Guide 1: Obtaining State Weather Station Data

Marc Los Huertos

August 30, 2025 (ver. 1.03)

1 Introduction

1.1 Goals

Using a list of active weather stations in the United States, we will download the data from atleast stations and write weather station data csv files.

1.2 Selected History of Climate Science

Geologists have known the climate has been changing over the Earth's history. But what causes these changes has been a major research area for over 100 years. There are numerous drivers that contribute to changing climates – including the arrangement of the continents on the planet, the distance to the sun, energy generated by the sun, volanic activity, and the composition of the Earth's atmosphere.

It's the last one that we'll spend time because the Earth's temperature are changing pretty dramatically over the last 100 years and the cause is no mystery – the human activity that has released carbon dioxide (CO_2) into the atmosphere. The two main sources of CO_2 is from land use change, e.g. deforestration, and the burning of fossil fuels, e.g. coal, oil, and natural gas.

The first person to propose the role of CO_2 on the Earth's atmosphere was a Swedish scientist Svante Arrhenius, who figured out that CO_2 absorbs infarred light (Rodhe et al., 1997). Moreover, he deduced that the Earth's temperature was actually warmer than it might otherwise be if CO_2 was not part of the Earth's atmosphere.

1.3 Why Look at Individual Stations?

I don't believe there is a single, perfect way to analyze and communicate climate change. However, one of the strengths of the global network of weather stations is that they capture weather as experienced by local communities. While individual stations may not always reflect broader regional or global trends, they provide a crucial mechanism for linking local experiences to larger-scale climate processes.

That said, some individuals may focus solely on local patterns and remain unconvinced of the broader context. For those audiences, alternative approaches to communicating climate data may be more effective. Additionally, some may deliberately use local patterns to dismiss or ignore trends observed in other regions.

Moreover, the impacts of climate change are highly specific to each region. Even when individuals understand how climate change affects their own community, they may struggle to grasp how differentand often more severeits effects are on other populations, particularly those that are more vulnerable.

Recognizing these challenges, we can approach this project with an awareness of these limitations and aim to address them in later stages.

1.4 Approach

1.4.1 NOAA Data Records

The US National Oceanic and Atmospheric Administration (NOAA) maintains several sources of digital weather data from the USA and beyond. These data have been collected from stations around the country to support a wide range of human activities that include farming, aviation, shipping, and even armed conflict.

At various times, these records have been used to evaluate long-term climate change with varying success. Without a doubt, these data are not perfect, but they remain that foundation of an effective adn professionally maintained environmental monitoring program that engenders integrity, even when facing budget cuts.

We use these data to select for a station with a long record for each state in the USA. Future projects might evaluate the record for stations around the world, but we will see about that.

1.4.2 R Programming Language

R is an open source programming environment that has become one of the most popular tools for statisticians and data scientists. Capitalizing on the open source framework, a wide range of libraries or packages have been developed to facilitate data processing, analysis, and graphical displays. I created a bunch of functions, which can then be used to collect Let's try this and see how it works! Please read as you go, there subtle things that can go wrong and you need to know what the code is supposed to do.

1.5 Revised Document Structure

Based on some feedback in the last year, I re-organized the document with bullets of the steps and then the code. After the sections that show you how to process the data, I added descriptions about "what the code does and why".

I suggest you look at these sections after running the code, but I put them in the end so it doesn't distract you from the main steps.

There are two weaknesses with this approach that I can see. First, if the code does not work, you will have a really hard time figuring out why, since you have no idea what the code is supposed to do. Second, if the code works, you will not really be learning any R coding. Just how to copy and paste.

1.6 Preparing R for the Project

I suggest you do the following to prepare:

1. Create a folder for the project, I suggest "Regional_Climate_Trends". Note that there are no spaces in the folder name.

2. Next download Marc's Template (an R markdown file) from `Canvas`. We will use to structure the project, where you can record both your code and explanation of what the code does. To avoid issues later and to make sure this step works, please see the details see Section **??**).

