STATS 506 Report

Yu Lin, Yizhou Zhang, Yupeng Yang, Shukun Liu

12/7/2022

1. Description

This project is aiming at investigating whether urban heat island effects and heat waves are correlated to the the health (mortality rate and diseases) of the aged.

The repository contains:

- an analysis of how the mortality rate and occurrences of various diseases of the elderly is affected by the hot climate and heat waves;
- an interactive interface to calculate heat index and exhibit the data and model.

2. Background

2.1 heat wave definition

The heat waves in this project are identified by a temperature-humidity threshold: one day or longer periods when the daily maximum Wet Bulb Globe Temperature (WBGTmax) > 28 °C. This WBGTmax threshold follows the International Standards Organization (ISO) criteria for recognizing the risk of occupational heat related illnesses.

2.2 Analysis Scope and Measurement

- Death data was directly downloaded in csv from California State: Death Profiles by County by Month, 1970-2020 (https://data.chhs.ca.gov/dataset/death-profiles-by-county). This website provides monthly mortality data of all counties in California. We picked the monthly data from 2011 to 2015, based on which our analysis is county-wise.
- NASA MODIS Terra Land Surface 8-Day Average Temperature Emissivity Satellite data was acquired from NASA Earthdata website (https://search.earthdata.nasa.gov/search). We only picked the data for regions covering California from 07/20/2013 to 07/27/2013. This data will help us make a county-wise analysis in a small scope.
- Daily California station-wise temperature and humidity data was crawled from Western Regional Climate Center (WRCC) (https://wrcc.dri.edu/wraws/ccaF.html). Each county in California has multiple stations, but the WRCC website does not label which county each station belong to. The codes for crawlers are written in Python and we utilized the online computational platform Colab to run multiprocessing jobs.

- The geographical information (latitude and longitude) of each station is then crawled from the same WRCC website and utilized to map the stations into counties. The latitude and longitude information of the border of each county is retrieved from California Open Data Portal (https://data.ca.gov/dataset/california-counties). The following feature engineering is based on this processed county-wise weather data.

3. Data Exploration and Feature Engineering

3.1 Data Collection

Methodology:

We plan to analyze the relationship between heat waves and the (elderly) death rate in California because Californian data is sufficient and easy to access.

- From the WRCC website, We crawled daily weather data of over 500 stations from all over California. The data we collected features daily maximum air temperature and relative humidity data of all stations. The website sends the data for a specific station to the explorer only after the users fill out a request form, indicating the target year and month. We found that the data request form can be accessed through the URLs in the form "https://wrcc.dri.edu/cgi-bin/wea_monsum.pl?ca" + station code. Therefore we crawled all the station codes first, and then enumerate the code list to collect the weather data, which is stored as csv files.
- The NASA Satellite data for California was downloaded manually, as the website is low-efficient and crawler-unfriendly. The satellite collects data by regions in the shapes of curved parallelograms, and then projects the data to a square region through averaging. The square region is a 1200*1200 matrix, with each point corresponding to the emissivity of a 1km*1km grid after the compression (from parallelogram to square). We acquire the longitude and latitude information of the vertexes of the square regions covering California, and approximately find the real longitude and latitude information for each point. The average value and variance of the emissivity in a county will help to measure the heat-island effect of that county.

3.2 Data Processing

3.2.1 Calculate HImax and WBGTmax

Methodology To convert the huge daily station-wise weather data for all counties into some features or indicators that are more applicable to data analysis (regression), we decided to use the temperature and humidity data to calculate the WBGT index (a numerical measure of extremely hot weather). Apart from counting the heat waves defined above (WBGTmax $>$ 28 °C), we also want to take into account the intensity of heat waves. To compare days with heat waves to days without extreme temperature, we also reused the heat intensity formula for heat waves to quantify the intensity for days without extreme temperature. To calculate this heat intensity, we used daily minimum relative humidity and maximum temperature for each county we collected that introduced in Section 3.1.

$$HI_{max} = \frac{(0.5 \times (T_{max} + 61.0 + (T_{max} - 68.0) \times 1.2) + (0.094 RH_{min})) + T_{max}}{2}$$
$$WBGT_{max} = -0.0034 HI_{max}^2 + 0.96 HI_{max} - 34$$

First load the packages that used in this section

The next part is generalized to accept different files. User is able to input the data they interested in, and utilize the following code to investigate a specific location, finally generate a new case study. They are also able to design their output file name for their convenience.

```
## user-defined input
# list all the name of the station daily temperature and relative humidity files here.
# It should include "Station_name", "Year", "Month",
# "Date", "Air_Temperature_max", "Relative_Humidity_min" columns
input_file_names <- c("2011_0_100.csv","2011_101_200.csv","2011_201_300.csv",
                "2011_301_400.csv","2011_401_500.csv","2011_501_555.csv",
                "2012_0_185.csv","2012_185_370.csv","2012_370_555.csv",
                "2013_0_185.csv","2013_185_336.csv","2013_336_555.csv",
                "2014_0_190.csv","2014_190_380.csv","2014_380_555.csv",
                "2015_0_185.csv","2015_185_310.csv","2015_310_435.csv","2015_435_555.csv")

## user-defined output filename
output_stationwise_daily_HI_WGBT_filename <- "TEMP_HUM_HI_WBGT_2011_2015"
output_stationwise_monthly_HI_WGBT_filename <- "processed_HI_WBGT_2011_2015"
```

Then we processed the maximum Air Temperature and min Relative Humidity information to generate the daily max heat index (HImax) and the daily maximum Wet Bulb Globe Temperature (WBGTmax). It should be noticed that some weather data collected from the WRCC website is theoretically wrong. E.g., unexpected -99 Fahrenheit degrees. As our project focuses on the hot situation, and the input variables for the later regression model are monthly and county-wise, we chose to drop the station daily data in which temperature is lower than 30 Fahrenheit degrees. We also dropped the data with relative humidity out of the range from 0 to 100, since a relative humidity is rigorously defined in that range theoretically.

```
## Construct a new data frame to record all the temperature data
df <- data.frame(matrix(ncol=8,nrow=0))
colnames(df) <- c("Station_name", "Year", "Month", "Date",
                  "Air_Temperature_max", "Relative_Humidity_min", "HI_max", "WBGT_max")

## define a function to calculate HImax and WGBTmax
HI_WBGT_cal <- function(file_name){
  data <- read.csv(file_name, header=FALSE)
  colnames(data) <- c("Station_name", "Year", "Month", "Date",
                      "Air_Temperature_max", "Relative_Humidity_min")

  # drop abnormal data
  data <- data[!is.na(data$Air_Temperature_max),]
  data <- data[!is.na(data$Relative_Humidity_min),]
  data <- data[data$Air_Temperature_max>=30,]
  data <- data[(data$Relative_Humidity_min<=100 & data$Relative_Humidity_min>=0),]

  T_max <- as.numeric(data$Air_Temperature_max)
  RH_min <- as.numeric(data$Relative_Humidity_min)

  HI_max <- ((0.5*(T_max+61.0+(T_max-68.0)*1.2)+(0.094*RH_min))+T_max)/2
  WBGT_max <- -0.0034*HI_max^2 + 0.96*HI_max-34

  data <- data %>%
    mutate(HI_max = HI_max,
           WBGT_max = WBGT_max)
```

```
  return(data)
}

# merge all the csv into one data frame and remove those with NA WBGT_max value
for (file_name in input_file_names) {
  data <- HI_WBGT_cal(file_name)
  df <- rbind(df, data)
}
```

After the data being processed, we write the merged data into RMDBS and store it as a csv

```
write.csv(df,paste(output_stationwise_daily_HI_WBGT_filename,".csv",sep=""), row.names = FALSE)
# mysqlconnection = dbConnect(RMySQL::MySQL(),
#                             dbname='506_project',
#                             host='35.2.205.222',
#                             port=3306,
#                             user='linyu',
#                             password='123456',
# )
# dbWriteTable(mysqlconnection, name=output_stationwise_daily_HI_WBGT_filename,
# value=df, row.names = FALSE, overwrite=TRUE)
```

3.2.2 Feature Engineering and Aggregate it into Monthly Data

As we mentioned before, our death profile data is recorded monthly. To parallel all the data into the same level, here we engineer the HImax and WBGTmax data into month level and conduct some feature engineering to explore further.

