Preparing Regional Weather Data Analysis Project

Marc Los Huertos

February 7, 2024 (ver. 0.74)

1 Background

1.1 Project Goals

Create a public product (video) that explains climate change trends in a state; what the state is doing to mitigate climate change; and what the state and it's residents could do to improve its efforts to mitigate climate change.

1.2 Project Stages

Project Overview (This Document) A brief overview of the project and the steps to complete it. In addition, this document, when run selects the oldest active station ID for each state (and territory). Students don not need to run this document, but creates an updated list for them each time it is run. In addition, this document explains some of the R code used to create this list as background information.

Guide 1 Data Collection (Download station data from the web and read into R.)

Guide 2 Data Processing (Clean and Pre-process Data)

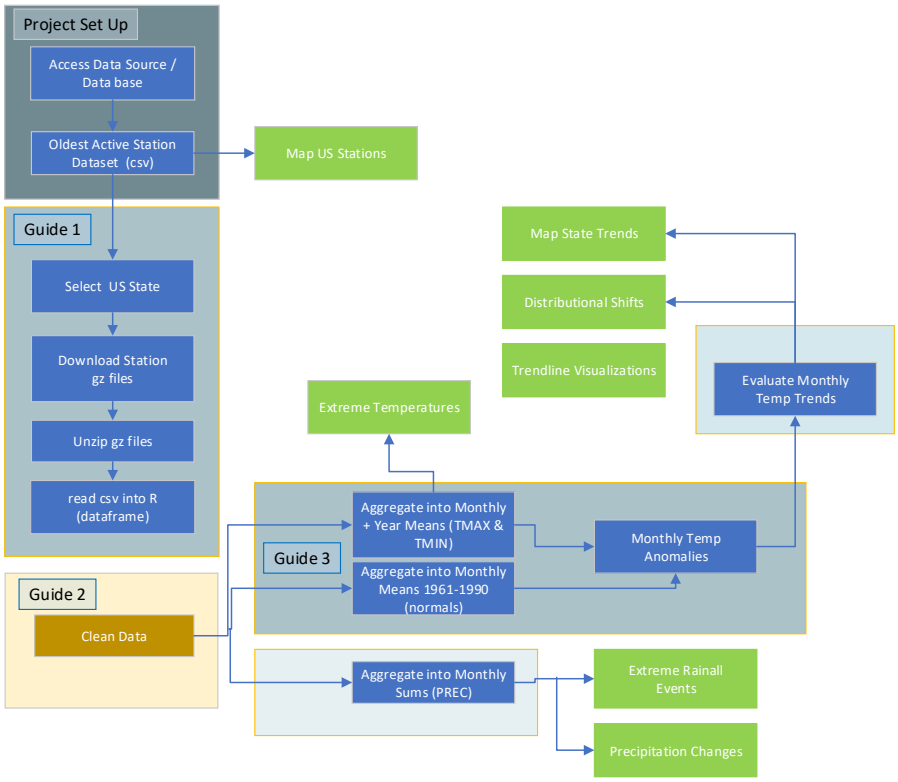
Guide 3 Data Analysis (Use R to analyze data (means, trends, etc))

Guide 4 Data Visualization (Create visualizations of data)

Guide 5 Communicating Project Results (Create a video that explains the data and the results of the analysis.)

At this point, I have created a DRAFT visual flow chart that displays each stage. I will be adjusting this chart as we progress and refine the R code and project guides.

Figure 1: A flow chart of the project stages.



1.3 Approach

I have created several Guides 1 through 4. Each guide is a Rmarkdown document that explains the steps to complete the guide. The guides are designed to be completed in order, but it is possible to skip a guide if the data is already available.

This particular document has NO code for you to run, but document the process that I used to obtain the data and create a list of stations from you to select from. So, please read this as an informational document and not something to follow. However, there are some insights here about how to find the data and the code that I used to prepare for the project for the class.

2 Selecting US Weather Station with Robust Records

2.1 Global Weather Station Data

The GHCNd is the primary source of weather station data. The data is available from the National Centers for Environmental Information (NCEI) at the following URL: https://www.ncei.noaa.gov/pub/data/ghcn/daily/. The data is available in a variety of formats, including .csv, .dat, and .txt.

