Preparing the Regional Climate Trends Project

Marc Los Huertos

February 3, 2025 (ver. 1.03)

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 Rnw file weaces text and R code that creates an inventory active weather station IDs for each state (and territory). Students do not need to run the R code in this document, but it is available for reference and a learning tool.

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

Guide 2 Cleaning and Pre-Processing Weather Station Data (Convert date formats, clean missing data, etc.)

Guide 3 Analyzing & Visual Display of Climate Trends (Analyze data (means, trends, etc) and create compelling visualizations)

Guide 4 Climate Science Narratives — State Initiatives and Community Belonging (Create creative stories that explain climate change trends)

Guide 5 Communicating Climate Stories – Combining Imagery and Audio (Edit a video that explains the data and the results of the analysis.)

At this point, I created a DRAFT visual flow chart that displays some of the stages (Figure 1). I will be adjusting this chart as we progress, adding detail. Moreover, I'll refine the R code and project guides if we found ambiguities.

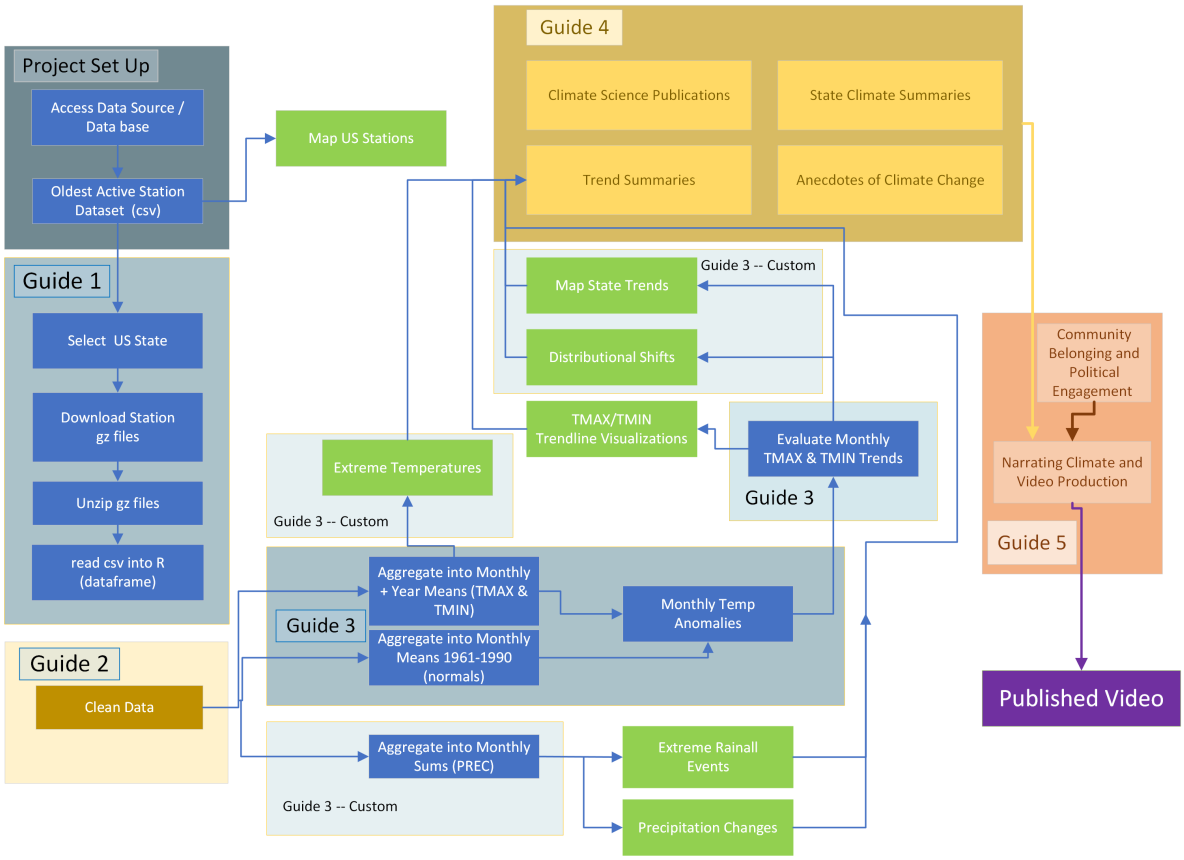


Figure 1: A flow chart of the project stages. Custom codes can be created with Marc's help.

1.3 Approach

I created several Guides 1 through 5. Each guide explains steps to complete the project and are designed to be completed in order.

Our approach prioritizes documentation and transparency in data sourcing and preparation while also providing executable code. We focus on clearly identifying data sources, detailing the process for obtaining the data, and curating relevant datasets for analysis. My intention is to promote a measure of independence to verify and replicate the data acquisition process.

2 Selecting US Weather Station with Robust Records

2.1 Global Weather Station Data

The GHCNd[1] 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 stations 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.