We want to construct 4 features:

1) Monthly average WGBT_max in each station
2) Duration of heat wave in each station by month (WGBT_max >28)
3) Monthly maximum WGBT_max in each station
4) Monthly average WGBT_max in each station for heat wave (WGBT_max > 28)

```
data_month <- df %>%
  group_by(Year, Month, Station_name) %>%
  summarise(
    avg_WBGT_max_monthly = mean(WBGT_max,na.rm=TRUE),
    max_WBGT_max_monthly = max(WBGT_max,na.rm=TRUE)
  )
```

```
## `summarise()` has grouped output by 'Year', 'Month'. You can override using the
## `.groups` argument.
```

```
data_month_heat_wave <- df %>%
  group_by(Year, Month, Station_name) %>%
  filter(WBGT_max>28) %>%
  summarise(
    duration_heat_wave_monthly = ifelse(is.na(n())==TRUE,0,n()),
    avg_heat_waves_WBGT_max_monthly = mean(WBGT_max,na.rm=TRUE)
  )
```

```
## `summarise()` has grouped output by 'Year', 'Month'. You can override using the
## `.groups` argument.
```

```
data_month <- left_join(data_month, data_month_heat_wave, by=c("Year","Month","Station_name"))
```

Write the processed data into RMDBS and store it as a csv

```
write.csv(data_month,paste(output_stationwise_monthly_HI_WGBT_filename,".csv",sep=""), row.names = FALSE
#dbWriteTable(mysqlconnection, name="processed_HI_WBGT_2011_2015",
# value=data_month, row.names = FALSE, overwrite=TRUE)
gc()
```

```
##            used (Mb) gc trigger  (Mb) max used  (Mb)
## Ncells  1472368 78.7    2567635 137.2  2567635 137.2
## Vcells  8607925 65.7   18060321 137.8 18060313 137.8
```

3.2.3 Map Station Data to Counties

First load the packages

Just like the previous section, users are able to modify the input files to generate a new case study. Note that the input files have to contain some columns with specific format.

```
## 1. a monitoring station geometric information csv: should include
###  1) a column "name" for the name of that monitoring station
###  2) a column "Longitude" for the longitude of that monitoring station (form: xx°xx'xx")
###  2) a column "Latitude" for the latitude of that monitoring station (form: xx°xx'xx")
monitoring_station_csv_filename <- "CA_monitoring_station.csv"
## 2. a shp document and its auxiliary document that contains county geometry boundary information
aoi_boundary_shp_filename <- "CA_Counties_TIGER2016.shp"
## 3. a processed station-wise daily Heat Index (HI) & Wet Bulb Globe Temperature (WBGT) document
HI_WBGT_stationwise_csv_filename <- "processed_HI_WBGT_2011_2015.csv"

## self-defined output filenames and database info
countywise_HI_WBGT_csv_filename <- "countywise_HI_WBGT_2011_2015.csv"

monitoring_station <- read.csv(monitoring_station_csv_filename)
monitoring_station <-  # overkill the repetitions
  monitoring_station %>%
  group_by(name) %>%
  summarize(code = min(code, na.rm = TRUE),
            Longitude = min(Longitude, na.rm = TRUE),
            Latitude = min(Latitude, na.rm = TRUE))
```

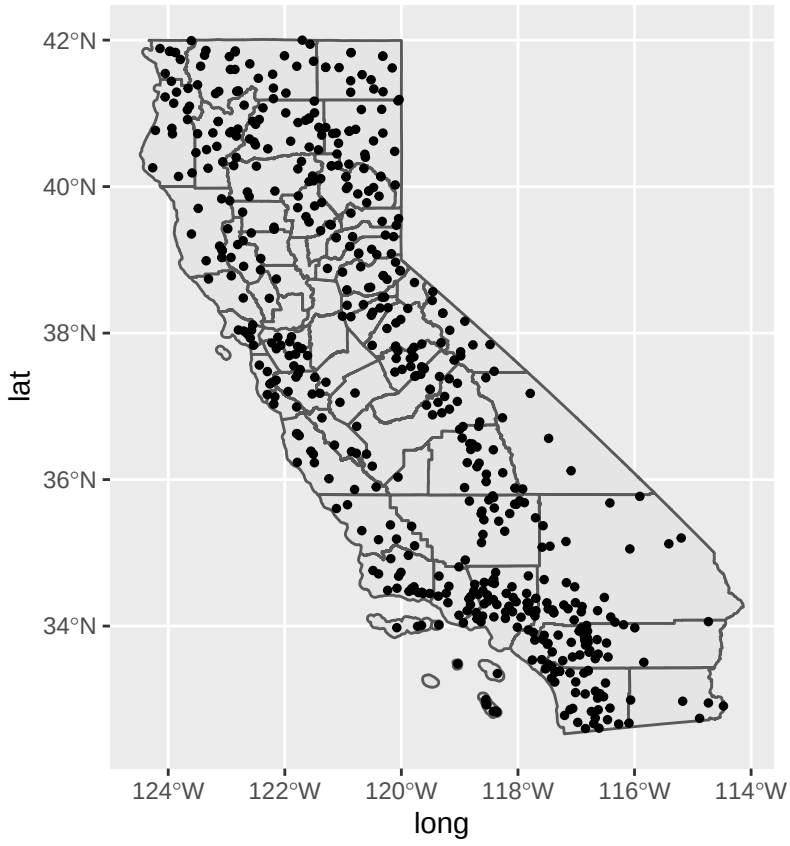
Since the geometric information in our file is str format and hard to compare, we then define a function to transform the longitude and latitude form from xx°xx'xx" to decimal degree.

```
str_transform <- function(str){
  str <- str_replace_all(str, "°", "")
  str <- str_replace_all(str, "\'", "")
  str <- str_replace_all(str, "\"", "")
  dec_str <- as.numeric(conv_unit(str, from = "deg_min_sec", to = "dec_deg"))
```

```
  return(dec_str)
}

lat <- str_transform(monitoring_station$Latitude)
long <- -str_transform(monitoring_station$Longitude)
long_lat <- data.frame("name"=monitoring_station$name,
                       "code"=monitoring_station$code,
                       "long"=long,
                       "lat"=lat)
```

