**The Dataset of The Climate Changes in the recent 10 years in U.S.**

Menglei Chen

# INTRODUCTION

The climate on earth has changed throughout history, and global warming is always a controversy among scientists. Except the temperate changes, there are also other climate changes exist. How did these changes effect the local people is of interest of scientists? I created a dataset contains the daily features about climates for approximately 1,600 U.S. locations in the recent 10 years and economic behaviors of local people. It may be helpful for the researchers to analyze the climate effect on local people or climate related projects.

# data sources

## Climate

Climate data from Local Climatological Data (LCD) released by National Centers For Environmental Information:

<https://www.ncdc.noaa.gov/data-access/land-based-station-data/land-based-datasets/quality-controlled-local-climatological-data-qclcd>

Including files from “QCLCD200705.zip” to “QCLCD201710.zip”

<https://www.ncdc.noaa.gov/orders/qclcd/>

## qgdpstate\_all\_R.zip

Regional economic data from Bureau of Economic Analysis.

Quarterly Real Gross Domestic Product (GDP) by State :

<https://www.bea.gov/regional/downloadzip.cfm>

It could not download the file automatically, because there is no URL provided directly to the file, so I downloaded it manually and attached it with the documentation.

# codebook

There are 4,271,609 observations of 46 variables. 26 variables are numeric, 3 ones are integer, 16 ones are characters, and one is date.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variable Name | Meaning | Detail | Data type | Possible values |
| WBAN | WBAN code of the station | A five-digit station identifier used at NCEI | Character | 5-digit |
| YearMonthDay | Measured Date | reported as "Local Standard Time" | Date, format:”2007-05-01” | From “2007-05-01” to “2017-10-31” |
| Tmax | Maximum temperature | reported in degrees Fahrenheit | Numeric | From -96 to 139 |
| Tmin | Minimum temperature | reported in degrees Fahrenheit | Numeric | From -99 to 104 |
| Tavg | Average temperature | reported in degrees Fahrenheit | Numeric | From -55 to 114 |
| Depart | Temperature depart | Forecast departure of Average Temperature from Normal | Numeric | From -79 to 62 |
| DewPoint | Dew Point Temperature | reported in whole degrees Celsius | Numeric | from -38 to 83 |
| WetBulb | Wet Bulb Temperature | reported in whole degrees Celsius | Numeric | From -27 to 84 |
| Heat | Heat degree days |  | Numeric | From 0 to 120 |
| Cool | Cool degree days |  | Numeric | From 0 to 49 |
| Sunrise | Local time of sunrise | Reported as "Local Standard Time" in 24-hour time system | Character | From “00:00” to “12:46” |
| Sunset | Local time of sunset | Reported as "Local Standard Time" in 24-hour time system | Character | From “12:14” to “24:51” |
| CodeSum | Significant weather code |  | Character | Sum of significant weather code |
| Depth | Snow depth | Reported in centimeter | Numeric | From 0 to 114 |
| SnowFall | Snowfall | Reported in millimeter | Numeric | From 0 to 93.1 |
| PrecipTotal | Precipitation | Reported in millimeter | Numeric | From 0 to 30 |
| StnPressure | Air pressure | Reported in inch | Numeric | From 14.7 to 31.1 |
| SeaLevel | Measured sea level | Reported in meter | Numeric | From 27.5 to 31.2 |
| ResultSpeed | Result speed of wind | Reported in km/h | Numeric | From 0 to 74.4 |
| ResultDir | Result direction of wind | Reported in 10 degrees | Numeric | From 1 to 36 |
| AvgSpeed | Average speed of wind | Reported in km/h | Numeric | From 0 to 325 |
| Max5Speed | Max5speed of wind | Reported in km/h | Numeric | From 0 to 431 |
| Max5Dir | Max5 direction of wind | Reported in degrees | Numeric | From 0 to 360 |
| Max2Speed | Max2 speed of wind | Reported in km/h | Numeric | From 1 to 345 |
| Max2Dir | Max2 direction of wind | Reported in degrees | Numeric | From 1 to 360 |
| WMO | WMO Meteorological code | A 7-digit code defined by the World Meteorological Organization in WMO Manual No. 306 | Character | 7-digit |
| CallSign | a unique designation for a transmitter station |  | Character | Code with letters and digits |
| ClimateDivisionCode | Code for the climate division |  | Character | From 0 to 13, and 61 |
| ClimateDivisionStateCode | Code for the climate division state |  | Character | From 1 to 51, and 66, 67, 91 |
| ClimateDivisionStationCode | Code for the climate division station |  | Character | From 16 to 9941 |
| Name | Station name |  | Character | Station name |
| State | Station state abbreviation |  | Character | State abbreviation |
| Location | Station location |  | Character | Location |
| Latitude | Station latitude | Reported in degrees | Numeric | From -14.33 to 71.32 |
| Longitude | Station longitude | Reported in degrees | Numeric | From -176.65 to 174.10 |
| GroundHeight | Measure ground height | Reported in inch | Numeric | From -122 to 12074 |
| StationHeight | Station height | Reported in inch | Numeric | From -118 to 12074 |
| Barometer | Barometer |  | Numeric | From -116 to 9938 |
| TimeZone | Station time zone | Related to UTC | Character | From -12 to -4 and 9 to 12 |
| Year | Measured year |  | Integer | From 2007 to 2017 |
| Month | Measured month |  | Integer | From 1 to 12 |
| Day | Measured day of the month |  | Integer | From 1 to 31 |
| Weekday | Measured weekday |  | Character | From Monday to Sunday |
| Quarter | Measured quarter of the year |  | Character | From Q1 to Q4 |
| StateName | Station state name |  | Character | State name |
| All industry total | Quarterly real GDP by state in all industry | Millions of chained 2099 dollars | Numeric | From 25219 to 2368261 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