This particular document has **no code** for you to run, but documents the source of the data; explains process to obtain the data; and creates a list of weather stations to be used in Guide 1. Please read this guide as an informational document about our data source. The document includes some insights about how to find climate data and the code that I used to prepare for the project for the class.

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

# 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
```

[1] Global Historical Climatology Network daily is maintained by the US. Monthly and Annual mean data are also available.

```
                      "LASTYEAR") #     42-45   Integer

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

# Check the structure of the data
str(inventory)

## 'data.frame': 765236 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 (and inactive) Weather Stations with Maximum Daily Temperature Readings

Selecting the stations that are both active and contain a basic measure of the maximum temperature (TMAX) is a good place to evaluate the quality and quantity of the data.

```
# 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': 40428 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 2025 2025 2025 2025 1992 2020 2021 2020 ...

# Subset Active Stations (observations that include 2024 and more recent)
active.TMAX = subset(inventory.TMAX, subset=LASTYEAR>=2024); str(inventory.TMAX)

## 'data.frame': 40428 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 2025 2025 2025 2025 1992 2020 2021 2020 ...

# Subset Inactive and Old Stations)
inactive.TMAX = subset(inventory.TMAX, subset=(FIRSTYEAR <= 1850 & LASTYEAR<=2023)); str(ina

## 'data.frame': 3 obs. of  6 variables:
##  $ ID       : chr  "CA006158350" "EZE00100082" "ITE00100554"
##  $ LATITUDE : num  43.7 50.1 45.5
##  $ LONGITUDE: num  -79.4 14.42 9.19
##  $ ELEMENT  : chr  "TMAX" "TMAX" "TMAX"
##  $ FIRSTYEAR: int  1840 1775 1763
##  $ LASTYEAR : int  2003 2005 2008
```

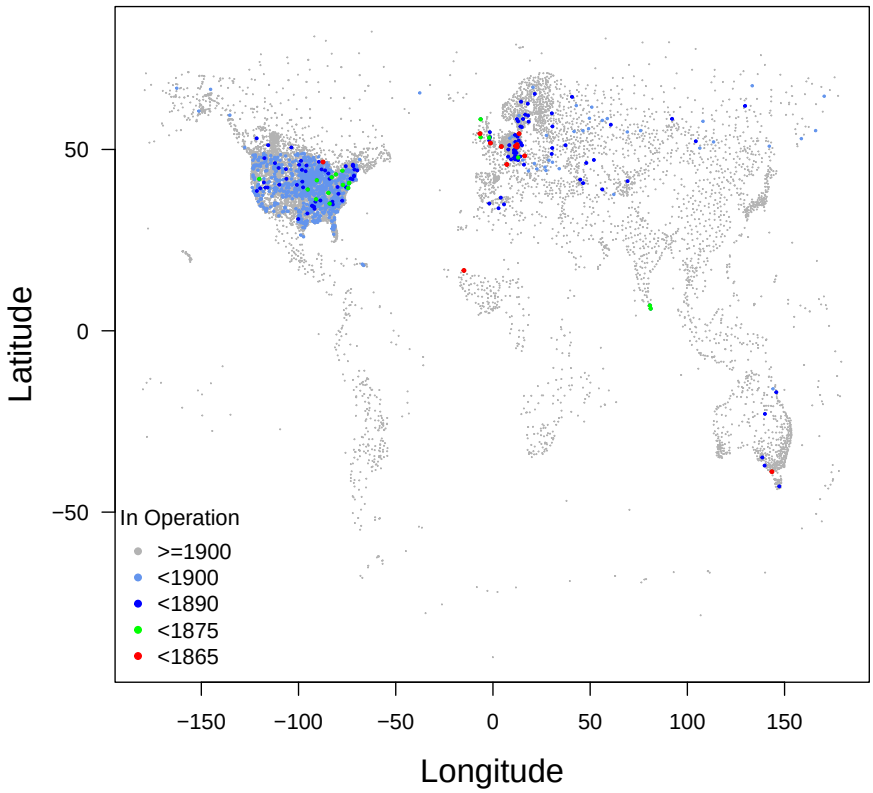
2.5 Subsetting the GHCNd Inventory

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.