To compare and map the monitoring station spot into county polygon, we transformed coordinate to keep they are in the same coordinates. We then generate a overview map for reference.

```
## transform coordinate !!!
aoi_boundary_HARV <- read_sf(aoi_boundary_shp_filename)
aoi_boundary_HARV$geometry <- st_transform(aoi_boundary_HARV$geometry, 4326)

## Plot a overview map
# png("map_station_county.png", width = 960, height = 960)
ggplot()+
  geom_sf(data=aoi_boundary_HARV$geometry)+
  geom_point(aes(long, lat),size=1)
```



```
# dev.off()
```

With all data in the same coordinate, we retrieve the required columns, and mapped monitoring station into counties by "st_within". Note that users could put other useful columns in the long_lat data frame as well.

```
sta_sf <- long_lat %>%
  mutate_at(vars(long, lat), as.numeric) %>%   # coordinates must be numeric
  st_as_sf(
    coords = c("long", "lat"),
    agr = "constant",
    crs = 4326,
    stringsAsFactors = FALSE,
    remove = TRUE
  )
station_in_county <- st_join(sta_sf, aoi_boundary_HARV, join = st_within)
head(station_in_county)
```

```
## Simple feature collection with 6 features and 19 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: -123.5903 ymin: 33.53611 xmax: -117.7533 ymax: 41.62778
## Geodetic CRS:  WGS 84
##                         name code STATEFP COUNTYFP COUNTYNS GEOID        NAME
## 1           Acton California CACT      06      037 00277283 06037 Los Angeles
## 2 Adin - Portable California CADI      06      049 00277289 06049       Modoc
## 3         Alcalde California CALC      06      019 00277274 06019      Fresno
## 4     Alder Point California CAPT      06      023 01681908 06023    Humboldt
## 5   Alder Springs California CALD      06      021 00277275 06021       Glenn
## 6    Aliso Laguna California CALG      06      059 00277294 06059      Orange
##             NAMELSAD LSAD CLASSFP MTFCC CSAFP CBSAFP METDIVFP FUNCSTAT
## 1 Los Angeles County   06      H1 G4020   348  31080    31084        A
## 2       Modoc County   06      H1 G4020  <NA>   <NA>     <NA>        A
## 3      Fresno County   06      H1 G4020   260  23420     <NA>        A
## 4    Humboldt County   06      H1 G4020  <NA>  21700     <NA>        A
## 5       Glenn County   06      H1 G4020  <NA>   <NA>     <NA>        A
## 6      Orange County   06      H1 G4020   348  31080    11244        A
##         ALAND     AWATER    INTPTLAT     INTPTLON                    geometry
## 1 10510651024 1794730436 +34.1963983 -118.2618616    POINT (-118.2 34.44583)
## 2 10140955630  745425201 +41.5929185 -120.7183704 POINT (-121.2983 41.62778)
## 3 15433177265  135374444 +36.7610058 -119.6550193 POINT (-120.4986 36.18333)
## 4  9240992572 1254297982 +40.7066731 -123.9258181 POINT (-123.5903 40.18667)
## 5  3403104376   33749275 +39.6025462 -122.4016998 POINT (-122.7236 39.65139)
## 6  2047702298  407606601 +33.6756872 -117.7772068 POINT (-117.7533 33.53611)
```

We then join the station in county table and processed WBGT stationwise table.

```
HI_WBGT_stationwise <- read.csv(HI_WBGT_stationwise_csv_filename)
county_in_station_HI_WBGT <-
  left_join(HI_WBGT_stationwise, station_in_county, by=c("Station_name"="name"))
```

To match the death profile data level, we still need to aggregate stationwise data into countywise data.

Finally, we could write the processed data into RMDBS and store it as a csv

```
write.csv(countywise_HI_WBGT, file = countywise_HI_WBGT_csv_filename, row.names = FALSE)
# dbWriteTable(mysqlconnection, name="countywise_HI_WBGT",
# value=data_month, row.names = FALSE, overwrite=TRUE)
```

3.2.4 Death Profile Data Manipulation

Regarding the death data, we are able to access the death data of each county per month on the website. We will use the total population, total death number and the elderly's death number of each county as our response variables. The data also includes the cause of deaths. We then will fit a regression model and find out the relationship between heat waves and the death rate (or relative illnesses) of California.

Since we intended to analyze data from 2011 to 2015. We downloaded two csv files from the California Health and Human Service website https://data.chhs.ca.gov/ . To lessen the redundancy of code. We wrote a function for extracting useful death number from the csv file. It needs to be pointed out that the function is only compatible with the data set from this website considering the unique and complex format of its data file.

```
source("gdn.R")

data1 = read.csv("2021-05-14_deaths_final_2009-2013_occurrence_county_month_sup.csv")
data2 = read.csv("2021-11-29_deaths_final_2014-2018_occurrence_county_month_sup.csv")

#Extracting useful data from the raw dataset.
d2011<-gdn(data1,2011)
d2012<-gdn(data1,2012)
d2013<-gdn(data1,2013)
d2014<-gdn(data2,2014)
d2015<-gdn(data2,2015)

#Bind the five years' data frames by row.
dd<-rbind(d2011,d2012,d2013,d2014,d2015)
write.csv(dd,"death.csv",row.names = F)
```

The function is attached below. We aims to get the death number of people with age equal to or larger than 65.

```
#The function is for extracting death number for the elderly.
gdn <- function(dataset, year) {
  dataset[is.na(dataset)]=0
  county_name<-unique(dataset$County)
  month<-unique(dataset$Month)
  d<-c()
  for (j in month) {
    for (i in county_name) {
      elderly_death<-sum(dataset$Count[dataset$County==i&dataset$Year==year&
                          dataset$Month==j&
                          dataset$Cause_Desc=="All causes (total)"&
                          dataset$Strata=="Age"][9:11])
      d<-c(d,elderly_death)
    }
  }
  #Creating a data frame that stores the death number with the Year, Month, and County.
  df=data.frame(county=rep(county_name,12),
```

```
                year=rep(year,12*length(county_name)),
                month=rep(1:12,each=length(county_name)),
                death=d)

  return(df)
}
```

3.2.5 Census data

We got the Californian census data from the package "tidycensus" in county level. We obtained the total population of each county in California and the population of elderly (Age ≥ 65).

In the block above, f1 is for taking the total population, f2 is for taking the elderly population. To make the data frame more readable and easier to conduct data analysis later. We add the Year and Month as two new columns.