3. Download the file oldest.activestation.csv from the Canvas page and upload it to the directory you created in the previous step.

4. Download the file called Guide1functions.R from the Canvas page and upload it to the folder ("Regional_Climate_Trends") you created in the previous step.

5. Open the file in Rstudio and run the code, using the "source" action item near the top, right of the editor window.

6. Create a folder inside (under) the Regional_Climate_Trends folder called "Data". This will be the location of the data you download.

2 R Functions for Getting Weather Station Data

2.1 Creating a Record of Your Commands w/R Markdown

To record your history of R code, I suggest use an Rmarkdown file to keep your code handy.

Normally, we use R Studio to creata a new file and select R Markdown. Then we define a title, author, and output format as PDF. Then save the file in the folder you created earlier (Figure 1).

However, for this prooject, I suggest you use my template and name the file "ST_Climate_Trends–Surname.Rmd", where ST is the state you selected and Surname is your last name.

Knit the file to create a PDF. Look at the PDF to see how various parts of the Rmd file were knitted. Notice that there are generally two sections: text and R chunks. As you write th R code, you will can record your code in the R chunks AND develop the output into a document for future reference.

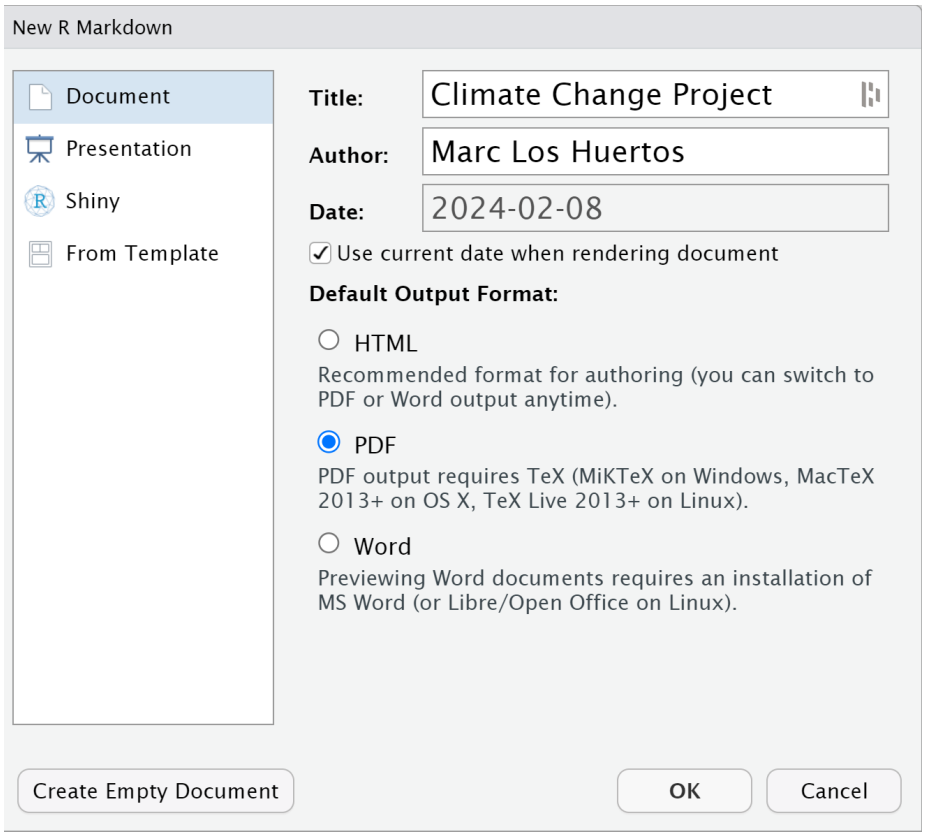


Figure 1: Creating a new R Markdown file.

