapps.io` *and you provide the URL, then the* `shinyapps.io` *version can be used to test functionality. This avoids any issues of reproducibility.*

In this project you will write a Shiny app which visualises meteorological data collected at several locations across the United Kingdom from 1st January 2020 to 30th November 2020.

The data is available on Moodle for you to download as a series of CSV files, which have been obtained from the Met Office Integrated Data Archive System (see `https://catalogue.ceda.ac.uk/uuid/220a65615218d5c9cc9e4785a3234bd0`). Data have been obtained from 20 weather stations. The file `Sites.csv` contains information about each weather station. There is one CSV file for each weather station which contains hourly weather measurements on air temperature, wind speed, relative humidity and visibility. The file names are of the form `Site_<Site_ID>.csv`. Each row of these files contains hourly measurements of the weather measurements at that station. For some stations there are some missing observations at certain time points.

The weather variables of interest that are being reported are:

`wind_speed` Average wind speed in knots.

`air_temperature` Air temperature measured in $^{\circ}C$. Note, temperature has been rounded to the nearest 0.1 $^{\circ}C$

`rltv_hum` Relative humidity, measured as a (%)

`visibility` Visibility, measured in metres.

Your task here is to develop a Shiny app which both visualises and summarises the data. Your app should also calculate and report a diagnostic called the Hutton Criteria used to alert farmers of the risk of potato blight forming on potato crops.

The Hutton Criteria occurs in a particular day when **both** the follwing criteria are met -

- The two previous days have a minimum temperature of **at least** 10 $^{\circ}C$
- The two previous days have at least six hours of relative humidity of 90% or higher

The app should have the following functionality:

- The app should be able to produce a variety of time series plots of relevant summary statistics against time.
- The user should be able to select from which weather variable and from which stations (at most 5) the quantities to be shown.
- The user should be able to decide what aggregation to plot. This should include the following aggregations:
  - Raw hourly data (no aggregation);
  - Daily averages;
  - Monthly averages;
  - Daily maxima;
  - Daily minima;

- The user should be able to choose how time is to be handled and what the x-axis should represent. Choices should include
  - Calendar time;
  - Day or hour within the week (going from 0 to 7 (days) or 0 to 168 (hours)); and
  - Hour in the day (going from 0 to 24 (hours)).

  For calendar time the plot should be a line plot, but for the other two choices the plot should be a scatter plot.
- Make sure that suitable axis labels and legends are used.
- The app should also show a second plot with the location of the weather station(s). You can use the package `maps` or `ggmaps` for this, or any other mapping library of your choice. The code below shows the location of all measuring stations together with a map of the United Kingdom (assuming the data from `Sites.csv` is stored in a data frame `sites`).

```
  maps::map("world","UK")
points(sites$Longitude, sites$Latitude, pch=16, col="red")
```

- Below the plot there should be a table providing a daily mean summary of all the weather variables for each site selected for the last seven days of data available (this should be from November 24th to November 30th).
- The user should be able to download the table as a CSV file and the plots and table together as a Word document (generated using `rmarkdown`).
- For a given weather site selected by the user, report the Hutton criteria on a monthly basis (which can be selected by the user) for each day of the month. You can report this either graphically or in a table. (For this final part, you may want to create a separate tab in your app to report the criteria if you find this easier).

You can either use classical R plots or `ggplot2` for any plots you create.