3.2.6 Calculate heat island index within counties

Load the required packages

```
library(measurements)
library(stringr)
library(dplyr)
library(sf)
library(tidyverse)
library(ggplot2)
```

Users are able to modify the input files to generate a new case study. Here we use the NASA emissivity satellite data within California from 7/20/2013 to 7/27/2013. The raw data was processed and stored as 1200*1200 matrices in txt files.

```
## self-defined input files: You can modify the input files to generate a new case study
## 1. a set of files including heatmap, latitude, longitude information respectively.
heat_txt_file <- "heat_map.txt"
lat_txt_file  <- "lat_mat.txt"
long_txt_file  <- "long_mat.txt"

sup_heat_txt_file <- "sup_heat_map.txt"
sup_lat_txt_file  <- "sup_lat_mat.txt"
sup_long_txt_file  <- "sup_long_mat.txt"

## 2. a shp document and its auxiliary document that contains county geometry boundary information
aoi_boundary_shp_filename <- "CA_Counties_TIGER2016.shp"

## self-defined output filenames and database info
heat_point_within_county_csv_filename <- "heat_point_within_county.csv"
countywise_Heatmap_csv_filename <- "countywise_Heatmap.csv"
```

Then we read the heat information and geometric information, and transform them into long vectors, in preparation for mapping each data point onto its corresponding county.

```
## read the heat information and geometric information
heatmap <- as.matrix(read.table(heat_txt_file,sep=","))
lat <- as.matrix(read.table(lat_txt_file,sep=","))
long <- as.matrix(read.table(long_txt_file,sep=","))

sup_heatmap <- as.matrix(read.table(sup_heat_txt_file,sep=","))
sup_lat <- as.matrix(read.table(sup_lat_txt_file,sep=","))
sup_long <- as.matrix(read.table(sup_long_txt_file,sep=","))

## Convert Matrix to Vector, record each entry from the original matrix plot as an observation
heatmap_vec <- c(as.vector(heatmap),as.vector(sup_heatmap))
lat_vec <- c(as.vector(lat),as.vector(sup_lat))
long_vec <- c(as.vector(long),as.vector(sup_long))
```

After the transformation, we combine three columns together and removed abnormal row. E.g., the emissity should be strictly larger than 0.

```
## combine three columns together
heatmap <- as.data.frame(cbind(heatmap_vec,lat_vec,long_vec))
colnames(heatmap) <- c("heat","lat","long")

## remove the abnormal observation (with heat<=0)
heatmap <- heatmap[heatmap$heat > 0, ]
lat <- heatmap$lat
long <- heatmap$long
```

We then got a new matrix with the following three columns: 1. heat emissivity;
2. latitude;
3. longitide

Using the similar method we conducted in Section 3.2.3, We mapped each observations into California counties and only kept those within California. We also wrote the output dataframe into a csv file.

```
## transform coordinate !!!
aoi_boundary_HARV <- read_sf(aoi_boundary_shp_filename)
aoi_boundary_HARV$geometry <- st_transform(aoi_boundary_HARV$geometry, 4326)

## You can put other useful columns in the long_lat data frame as well.
heat_sf <- heatmap %>%
  mutate_at(vars(long, lat), as.numeric) %>%   # coordinates must be numeric
  st_as_sf(
    coords = c("long", "lat"),
    agr = "constant",
    crs = 4326,
    stringsAsFactors = FALSE,
    remove = TRUE
  )

point_in_county <- st_join(heat_sf, aoi_boundary_HARV, join = st_within)
point_within_county <- point_in_county[!is.na(point_in_county$NAME),]
point_within_county <- data.frame("heat"=point_within_county$heat,
                                  "county"=point_within_county$NAME,
                                  "geometry"=point_within_county$geometry)
```

```
## write out the processed heatmap within counties
write.csv(point_within_county, file = heat_point_within_county_csv_filename, row.names = FALSE)
```

The final step in handling these heat island data is to aggregate point-wise data into county-wise data and conduct feature engineering.

4. Model Fitting

4.1 Regression Model for heat wave data and death-related data

4.1.1. Preparation

Step 1: Load the data and the packages

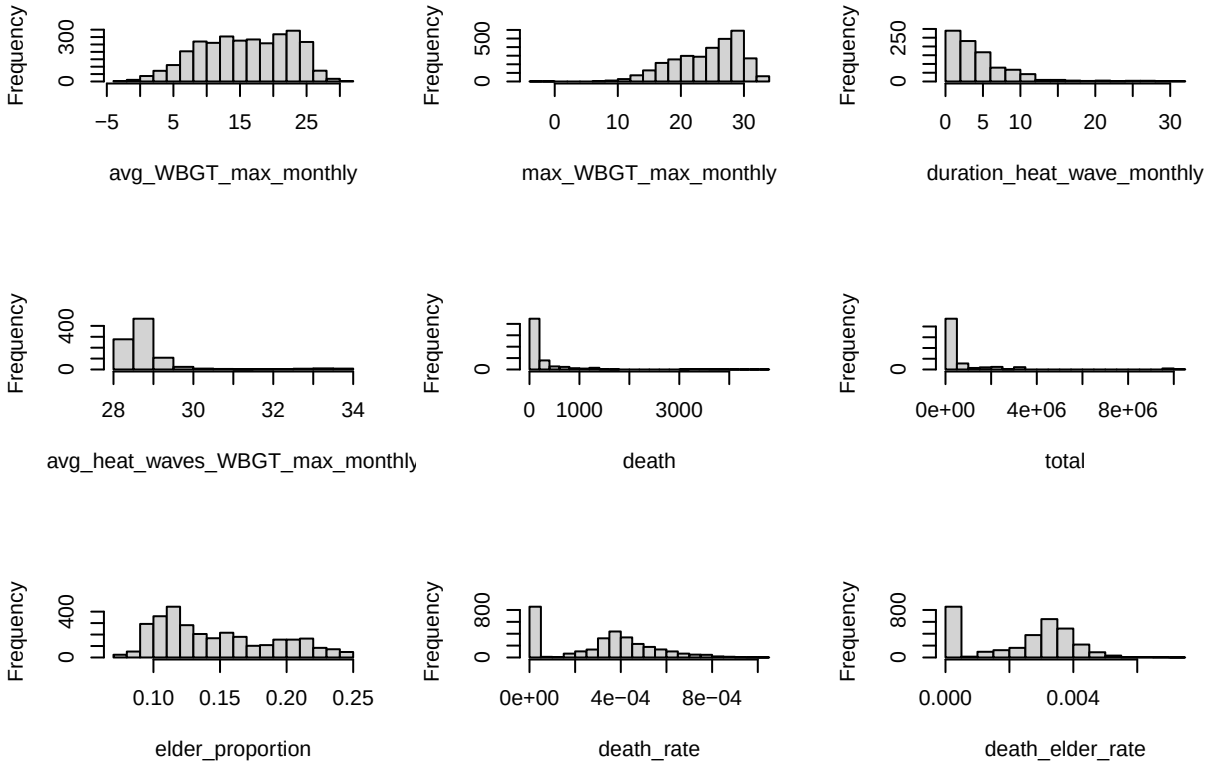
```
library(dplyr)
library(tidyr)

death <- read.csv("death.csv")
elderly <- read.csv("cal_census_2011_to_2015.csv")
HI_WBGT <- read.csv("countywise_HI_WBGT_2011_2015.csv")
```

Step 2: Join the data into one dataframe

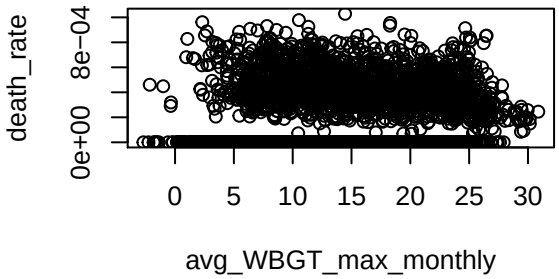
```
# data manipulation
HI_WBGT$Year <- as.integer(HI_WBGT$Year)
HI_WBGT$Month <- as.integer(HI_WBGT$Month)

# join the HI_WBGT table and the death profile by year, month and county name
heat_death <- HI_WBGT %>% inner_join( death,
                                      by=c("Year"="year","Month"="month","NAME"="county"))


# join the processed table and the elderly information table by year and county name
heat_death <- heat_death %>% left_join( elderly,
                                        by=c("Year"="Year","NAME"="County"))

# add columns to record the death rate
heat_death <- heat_death %>%
  mutate( death_rate = death / total,
    death_elder_rate = death / elderly )

# remove the outliners
heat_death <- heat_death[heat_death$death_elder_rate <= 0.5,]

# Write the processed data into RMDBS and store it as a csv
write.csv(heat_death, file = "heat_death_2011_2015.csv", row.names = FALSE)
```