Look carefully at the Rmd file and take note of the following: First, there are chunks of R code separated from the text by a line of three backticks. Second, the R code is written in the chunks and the output is written below the code. Finally, there is a set of backticks at the end of the code chunk. This section is gray and is where you can record your Rcode.

I suggest you modify the text and use it to guide the process and record issues you have so Marc can make the process eaiser for each iteration!

2.1.1 R Environment and knitr

When you knit, R starts a new environment and runs the code in the chunks. But that means the new environment doesn't have the data you might create in the console, which is then in the R environment. So, I suggest you use the R markdown to double check that everything is working well in both the console and in the knit environment.

Use the Rmd file to record your R code and comments. This will help you

remember what you did and why. See MarcsTemplate.Rmd as a skeleton file.

Unfortunately, knit creates it's own session of R – so the stuff we do in the console isn't available in when you knit. So, we'll have to get R to explicitly load the functions wit each time we knit. It's not hard, but important.

I suggest you start with the R block labelled cars – removed the code and replace it with the following code:

2.2 Marc's Custom Functions: Guide1functions.R

These functions are designed so you don't have to type all the code! But check to see if they work for you, because they can (apparently) break really easily.

From the Canvas page, go to the Guide1functions.R code and download the file to your computer. Then upload the file to a directory in Rstudio.

By placing this in the Rmd file you can run the code and the functions will be available to you. But you must include the whole path (hierarchy of folders) to the file (See Section 3.1 for some tips on how to do this). If the file is in the same directory as the Rmd file it will look something like this, where the path can be defined by using `file.choose()` in the console:

```
source(``XX/..XXX/Regional_Climate_Trends/Guide1functions.R'')
```

NOTE: R is case sensitive. AND the filepath and file to be in quotes.

2.3 Function Descriptions and Use

The following functions are used to select the weather stations, download data, and write weather station data csv files.

Subset Inventory Data with my.state The inventory data is a list of all the weather stations in the USA, US Territories, and Canada. This function will subset the data to only include the stations in the state of interest.

This function requires two parameters to be set: filename and my.state. See Section 4.1 for more details.

I suggest you use file.choose() and assign filename.csv to the path and name of the csv file in your project directory.

Example of how to use the function:

```
my.state <- "NV"
filename.csv <-
  "/home/mwl04747/RTricks/05_Regional_Climate_Trends/stations.active.oldest.csv"
my.inventory <- readInventory.fun(filename.csv, my.state)
```

Creates a dataframe called my.inventory that contains the inventory data for the state/territory of interest. You should see that in the R environment. Note: The filename is cut off because my path is too long!

Download and Read Selected Weather Station Data into R This function will download the weather station data from the NOAA website and read it into R.

Function: downloadStations.fun()

This functions requires two parameters: data path and name of inventory dataframe (my.inventory). The data path is a folder that you can always get your station data from if you need it. For example, if you need to update the data, you can just download the data from the NOAA website and then read it into R. For more information on this function see Section 4.2.

The path to your data folder will be different, I created a directory to separate my data by class year, you don't need that! But it is nice to keep neat. Use the "Data" folder you created when you set up the project.

An example of how I used the function (for my data and state) In this case, I am using Nevada:

```
datapath = "/home/mwl04747/RTricks/05_Regional_Climate_Trends/Data/FA25/"
downloadStations.fun(datapath, my.inventory)
```

Your data path will NOT be the same as mine. First, you can see your working directory with `getwd()`. But it probably doesn't include the folder you are working in. I suggest you use file.choose() and select any file in the project folder.

Then delete the file name (remove the "stations.active.oldest.csv"!!) and add the "/Data/", so it might look something like this:

```
datapath = "/home/Regional_Climate_Trends/Data/"
```

The / at the end turns out to be important, so include that.

This will download the data from the NOAA website and read it into R then written as compresseed gz and uncompressed czv files in your data folder. Check to see that you have both csv.gz and .csv files in your data folder.