Coding hints

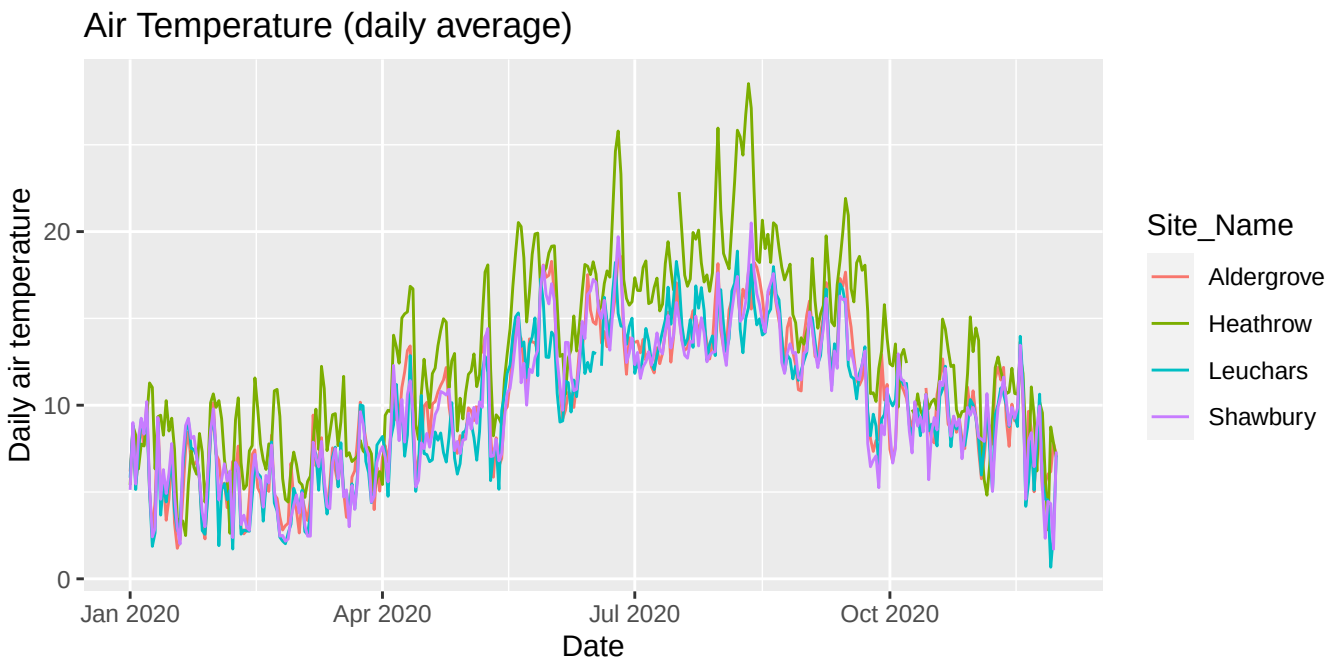
- You may find your app runs very slowly if you try to read all the data into R in one go. One way to deal with this is to read in the data only when it is needed. Once the user has chosen the weather stations, at most five CSV files need to be read into memory. To minimise the storage space taken up by the data on disk, you can convert each CSV file to an RData file.
- You may find the `lubridate` package useful for handling date-time data (`https://lubridate.tidyverse.org/`). You may find functions such as `decimal_date` useful for some calculations. In each weather station CSV file, aggregations by day, hour and month have been included as columns which you may find useful when creating plots.
- The code below renders the file 'report.Rmd' (in the same directory as your Shiny app) as a Word document and makes it available to download.