2.2 Goals for this Document

This document selects the oldest active weather station for each state (and territory) in the US. The station inventory is available as a .txt file. The file is a fixed width file, which means that each column has a specific width. The file is available at the following URL: https://www.ncei.noaa.gov/pub/data/ghcn/daily/ghcnd-inventory.txt.

But you do not need to follow the steps in this document. This document is for background information only and show how I prepared the list of stations for you to select from.

Note, it maybe that there are too many missing values in one or more of your stations, then we can use this document to find additional stations, if they exist in the database.

2.3 Download Station Inventory

The station inventory is available as a .txt file. The file is a fixed width file, which means that each column has a specific width. The file is available at the following URL: https://www.ncei.noaa.gov/pub/data/ghcn/daily/ghcnd-inventory.txt.

```
library(here)

## here() starts at /home/mwl04747/RTricks
```

```r
# Get Stations Data (Inventory)
inventory = read.table("https://www.ncei.noaa.gov/pub/data/ghcn/daily/ghcnd-inventory.txt")

# Define Variable Names because there is no header in the file
inventory_names = c("ID", #            1-11   Character
                    "LATITUDE", #     13-20   Real
                    "LONGITUDE", #    22-30   Real
                    "ELEMENT", #      32-35   Character
                    "FIRSTYEAR", #    37-40   Integer
                    "LASTYEAR") #     42-45   Integer

# Assign Variable Names to inventory dataframe
names(inventory) = inventory_names

# Check the structure of the data
str(inventory)
```

```
## 'data.frame': 747163 obs. of  6 variables:
##  $ ID       : chr  "ACW00011604" "ACW00011604" "ACW00011604" "ACW00011604" ...
##  $ LATITUDE : num  17.1 17.1 17.1 17.1 17.1 ...
##  $ LONGITUDE: num  -61.8 -61.8 -61.8 -61.8 -61.8 ...
##  $ ELEMENT  : chr  "TMAX" "TMIN" "PRCP" "SNOW" ...
##  $ FIRSTYEAR: int  1949 1949 1949 1949 1949 1949 1949 1949 1949 1949 ...
##  $ LASTYEAR : int  1949 1949 1949 1949 1949 1949 1949 1949 1949 1949 ...
```

2.4 Selecting Active Weather Stations with Maximum Daily Temperature Readings

```r
# Subset data for TMAX  (Max Temperature) Element
inventory.TMAX = subset(inventory, subset=ELEMENT=="TMAX")

# Check the structure of the data
str(inventory.TMAX)
```

```
## 'data.frame': 40395 obs. of  6 variables:
##  $ ID       : chr  "ACW00011604" "ACW00011647" "AE000041196" "AEM00041194" ...
##  $ LATITUDE : num  17.1 17.1 25.3 25.3 24.4 ...
##  $ LONGITUDE: num  -61.8 -61.8 55.5 55.4 54.7 ...
##  $ ELEMENT  : chr  "TMAX" "TMAX" "TMAX" "TMAX" ...
##  $ FIRSTYEAR: int  1949 1961 1944 1983 1983 1994 1973 1973 1966 1973 ...
##  $ LASTYEAR : int  1949 1961 2024 2024 2024 2024 1992 2020 2021 2020 ...
```

```r
# Subset Active Stations (observations that include 2022 and more recent)
inventory.TMAX = subset(inventory.TMAX, subset=LASTYEAR>=2022); str(inventory.TMAX)
```

```
## 'data.frame': 12745 obs. of  6 variables:
##  $ ID       : chr  "AE000041196" "AEM00041194" "AEM00041217" "AEM00041218" ...
##  $ LATITUDE : num  25.3 25.3 24.4 24.3 36.7 ...
##  $ LONGITUDE: num  55.52 55.36 54.65 55.61 3.25 ...
##  $ ELEMENT  : chr  "TMAX" "TMAX" "TMAX" "TMAX" ...
##  $ FIRSTYEAR: int  1944 1983 1983 1994 1940 1940 1958 1886 1878 1880 ...
##  $ LASTYEAR : int  2024 2024 2024 2024 2024 2024 2024 2024 2024 2024 ...
```