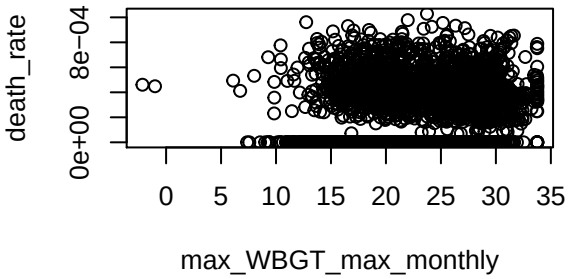
4.1.2 Exploratory Data Analysis and Data Standardization

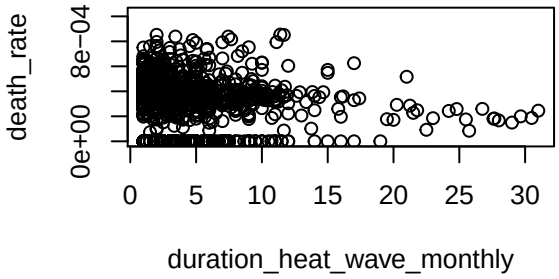
```r
# Plot for all variables
par(mfrow=c(3,3))
x_lab <- colnames(heat_death)
for (i in c(4:9,11:ncol(heat_death))){
  hist(heat_death[,i],
       xlab = x_lab[i],main="",nclass=20)
}
```

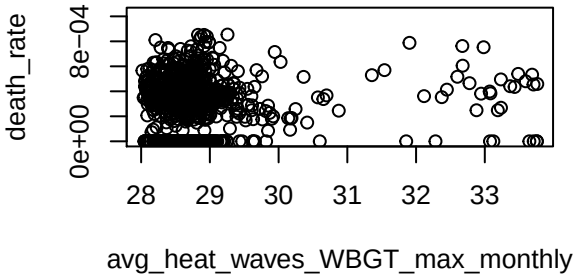



```r
# relationship between features and death rate
par(mfrow=c(2,2))
for (i in 4:7){
  plot(heat_death[,i],heat_death$death_rate,
       xlab=x_lab[i],ylab="death_rate")
}
```

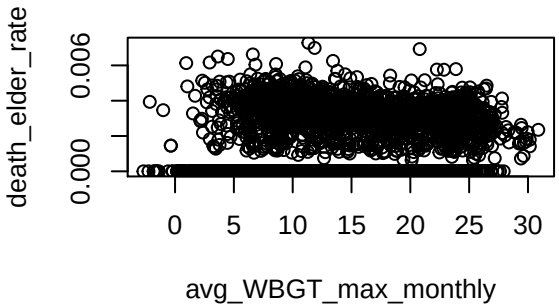


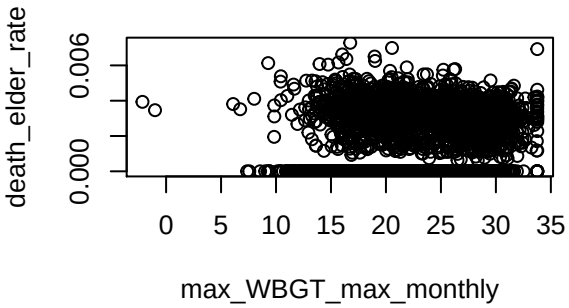


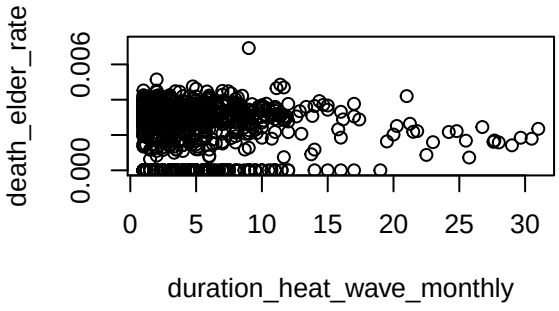


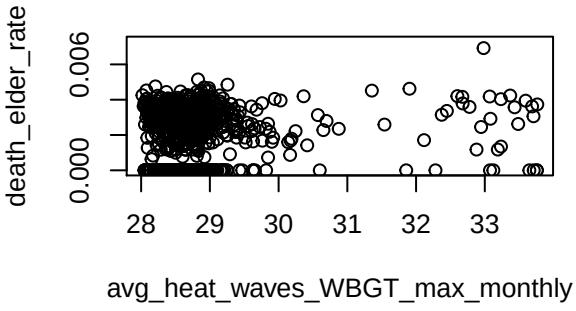



```r
# relationship between features and death rate in the old generation
par(mfrow=c(2,2))
for (i in 4:7){
  plot(heat_death[,i],heat_death$death_elder_rate,
       xlab=x_lab[i],ylab="death_elder_rate")
}
```










```
standardize <- function(col){
  sample_mean <- mean(col,na.rm=TRUE)
  sd <- sqrt(var(col,na.rm = TRUE))
  return((col-sample_mean)/sd)
}

standard_heat_death <- heat_death
for (i in 4:ncol(heat_death)){
  standard_heat_death[,i] <- standardize(heat_death[,i])
}
```

4.1.3 Construct relationship between different variables

```
# relationship between the death rate and the intensity of heat wave
# (within county that have at least 1 heat waves in one month)
m1 <- lm(death_rate ~
                avg_WBGT_max_monthly +
                max_WBGT_max_monthly +
                duration_heat_wave_monthly +
                avg_heat_waves_WBGT_max_monthly, data = standard_heat_death)
summary(m1)
```

```
##
```

```
## Call:
## lm(formula = death_rate ~ avg_WBGT_max_monthly + max_WBGT_max_monthly +
##     duration_heat_wave_monthly + avg_heat_waves_WBGT_max_monthly,
##     data = standard_heat_death)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2590 -0.4094  0.1156  0.4668  2.5315
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -0.25992    0.23383  -1.112  0.26661
## avg_WBGT_max_monthly            -0.33479    0.06857  -4.882 1.24e-06 ***
## max_WBGT_max_monthly             0.52921    0.22180   2.386  0.01724 *
## duration_heat_wave_monthly      -0.04125    0.04040  -1.021  0.30745
## avg_heat_waves_WBGT_max_monthly -0.14733    0.05502  -2.678  0.00755 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8707 on 916 degrees of freedom
##   (2193 observations deleted due to missingness)
## Multiple R-squared:  0.04148,    Adjusted R-squared:  0.03729
## F-statistic:  9.91 on 4 and 916 DF,  p-value: 7.489e-08
```