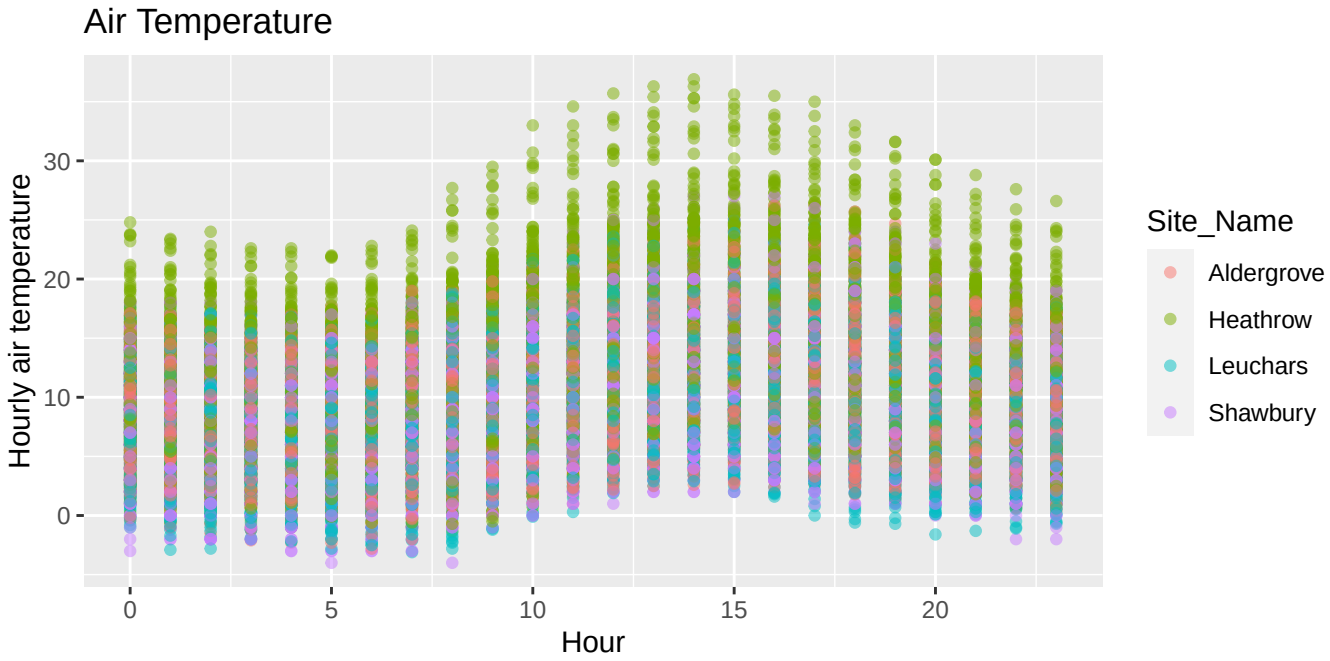
```
  output$downloadReportButton <- downloadHandler(
    filename = "report.docx",
    content = function(file) {
      render("report.Rmd", output_format="word_document",
             output_file=file, params=list(a=input$a))
      #                                  ^^^^^^^^^^^^^^^^^^^^^^
      #                                  input$a now available as params$a
      #                                  in rmarkdown
      #                                  (remember to define parameters in
      #                                   header of Rmd file)
    }
  )
```

  Instead of passing on objects to `rmarkdown` using `params` you can also store them in a temporary RData file in your Shiny app and then load this file in the `rmarkdown` document.

  If there are functions you want to use both in the Shiny app and in your `rmarkdown` report, consider placing them in an separate R file. This R file can then be included using `source`. (Inside the Shiny app you have to use `source` with the additional argument `local=TRUE`).

Examples of plots





What will be assessed?

The app functionality and the code implementation will be assessed. The following scheme will be used to assess your submission.

A bonus 2 marks are available for formatting and aesthetics. It is important to note that your Shiny app should be able to perform the key functionality highlighted and this should be your priority.

Please note, the maximum number of marks that can be achieved for this assignment is 20 marks.

A Teams channel will be set up for you to pose any questions on the project. Here you may ask questions on general Shiny implementation and high level questions on plotting. Please **do not** share any code directly related to your app or any code which answers components of the assessment.

Criterion / "Unit test"	Marks if correct
Does the user interface have the required functionality?	4 marks
Aggregations calculated correctly?	3 marks
Main plot shown as described?	2 marks
Plot of map shown as described?	1 mark
Data shown in table and downloadable?	2 marks
Report generation	2 marks
Hutton Criteria calculated correctly and reported?	4 marks
Code clear and well organised?	2 marks
BONUS - Use of advanced aesthetic features or functionality	2 marks