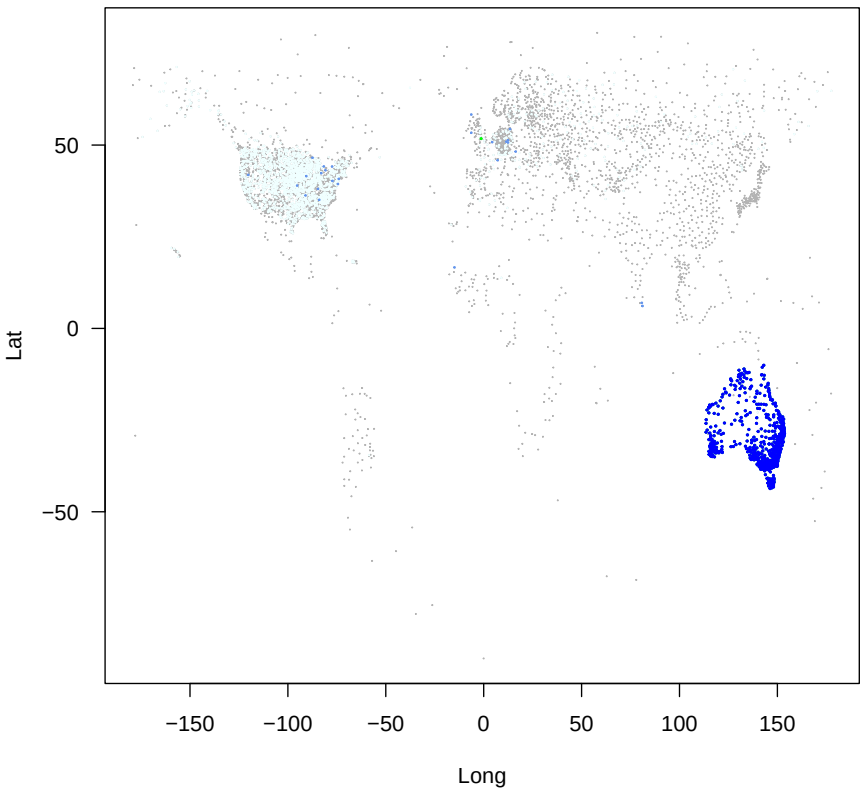
2.5 Using the GHCND-station Dataset to Select US Stations

The inventory has a list of stations and map coordinates (latitude and longitude). However, it's not easy to select a region, like a state, from the latitude and longitude values. Thus, we need to merge the inventory with a dataset that includes state names and merge them based on the station ID.

It's a bit strange, but the dataset, GHCNd includes US states and Canadian Provinces, plus various territories of the US.

```
station_names = c("ID",             #  1-11   Character   11
                  "LATITUDE",       # 13-20   Real        8
                  "LONGITUDE",      # 22-30   Real        9
                  "ELEVATION",      # 32-37   Real        6
                  "STATE",          # 39-40   Character   2
                  "NAME",           # 42-71   Character
                  "GSN FLAG",       # 73-75   Character
                  "HCN/CRN FLAG",   # 77-79   Character
                  "WMO ID"          # 81-85   Character
                   )

# Read ghcnd-stations.txt with fixed width format
Stations = read.fwf("https://www.ncei.noaa.gov/pub/data/ghcn/daily/ghcnd-stations.txt",
        col.names=station_names, fill=2,
        widths=c(11, -1, 8, -1, 9, -1, 6, -1, 2, -1, 30, -1, 3, -1, 3, -1, 5 ))

# NOTE: Got to be a better way to get these data!

str(Stations) # Missing State Name

## 'data.frame': 125988 obs. of  9 variables:
##  $ ID          : chr  "ACW00011604" "ACW00011647" "AE000041196" "AEM00041194" ...
##  $ LATITUDE    : num  17.1 17.1 25.3 25.3 24.4 ...
##  $ LONGITUDE   : num  -61.8 -61.8 55.5 55.4 54.7 ...
##  $ ELEVATION   : num  10.1 19.2 34 10.4 26.8 ...
##  $ STATE       : chr  "  " "  " "  " "  " "  " ...
##  $ NAME        : chr  "ST JOHNS COOLIDGE FLD         " "ST JOHNS                    " "
```

