HDD CDD

Audrey Smith

5/11/2020

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.6.3
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(tidyr)
library(stringr)
```

```
#Generate list of filepaths for heating and cooling data
heatFiles <- list.files(pattern = 'heating*')
coolFiles <- list.files(pattern = 'cooling*')

#Write function that can work for both heating and cooling to read in many data years and join t
hem with one another
readTempFiles <- function(fileList){
  i <- 1

  #Iterate through items 1-5 in file list - stop loop when i > length of file list (length(filel
ist should = 5))
  while(i <= length(fileList)){

    #First item in list - where this is no other file to join with
    if(i == 1){
      currentData <- read.csv(fileList[i], header = TRUE, sep = '|', stringsAsFactors = FALSE)
      }

    #All subsequent files in list are read in then joined with the existing dataframe
    else{
      newData <- read.csv(fileList[i], header = TRUE, sep = '|', stringsAsFactors = FALSE)
      currentData <- full_join(currentData, newData, by = 'StationID')
    }

    #Reset count to move on to next item
    i <- i + 1
  }

  #Spit out the final dataframe with all data years joined
  return(currentData)
}

#Run function for heating and cooling data
heatData <- readTempFiles(heatFiles)
coolData <- readTempFiles(coolFiles)

#NOAA codes NAs as -9999 - setting to zero
heatData[heatData == -9999] <- 0
coolData[coolData == -9999] <- 0
```

```
#Initialize blank vector of same length as number of rows in heat data
heatTotal <- rep_len(NA, nrow(heatData))

#Iterate through each row in each data, take sum of columns 2 through the final column and divid
e by number of years to get annual avg
for(i in 1:nrow(heatData)){
  heatTotal[i] <- sum(heatData[i, 2:ncol(heatData)])/length(heatFiles)
}

#Appending total HDD as separate column
heatData$totalHDD <- heatTotal
```

```
#Follows same steps as heating data above
coolTotal <- rep_len(NA, nrow(coolData))

for(i in 1:nrow(coolData)){
  coolTotal[i] <- sum(coolData[i, 2:ncol(coolData)])/5
}

coolData$totalCDD <- coolTotal
```

```
#Isolating columns of interest - station ID and total values (don't need the thousands of daily
 values)
coolData_avg <- select(coolData, c(StationID, totalCDD))
heatData_avg <- select(heatData, c(StationID, totalHDD))

#Joining heating and cooling data by station ID
HDD_CDD <- full_join(coolData_avg, heatData_avg, by = 'StationID')
```

Note, there are several NOAA files in the folder. In sequential order of completeness, they are: allNOAAstn.txt, US_NOAAstn.txt, and USx_NOAAstn.txt.

The file read in here (USx_NOAAstn.csv) has been narrowed down to stations coded with a full ID starting with characters ‘US_’ with ’_’ being some other letter. These are the only stations that contain a column at the end with a numeric ID that matches those in the HDD/CDD station IDs. NOAA files were without metadata, so uncertain about the differences between nomenclature. Validating that these IDs are correct by plotting in ArcMap and verifying they are in the correct state and that the location name matches mapped location (i.e. is ‘Sacramento Airport’ in Sacramento?)

```
#Reading in NOAA station data
NOAAstn <- read.csv('NOAA_USXstn.csv', stringsAsFactors = F, header = T)

#Converting station names to string
NOAAstn$StationID <- as.character(NOAAstn$Column8)

#Joining station data and heating and cooling degree data, while selecting and renaming columns
 of interest
HDD_CDD_coords <- inner_join(NOAAstn, HDD_CDD, by = 'StationID') %>%
  select(-c(Column1, Column4, Column7, Column8)) %>%
  rename('latitude'='Column2', 'longitude'='Column3', 'state'='Column5', 'stnName'='Column6')

#Writing output for integration into ArcGIS for x/y plotting and interpolation
#write.csv(HDD_CDD_coords, 'HDD_CDD_coords.csv')
```