The dataset, GHCNd includes US states and various territories of the US and, oddly, Canadian Provinces.

```
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': 129655 obs. of  9 variables:
##  $ ID          : chr  "ACW00011604" "ACW00011647" "AE000041196" "AEM00041194" ...
```

Figure 2: A plot of active global weather stations (GHCNd). Note the increase in stations over time and spatial distribution. It's a story of the European industrialization, US Expansion, and colonialism.



```
##  $ 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                        " "
##  $ 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(active.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

To ensure we can limit our data to stations that are active, we need to subset the data to include only stations that have data from 2024 and later. However, I relaxed the criteria to 2023 since there is a time lag for data to be assimilated for some wether stations.

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

```
stations.active = subset(stations.USCan, subset=LASTYEAR>=2023)
str(stations.active)
```

```
## 'data.frame': 7398 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 1996 1991 ...
##  $ LASTYEAR  : int  2025 2024 2024 2024 2024 2024 2024 2024 2024 2024 ...
##  $ STATE     : chr  "AS" "BC" "BC" "BC" ...
##  $ STATE_NAME: chr  "AMERICAN SAMOA" "BRITISH COLUMBIA" "BRITISH COLUMBIA" "BRITISH COLUM
```

```
nrow(stations.active)
```

```
## [1] 7398
```

2.7 Selecting Stations for Each State

To accomplish this, I created a loop to select the stations from each state. The loop selects the 15 oldest, active stations from each state. If there are fewer than 15 stations, it selects all the stations.

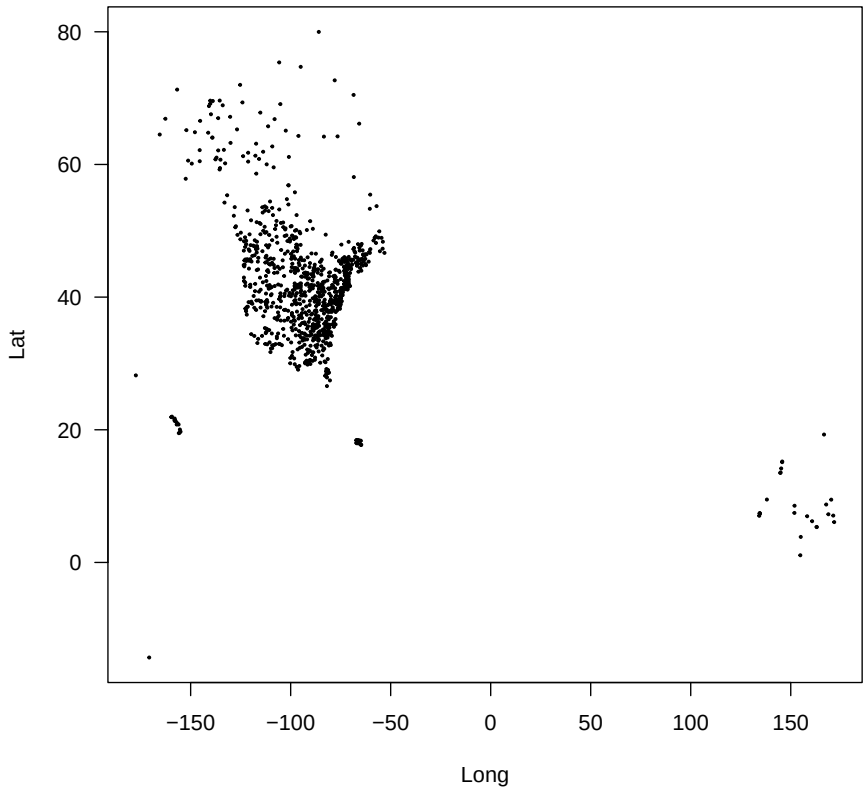
```
# Loop to select 15 stations for each state
#stations.active.oldest = subset(stations.active, subset=FIRSTYEAR==min(FIRSTYEAR))
i=10
for(i in 1:nrow(States)) {
  state.df = subset(stations.active, subset=STATE==States$STATE[i])
  if(nrow(state.df) >= 15) {
    state.df = state.df[order(state.df$FIRSTYEAR),][1:15,]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

2.8 Plotting Active Stations

At some point, I'd like to "map" the stations with actual state boundaries and a projection that make more sense. For example, what are the stations north of the Equator and 150 E in longitude?

Maybe I'll use the `ggplot2` package to plot the stations in the future. Transforming the data to a `sf` object would be a good idea, which is designed for mapping data.

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



2.9 Write the Active Stations to a CSV File

To save the active stations to a CSV file, we use the `write.csv` function.

```
# export file to csv
write.csv(stations.active.oldest,
    here("05_Regional_Climate_Trends",
    "stations.active.oldest.csv"))
```

3 Next Steps

3.1 Estimate Time for Project

Based on the class survey Table 1 estimates the time and resources needed to complete the project.

Table 1: Probably Tasks and Time Estimates for the Project. Time is based on our survey estimates. [1]Guide 1 can take extra time because NOAA data are not consistently formatted. [2] TBD.

Task/Guide	**Time (min)**	**Resources** (Notes)
Guide #1	x	Rstudio, R code, NOAA website[1]
Guide #2	x	Rstudio, R code
Guide #3	x	Rstudio, R code
Guide #4	x	EPA documents and other literature
Guide #5	x	EA Streaming/Video Booth
Total (hrs)	X	

3.2 Start Guide #1

In Guide #1, we use the **stations.active.oldest.csv** dataset to download the weather data for the "oldest, active stations."