Figure 2: A plot of the global weather stations. Note the increase in stations over time and spatial distribution. Austrailia has most of it's stations with 1750 start dates, but I suspect most of these stations have lots of missing data.



```
##  $ GSN.FLAG    : chr  "   " "   " "GSN" "   " ...
##  $ HCN.CRN.FLAG: chr  "   " "   " "   " "   " ...
##  $ WMO.ID      : int  NA NA 41196 41194 41217 41218 40930 40938 40948 40990 ...

# Now we'll get the state names for the states.
State_names = c("STATE", #         1-2    Character 2
                "STATE_NAME") #         4-50    Character 46
States = read.fwf("https://www.ncei.noaa.gov/pub/data/ghcn/daily/ghcnd-states.txt",
                  col.names=State_names, fill=2,
                  widths=c(2, -1, 46))

str(States)

## 'data.frame': 74 obs. of  2 variables:
##  $ STATE     : chr  "AB" "AK" "AL" "AR" ...
##  $ STATE_NAME: chr  "ALBERTA" "ALASKA" "ALABAMA                                        " "

# Merge the two datasets
StateIDs = subset(Stations, select=c("ID", "STATE"))
StateIDs = merge(StateIDs, States, by="STATE") # Add State Names

temp.TMAX = merge(inventory.TMAX, StateIDs, by="ID")
# Note: Some outer join would be better, to be completed later.

# Remove Stations that STATE = blank!
stations.USCan = subset(temp.TMAX, subset=(STATE!="  "))
```

2.6 Select Active Stations

How many stations are in the state? 'r nrow(stations.USCan)'!

```
stations.active = subset(stations.USCan, subset=LASTYEAR>=2022)
str(stations.active)

## 'data.frame': 7751 obs. of  8 variables:
##  $ ID        : chr  "AQW00061705" "CA001011500" "CA001012055" "CA001012475" ...
##  $ LATITUDE  : num  -14.3 48.9 48.8 48.4 48.4 ...
##  $ LONGITUDE : num  -171 -124 -124 -123 -123 ...
##  $ ELEMENT   : chr  "TMAX" "TMAX" "TMAX" "TMAX" ...
##  $ FIRSTYEAR : int  1966 1979 1960 1997 1991 1991 2007 1972 1970 1996 ...
##  $ LASTYEAR  : int  2024 2024 2023 2024 2024 2024 2024 2024 2022 2024 ...
##  $ STATE     : chr  "AS" "BC" "BC" "BC" ...
##  $ STATE_NAME: chr  "AMERICAN SAMOA" "BRITISH COLUMBIA" "BRITISH COLUMBIA" "BRITISH COLUM

nrow(stations.active)

## [1] 7751
```

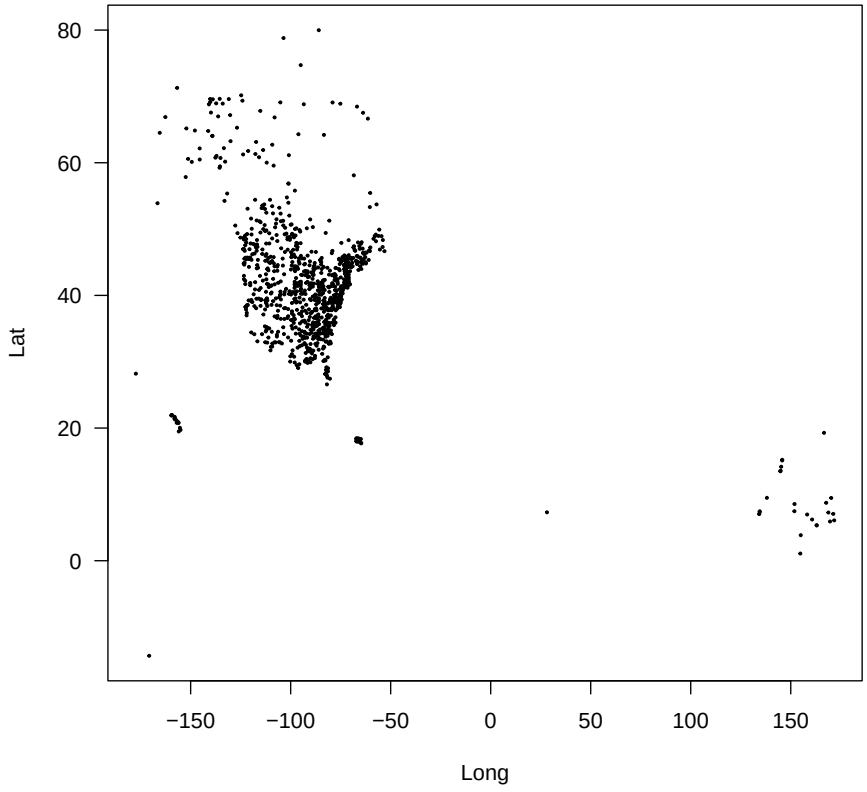
2.7 Select Upto 15 Stations for Each State