conclusion: within counties that have heat waves, counties with larger average WBGTmax and larger average intensity of heat waves tend to have less death rate, counties with larger maximum intensity of heat waves tend to have smaller death rate.

```
# relationship between the death elder rate and the intensity of heat wave
# (within county that have at least 1 heat waves in one month)
m2 <- lm(death_elder_rate ~
                avg_WBGT_max_monthly +
                max_WBGT_max_monthly +
                duration_heat_wave_monthly +
                avg_heat_waves_WBGT_max_monthly
         , data = standard_heat_death)
summary(m2)
```

```
##
## Call:
## lm(formula = death_elder_rate ~ avg_WBGT_max_monthly + max_WBGT_max_monthly +
##     duration_heat_wave_monthly + avg_heat_waves_WBGT_max_monthly,
##     data = standard_heat_death)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3368 -0.6054  0.3134  0.6048  3.1269
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -0.42532    0.23219  -1.832 0.067309 .
## avg_WBGT_max_monthly            -0.31541    0.06809  -4.632 4.15e-06 ***
## max_WBGT_max_monthly             0.68974    0.22024   3.132 0.001793 **
```

```
## duration_heat_wave_monthly      -0.01023    0.04011  -0.255 0.798694
## avg_heat_waves_WBGT_max_monthly -0.18970    0.05464  -3.472 0.000541 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8646 on 916 degrees of freedom
##   (2193 observations deleted due to missingness)
## Multiple R-squared:  0.03236,    Adjusted R-squared:  0.02814
## F-statistic: 7.659 on 4 and 916 DF,  p-value: 4.528e-06
```

conclusion: within counties that have heat waves, counties with larger maximum WBGTmax tend to have less death rate, counties with larger average intensity of heat waves tend to have larger death rate in old generation.

```
# relationship between the death rate and the intensity of heat wave
# (including all the counties)
m3 <- lm(death_rate ~
                avg_WBGT_max_monthly +
                max_WBGT_max_monthly, data = standard_heat_death)
summary(m3)
```

```
##
## Call:
## lm(formula = death_rate ~ avg_WBGT_max_monthly + max_WBGT_max_monthly,
##     data = standard_heat_death)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6472 -1.3206  0.1905  0.6393  3.1017
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.109e-16  1.791e-02   0.000   1.0000
## avg_WBGT_max_monthly -8.508e-02  3.371e-02  -2.524   0.0117 *
## max_WBGT_max_monthly  7.781e-02  3.371e-02   2.308   0.0210 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9993 on 3111 degrees of freedom
## Multiple R-squared:  0.002076,   Adjusted R-squared:  0.001435
## F-statistic: 3.236 on 2 and 3111 DF,  p-value: 0.03945
```

conclusion: for all counties, counties with larger average WBGTmax tend to have less death rate, counties with larger maximum WBGTmax tend to have less death rate.

```
# relationship between the death elder rate and the intensity of heat wave
# (including all the counties)
m4 <- lm(death_elder_rate ~
                avg_WBGT_max_monthly +
                max_WBGT_max_monthly, data = standard_heat_death)
summary(m4)
```

```
##
```

```
## Call:
## lm(formula = death_elder_rate ~ avg_WBGT_max_monthly + max_WBGT_max_monthly,
##     data = standard_heat_death)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5670 -1.3767  0.3685  0.7418  3.1064
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            6.463e-16  1.791e-02   0.000    1.000
## avg_WBGT_max_monthly -2.152e-02  3.372e-02  -0.638    0.523
## max_WBGT_max_monthly  5.186e-02  3.372e-02   1.538    0.124
##
## Residual standard error: 0.9997 on 3111 degrees of freedom
## Multiple R-squared:  0.001261,   Adjusted R-squared:  0.0006192
## F-statistic: 1.964 on 2 and 3111 DF,  p-value: 0.1404
```

conclusion: for all counties, no significant relationship between death elder rate and the heat waves.

```
# relationship between the elder proportion and the intensity of heat wave
# (within county that have at least 1 heat waves in one month)
m5 <- lm(elder_proportion ~
                avg_WBGT_max_monthly +
                max_WBGT_max_monthly +
                duration_heat_wave_monthly +
                avg_heat_waves_WBGT_max_monthly, data = standard_heat_death)
summary(m5)
```

```
##
## Call:
## lm(formula = elder_proportion ~ avg_WBGT_max_monthly + max_WBGT_max_monthly +
##     duration_heat_wave_monthly + avg_heat_waves_WBGT_max_monthly,
##     data = standard_heat_death)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5956 -0.7702 -0.2868  0.6465  2.3883
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -0.19683    0.26160  -0.752 0.452009
## avg_WBGT_max_monthly            -0.05148    0.07672  -0.671 0.502367
## max_WBGT_max_monthly             0.15424    0.24814   0.622 0.534374
## duration_heat_wave_monthly      -0.15973    0.04519  -3.534 0.000429 ***
## avg_heat_waves_WBGT_max_monthly  0.02510    0.06156   0.408 0.683613
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9741 on 916 degrees of freedom
##   (2193 observations deleted due to missingness)
## Multiple R-squared:  0.02661,    Adjusted R-squared:  0.02236
## F-statistic: 6.259 on 4 and 916 DF,  p-value: 5.706e-05
```

conclusion: within counties that have heat waves, counties with longer heat waves tend to have lower elder proportion.

```
# relationship between the total population and the intensity of heat wave
# (within county that have at least 1 heat waves in one month)
m6 <- lm(total ~
                 avg_WBGT_max_monthly +
                 max_WBGT_max_monthly +
                 duration_heat_wave_monthly +
                 avg_heat_waves_WBGT_max_monthly,
         data = standard_heat_death)
summary(m6)
```

```
##
## Call:
## lm(formula = total ~ avg_WBGT_max_monthly + max_WBGT_max_monthly +
##     duration_heat_wave_monthly + avg_heat_waves_WBGT_max_monthly,
##     data = standard_heat_death)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6002 -0.5312 -0.2572  0.0742  6.1641
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      -2.80291    0.33023  -8.488  < 2e-16 ***
## avg_WBGT_max_monthly             -0.36634    0.09684  -3.783 0.000165 ***
## max_WBGT_max_monthly              3.13098    0.31323   9.996  < 2e-16 ***
## duration_heat_wave_monthly       -0.20688    0.05705  -3.626 0.000304 ***
## avg_heat_waves_WBGT_max_monthly -0.65231    0.07771  -8.394  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.23 on 916 degrees of freedom
##   (2193 observations deleted due to missingness)
## Multiple R-squared:  0.1042, Adjusted R-squared:  0.1003
## F-statistic: 26.65 on 4 and 916 DF,  p-value: < 2.2e-16
```

conclusion: within counties that have heat waves, counties with larger average WBGTmax, larger average intensity of heat waves and longer duration of heat waves tends to have less death rate, counties with larger maximum intensity of heat waves tend to have less total population.