You're done processing the code in this Guide 1! I suggest you review the code below and then go to Guide 2.

3 Explaining the Process and Code

3.1 Defining Path & Read Data

First, we install some packages and read in the data. I suggest you create a folder for the project (I created one called "05_Regional_Climate_Trends"). To get the working directory and filename, we can use the file.choose() function from the console:

```
> file.choose()
```

Then paste the result in the Rmd file to track the results to define the file name:[1]

```
filename = "home/mwl04747/RTricks/05_Regional_Climate_Trends/stations.active.oldest.csv"


# OR what I use! see the footnote.
library(here)
filename = here("05_Regional_Climate_Trends", "stations.active.oldest.csv")
```

Note: the filename is define as the path to find the file and the filename – the "path" is a hierarchical structure of folders to find the path, e.g. room/cabinet/drawer/file.

3.2 Selecting Weather Records by State

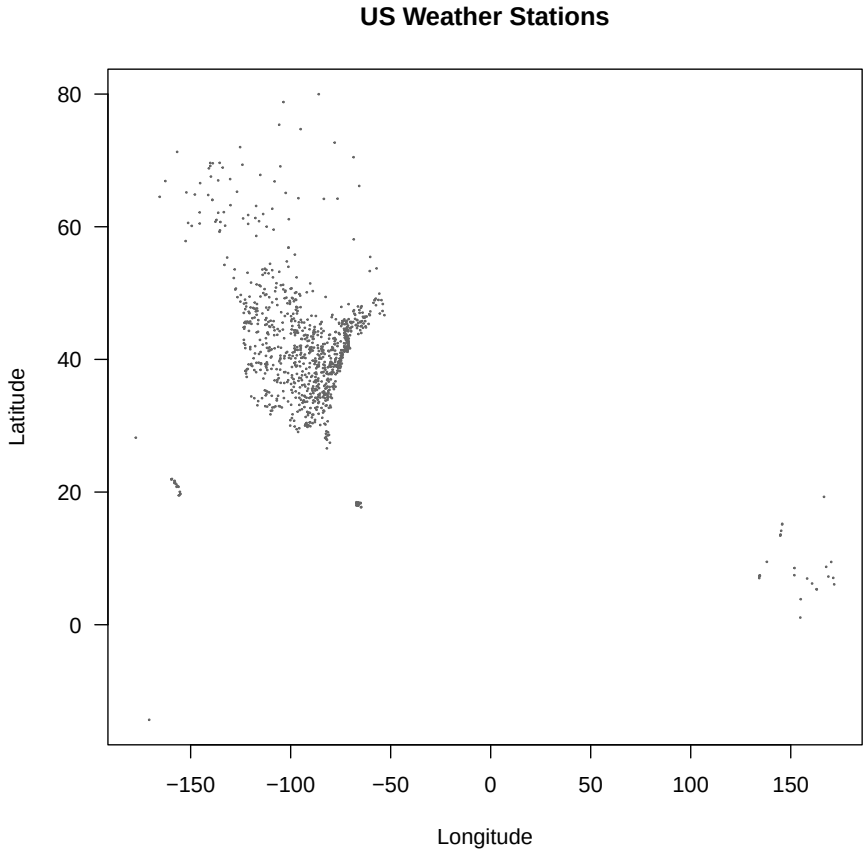
3.2.1 Map US Weather Stations

Here's a map of the weather stations in the dataset. Pretty lame map! We'll make a better one later.[2]

```
stations.active.oldest =
  read.csv(paste0(here(), "/05_Regional_Climate_Trends/",
                  "stations.active.oldest.csv"))
plot(stations.active.oldest$LONGITUDE,
     stations.active.oldest$LATITUDE,
     xlab = "Longitude",ylab = "Latitude",
     pch=20, cex=0.2, col='gray40', las=1,
     main = "US Weather Stations")
```

[1] I use a function here() from the "here library", but it's a bit tricky to use and you can use that if you prefer. Notice that each folder is separated by a comma.