To accomplish this, I need to do a loop to select the first 5 stations for each state.

```
# Loop to select 15 stations for each state
#stations.active.oldest = subset(stations.active, subset=FIRSTYEAR==min(FIRSTYEAR))
i=10
for(i in 1:nrow(States)) {
  state.df = subset(stations.active, subset=STATE==States$STATE[i])
  if(nrow(state.df) >= 15) {
    state.df = state.df[order(state.df$FIRSTYEAR),][1:15,]
  }
  if(nrow(state.df) < 15){
    state.df = state.df[order(state.df$FIRSTYEAR),][1:nrow(state.df),]
  }
  if(i==1) {
    stations.active.oldest = state.df
  } else {
    stations.active.oldest = rbind(stations.active.oldest, state.df)
  }
}
```

3 Plot Results

```
plot(stations.active.oldest$LONGITUDE, stations.active.oldest$LATITUDE,
     pch=20, cex=.4, xlab="Long", ylab="Lat", las=1)
```



3.1 Next Steps

In Guide #1, we'll use the stations.active.oldest dataset to download the daily temperature and precipitation data for the "oldest, active stations".

```
# export file to csv
write.csv(stations.active.oldest,
    here("04_Regional_Climate_Trends",
    "stations.active.oldest.csv"))
```

Here's a Table of Task for the Project:

We use Table **??** to estimate the time and resources needed to complete the project.

Table 1: Probably Tasks and Time Estimates for the Project. Time is based on our estimate (x) and post project evaluation of time required (y) (Source: EA30 2024).

Task	**Time**	**Resources**	**Notes**
1. Select State	x (y)	Rstudio, R code and NOAA website	Guide 1
2. Download Station Data	x (y)	Rstudio, R code and NOAA website	Guide 1
3. Read Station Data into R	5 (y)	Rstudio, R code	Guide 1
4. Fix Dates Guide 2	x (y)	Rstudio, R code	
5. Transform Data	x (y)	Rstudio, R code	Guide 2
6. Subset for TMAX	x (y)	Rstudio, R code	Guide 3
7. Linear Model for TMAX	x (y)	Rstudio, R code	Guide 3
Evaluation of Model	x (y)	Rstudio, R code	Guide 3
Subset for TMIN	x (y)	Rstudio, R code	Guide 3
Linear Model for TMIN	x (y)	Rstudio, R code	Guide 3
Evaluation of Model	x (y)	Rstudio, R code	Guide 3
Subset for PRCP	? (y)	Rstudio, R code	Guide 3
Linear Model for PRCP	? (y)	Rstudio, R code	Guide 3
Evaluation of Model	? (y)	Rstudio, R code	Guide 3
Read EPA State Impact Report	? (y)	EPA website	Canvas Link
Explore Other Analysis Options	?0 (y)	Rstudio, R code	Guide 3
Create Presentation Ready Graphics	?0 (y)	Rstudio, R code	Guide 4
Scripting for Video	?0 (y)	Rstudio, R code	Guide 4
Make Video	?0 (y)	Rstudio, R code	Guide 4
Upload Video	? (y)	Rstudio, R code	Guide 4
Total	X (Y)		