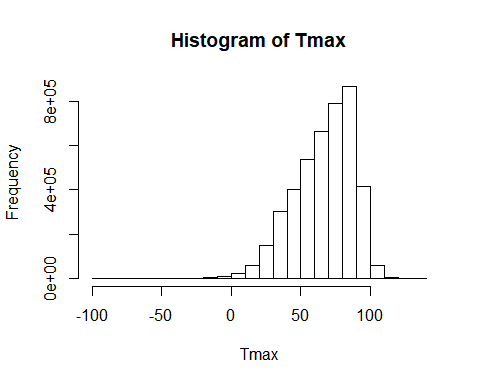
## WBAN—Character

5-digit station identifier

## YearMonthDay—Date, format:”2007-05-01”

From “2007-05-01” to “2017-10-31”

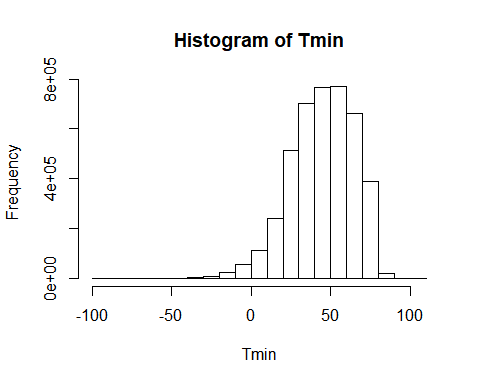
## Tmax-- Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max.

-96.00 53.00 70.00 66.99 84.00 139.00

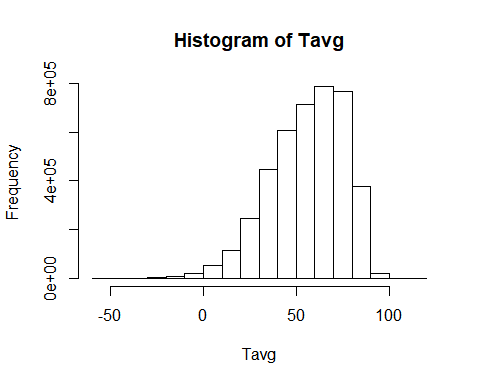
## Tmin-- Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max.

-99.0 32.0 47.0 45.3 61.0 104.0

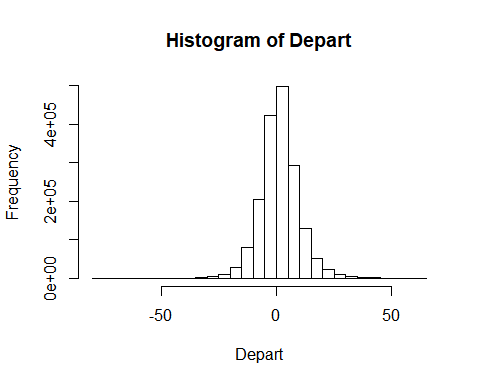
## Tavg—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

-55.00 43.00 59.00 56.37 72.00 114.00 105321

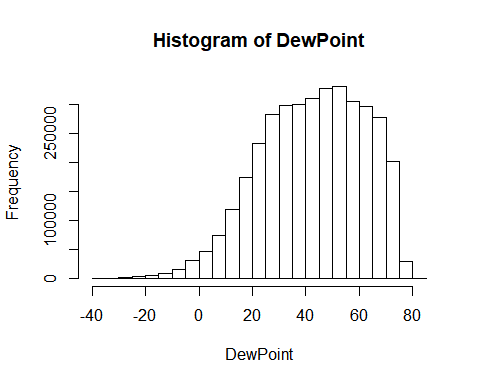
## Depart—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