4.2 Regression Model for heat wave data and death data of each cause of death

4.2.1 Load data and extract useful data.

```
# This function extracts the death number of every cause of death by month by county.
disease <- function(dataset, year) {
  dataset[is.na(dataset)]=0
  set<-c()
  county_name<-unique(dataset$County)
```

```
  month<-unique(dataset$Month)
  d = data.frame(matrix(nrow = 0,ncol = 15))
  colnames(d)<-unique(dataset$Cause_Desc)
  for (j in month) {
    for (i in county_name) {
      elderly_death<-dataset$Count[dataset$County==i&dataset$Year==year&dataset$Month==j
                                   &dataset$Strata=="Total Population"]

      d = rbind(d,elderly_death)
    }
  }
  df=data.frame(county=rep(county_name,12),
                year=rep(year,12*length(county_name)),
                month=rep(1:12,each=length(county_name)))
  df=cbind(df,d)
  colnames(df)<-c("county","year","month",unique(data1$Cause_Desc))
  return(df)
}
```

```
# Death number of year from 2011 to 2015
d2011<-disease(data1,2011)
d2012<-disease(data1,2012)
d2013<-disease(data1,2013)
d2014<-disease(data2,2014)
d2015<-disease(data2,2015)
disease<-rbind(d2011,d2012,d2013,d2014,d2015)
```

4.2.2 Data Analysis and Relationship between heat wave data and the death number of each cause of death

```
# We merged the heat index table and the disease statitics table
heat_index = read.csv('heat_death_2011_2015.csv')
colnames(heat_index)[1:3]=c('year','month','county')
merged = merge(heat_index, disease, by = c('year','month','county'))
```

```
# To reduce the variance and increase the accuracy, we deleted the county with a very small population
table_big = merged[merged$total>100000,]
table_big[is.na(table_big)]=0
m1 = lm(death/`All causes (total)`~avg_WBGT_max_monthly+max_WBGT_max_monthly+duration_heat_wave_monthly+
summary(m1)
```

```
##
## Call:
## lm(formula = death/`All causes (total)` ~ avg_WBGT_max_monthly +
##     max_WBGT_max_monthly + duration_heat_wave_monthly + avg_heat_waves_WBGT_max_monthly,
##     data = table_big)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.71439 -0.03443  0.01431  0.05284  0.18220
##
```

```
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     0.7370008  0.0142048  51.884  < 2e-16 ***
## avg_WBGT_max_monthly           -0.0038289  0.0006196  -6.180 7.91e-10 ***
## max_WBGT_max_monthly            0.0022177  0.0008070   2.748  0.00606 **
## duration_heat_wave_monthly     -0.0045949  0.0007334  -6.265 4.66e-10 ***
## avg_heat_waves_WBGT_max_monthly  0.0002765  0.0002435   1.136  0.25626
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08759 on 1797 degrees of freedom
## Multiple R-squared:  0.08064,    Adjusted R-squared:  0.07859
## F-statistic:  39.4 on 4 and 1797 DF,  p-value: < 2.2e-16
```

This shows that the proportion of the elderly in death cases increases as the max_WBGT_max_monthly increases. However, the acg_WBGT_max_monthly, duration_heat_wave_monthly, and avg_heat_waves_WBGT_max_mo will reduce the elderly proportion of death cases. The reason of this may be that the extreme hot weather will take a more severe damage to the health of the elderly.

```
#The respective analysis of causes of death
table_b = table_big[,c(4:7,14:28)]
df1 = data.frame(matrix(nrow=0,ncol=2))
df2 = data.frame(matrix(nrow=0,ncol=2))
for (i in colnames(table_b)[5:19]) {
  m = lm(table_b[,i]/table_big$elderly~avg_WBGT_max_monthly+max_WBGT_max_monthly+
           duration_heat_wave_monthly+avg_heat_waves_WBGT_max_monthly,data = table_b)
  a = summary(m)$coefficients[3,c(1,4)]
  df1 = rbind(df1,a)
  b = summary(m)$coefficients[2,c(1,4)]
  df2 = rbind(df2,b)
}
colnames(df1)<-c('max_WBGT_max_monthly', 'p-value')
rownames(df1)<-colnames(table_b)[5:19]
print(df1)
```

```
##                                                    max_WBGT_max_monthly
## All causes (total)                                         1.789392e-05
## Alzheimer's disease                                        6.211256e-06
## Malignant neoplasms                                        7.422491e-06
## Chronic lower respiratory diseases                         1.045145e-05
## Diabetes mellitus                                          5.131686e-06
## Assault (homicide)                                         5.118413e-07
## Diseases of heart                                          1.210335e-05
## Essential hypertension and hypertensive renal disease      2.579787e-06
## Accidents (unintentional injuries)                         5.997267e-06
## Chronic liver disease and cirrhosis                        3.136949e-06
## Nephritis, nephrotic syndrome and nephrosis                1.606666e-06
## Parkinson's disease                                        1.009045e-06
## Influenza and pneumonia                                    2.165389e-06
## Cerebrovascular diseases                                   9.011280e-06
## Intentional self-harm (suicide)                            2.197109e-06
##                                                               p-value
## All causes (total)                                      1.793593e-02
```

```
## Alzheimer's disease                                          7.511733e-06
## Malignant neoplasms                                          1.108859e-03
## Chronic lower respiratory diseases                           2.460889e-11
## Diabetes mellitus                                            3.661778e-09
## Assault (homicide)                                           9.268786e-03
## Diseases of heart                                            9.580042e-06
## Essential hypertension and hypertensive renal disease        5.616299e-07
## Accidents (unintentional injuries)                           5.621719e-05
## Chronic liver disease and cirrhosis                          6.953451e-13
## Nephritis, nephrotic syndrome and nephrosis                  5.002450e-11
## Parkinson's disease                                          1.480713e-06
## Influenza and pneumonia                                      1.441918e-03
## Cerebrovascular diseases                                     3.700074e-10
## Intentional self-harm (suicide)                              1.894977e-09
```

```
colnames(df2)<-c('avg_WBGT_max_monthly', 'p-value')
rownames(df2)<-colnames(table_b)[5:19]
print(df2)
```

```
##                                                       avg_WBGT_max_monthly
## All causes (total)                                              -7.417471e-05
## Alzheimer's disease                                             -6.889060e-06
## Malignant neoplasms                                             -1.223766e-05
## Chronic lower respiratory diseases                              -1.601506e-05
## Diabetes mellitus                                               -5.147441e-06
## Assault (homicide)                                              -3.197160e-07
## Diseases of heart                                               -2.650747e-05
## Essential hypertension and hypertensive renal disease           -2.315809e-06
## Accidents (unintentional injuries)                              -5.756487e-06
## Chronic liver disease and cirrhosis                             -1.952303e-06
## Nephritis, nephrotic syndrome and nephrosis                     -7.505390e-07
## Parkinson's disease                                             -3.918894e-07
## Influenza and pneumonia                                         -4.161050e-06
## Cerebrovascular diseases                                        -9.393672e-06
## Intentional self-harm (suicide)                                 -8.531427e-07
##                                                                     p-value
## All causes (total)                                           6.356599e-36
## Alzheimer's disease                                          1.110894e-10
## Malignant neoplasms                                          3.233705e-12
## Chronic lower respiratory diseases                           3.788379e-39
## Diabetes mellitus                                            1.575103e-14
## Assault (homicide)                                           3.419433e-02
## Diseases of heart                                            2.864534e-35
## Essential hypertension and hypertensive renal disease        5.104730e-09
## Accidents (unintentional injuries)                           4.904542e-07
## Chronic liver disease and cirrhosis                          5.372529e-09
## Nephritis, nephrotic syndrome and nephrosis                  5.983553e-05
## Parkinson's disease                                          1.464092e-02
## Influenza and pneumonia                                      2.435656e-15
## Cerebrovascular diseases                                     2.452214e-17
## Intentional self-harm (suicide)                              2.292353e-03
```

From the result of linear regression, we discovered that the max_WBGT_max_monthly value has significant positive influence on the death number of every cause of death. Other factors does not have significant positive

influence on the death number. Since the max_WBGT_max_monthly can better present the occurrence of extreme hot weather, we can conclude that the extreme how weather greatly damages the health of the elderly.