[2] Evelyn/Brody: This is a good change to see how to use R for map making! First we need to transform that data.



There are numerous ways to analyze temperature records, where stations can be analyzed individually or records could be sampled and analyzed in spatially in grids. Each of these are valid approaches depending on the question to be addressed.

Here are the questions we will address:

- What stations have the longest meteorological records in the USA?
- Can we determine the reliability of these stations?
- Finally, is there a temperature trend?

3.3 Subset Station Inventory by State

This uses the stations.active.oldest file to download the data from the NOAA website based on the state you have choose.

The URL for a description of the data is: https://www.ncdc.noaa.gov/cdo-web/datasets/GHCND/stations/GHCND:USW00023230/detail.

4 Function Details

4.1 readInventory.fun

This function reads the inventory of stations and subsets the data by state.

Here's the function:

```
## function (filename, my.state)
## {
##     inventory.active.oldest <- read.csv(filename)
##     my.inventory = subset(inventory.active.oldest, STATE == my.state)
##     return(my.inventory)
## }
```

4.2 downloadStations.fun

This function uses the list of 15 weather stations in your state and requests the data to be downloaded from NOAA. As each compressed files arrives on the college's server it is uncompressed (gz –¿ csv). Pretty handy, eh? Some other stuff that I needed to do was define the column names and then assign them when creating the data frame.

Here's the function:

```
## function (datafolder, my.inventory = my.inventory)
## {
##     colnames <- c("ID", "DATE", "ELEMENT", "VALUE", "M-FLAG",
##         "Q-FLAG", "S-FLAG", "OBS-TIME")
##     for (i in 1:nrow(my.inventory)) {
##         url = paste0("https://www.ncei.noaa.gov/pub/data/ghcn/daily/by_station/",
##             my.inventory$ID[i], ".csv.gz")
##         download.file(url, paste0(datafolder, my.inventory$ID[i],
##             ".csv.gz"), quiet = FALSE, mode = "w", cacheOK = TRUE)
##         station.temp <- read.csv(paste0(datafolder, my.inventory$ID[i],
##             ".csv.gz"), header = FALSE)
##         names(station.temp) <- colnames
##         filename = paste0(datafolder, my.inventory$ID[i], ".csv")
##         write.csv(station.temp, filename, row.names = FALSE)
##         print(paste("Index (Loop) ", i, " Completed."))
##         print("NOAA site can stall -- if the loop errors out, try again.")
##     }
##     print("Think about something you are grateful for today!")
```

```
##     write.csv(my.inventory, paste0(datafolder, "my.inventory.csv"),
##         row.names = FALSE)
## }
## <bytecode: 0x2c739c0>
```

How do you determine the data path?

5 Trouble Shooting and Workarounds

I will be getting all the guides working before working on this! But if there are errors with the custom function, this is where workarounds will be described! Please Slack me and mentors if you have any problems!

5.1 NA.csv as a duplicate?

I am not sure why, but in the downloadStations.fun(), the file is saved as NA.csv. I have no idea why. If you find this, let me know, I'd like to figure out if their is a bug in the code creating this csv. I notices it with some states but not others. It doesn't seem to impact the analysis, but if you find a NA.csv in your folder, please let put a note in the Rmd file to let me know.

5.2 Download gz files with folder concantentated names

If the downloaded data have names that include the name of the folder, then I suspect you are missing a "/" in the data path. It's an easy fix, but you might want to delete the files and re-run the code to download again with a corrected path, thus name.

6 rNOAA – Unsupported and Deprecated Library

The rNOAA package has been deprecated, so all the 2000-2022 code is no longer working. I have commented out the text – but should I need to re-run the code, I have some of the description in the Rmd file.

References

Rodhe, H., Charlson, R., and Crawford, E. (1997). Svante arrhenius and the greenhouse effect. *Ambio*, pages 2–5.