4.3 Regression Model for heat map data and death-related data

4.3.1 Load data and extract useful data.

```
# Input of data
heat_map = read.csv('countywise_Heatmap.csv')
death = read.csv('heat_death_2011_2015.csv')
```

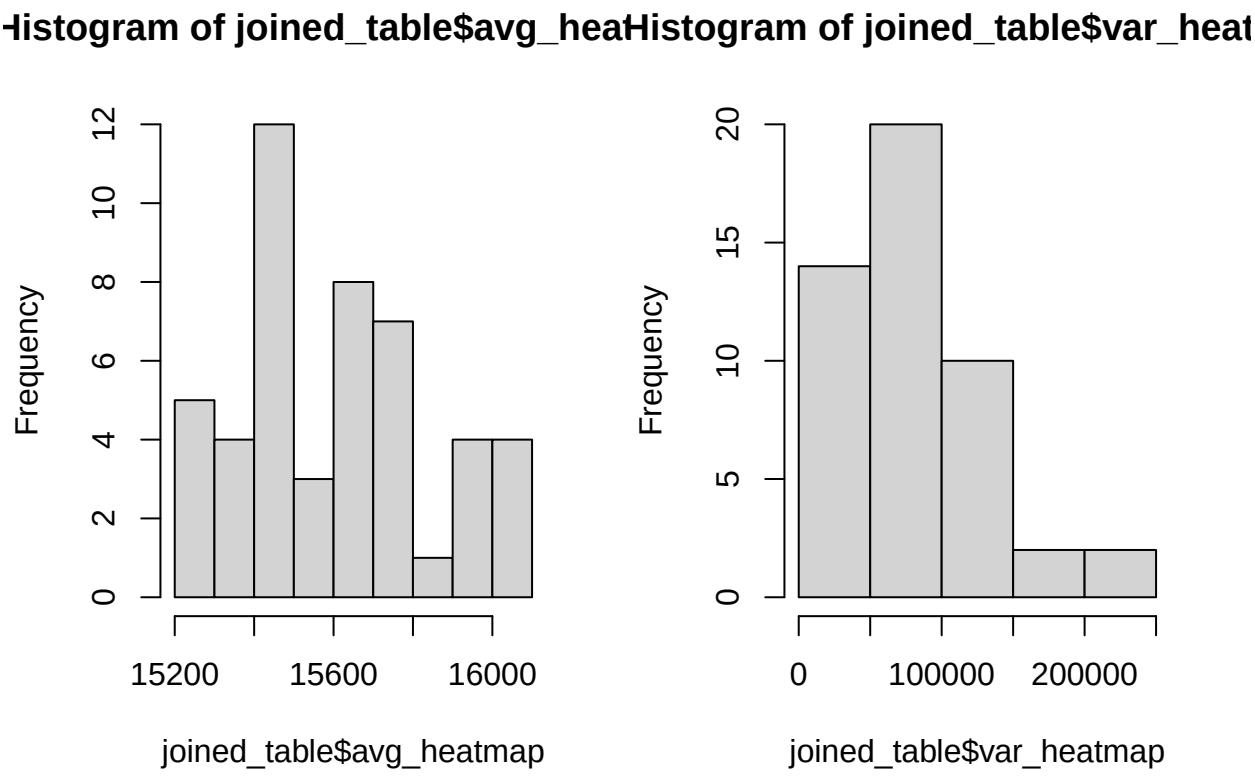
The data contains the heatmap data of each california county. We have the average heatmap index and variance heatmap index. We intended to detect the heat island phenomenon by the average and variance of heatmap index.

```
death_m=death[death$Year==2013&death$Month==7,]
death_m = death_m[,c(3,8,9,10)]
colnames(death_m)<-c("county","death","total","elderly")
joined_table=merge(heat_map,death_m, by = "county")
```

Since the heatmap table contains data from July 20th to July 27th in 2013. We just find the death number of July 2013 to analysis the relationship between heatmap and death. Then we join the heatmap table and death table by the county.

4.3.2 Exploratory analysis

```
#The distribution of average heatmap index and variance heatmap index.
par(mfrow=c(1,2))
hist(joined_table$avg_heatmap)
hist(joined_table$var_heatmap)
```




```r
#The relationship between death proportion of the elderly and the average and the variance of heatmap i
m1=lm(death/elderly~avg_heatmap+var_heatmap,data = joined_table)
summary(m1)
```

```
##
## Call:
## lm(formula = death/elderly ~ avg_heatmap + var_heatmap, data = joined_table)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0023393 -0.0017286  0.0005228  0.0013400  0.0031202
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.005e-02  1.607e-02  -0.626    0.535
## avg_heatmap  7.526e-07  1.035e-06   0.727    0.471
## var_heatmap  4.284e-09  4.949e-09   0.866    0.391
##
## Residual standard error: 0.001657 on 45 degrees of freedom
## Multiple R-squared:  0.03531,    Adjusted R-squared:  -0.007569
## F-statistic: 0.8235 on 2 and 45 DF,  p-value: 0.4454
```

This shows the death proportion of the elderly has no significant relationship with the heat map index.

```r
#The relationship between the total population and the average and the variance of heatmap index
m2=lm(total~avg_heatmap+var_heatmap,data = joined_table)
summary(m2)
```

```
##
## Call:
## lm(formula = total ~ avg_heatmap + var_heatmap, data = joined_table)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1307438  -653434  -229848   187547  8823820
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.871e+07  1.459e+07  -1.967   0.0553 .
## avg_heatmap  1.865e+03  9.402e+02   1.984   0.0534 .
## var_heatmap  2.997e+00  4.495e+00   0.667   0.5083
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1505000 on 45 degrees of freedom
## Multiple R-squared:  0.1044, Adjusted R-squared:  0.06463
## F-statistic: 2.624 on 2 and 45 DF,  p-value: 0.08359
```

This shows that the total population tends to be larger when the heatmap index is higher. Again, the variance of heatmap index has no relationship with the total population.

```r
#The relationship between the elderly proportion and the average and the variance of heatmap index.
m3=lm(elderly/total~avg_heatmap+var_heatmap,data = joined_table)
summary(m3)
```

```
##
## Call:
## lm(formula = elderly/total ~ avg_heatmap + var_heatmap, data = joined_table)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.068506 -0.024444 -0.004764  0.029230  0.076924
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.499e+00  3.487e-01   4.299 9.08e-05 ***
## avg_heatmap -8.547e-05  2.246e-05  -3.805 0.000426 ***
## var_heatmap -2.260e-07  1.074e-07  -2.104 0.040973 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03597 on 45 degrees of freedom
## Multiple R-squared:  0.3454, Adjusted R-squared:  0.3163
## F-statistic: 11.87 on 2 and 45 DF,  p-value: 7.245e-05
```

This model shows that with the heatmap index increasing, the elderly proportion tends to decrease. This implies that the elderly tends not to live in a hot place. The variance of heatmap index also has negative

effect on the elderly proportion. The variance of heatmap index indicates the change of heatmap index. Higher variance of heatmap may mean that there is a heat island (A heat island means there will be a place that has a abnormally higher temperature than the surrounding areas) As a result, it is possible that the heat island has negative influence on the health of the elderly, so the elderly tends not to live there.