-79.0 -3.0 2.0 2.1 7.0 62.0 2503809

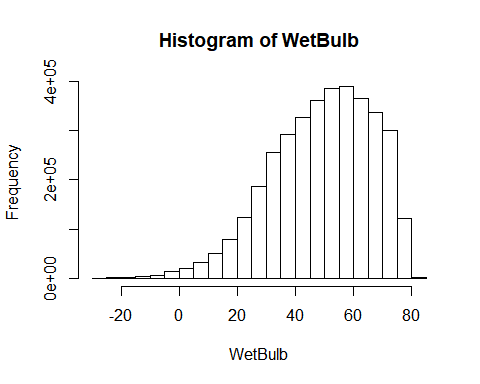
## DewPoint—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

-38.0 29.0 44.0 43.2 59.0 83.0 594635

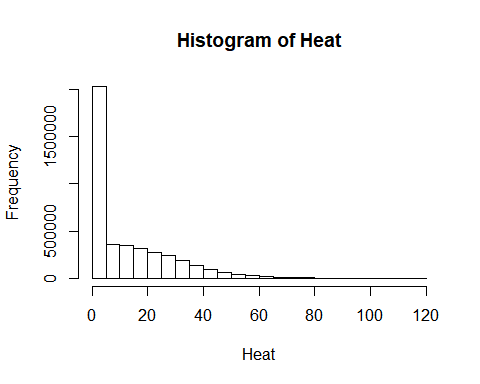
## WetBulb—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

-27.0 38.0 51.0 49.7 63.0 84.0 619411

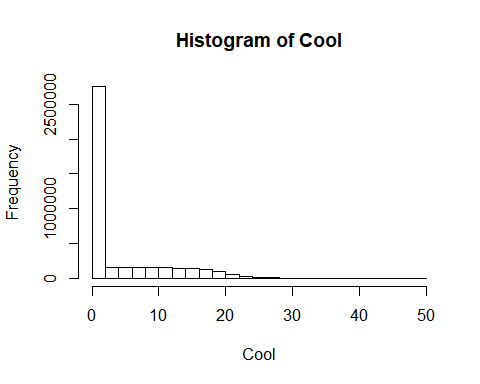
## Heat—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

0.0 0.0 6.0 12.7 22.0 120.0 105321

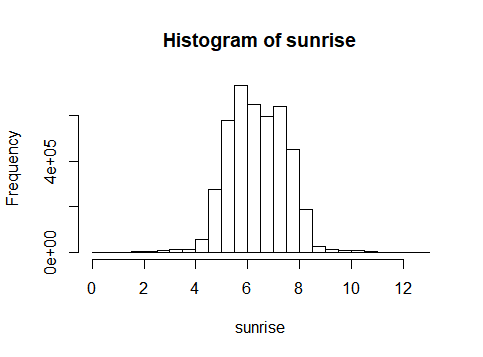
## Cool—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

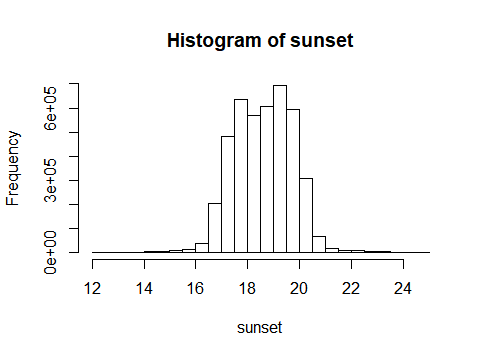
0.00 0.00 0.00 4.07 7.00 49.00 105321

## Sunrise—Character



Reported as "Local Standard Time" in 24-hour time system, and its values are from “00:00” to “12:46”.

## Sunset—Character

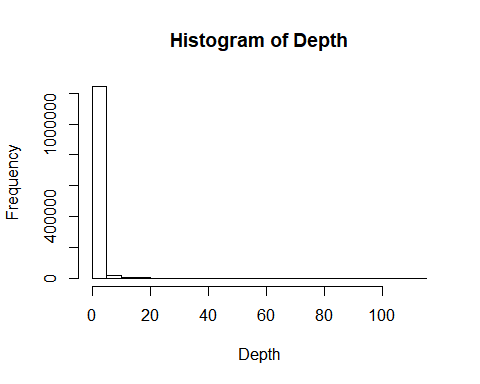


Reported as "Local Standard Time" in 24-hour time system, and its values are from “12:14” to “24:51”.

## CodeSum—Character

Sum of significant weather code (“RA”, ”BR”, ”HZ”, ”SN”, ”FG”, “FG+”, “FC”, “+FC”, ”VCTS”, ”TSRA”, ”DZ”, ”UP”, ”FZFG”, “BLDU”, “MIFG”, “VCSH”, “SQ”).

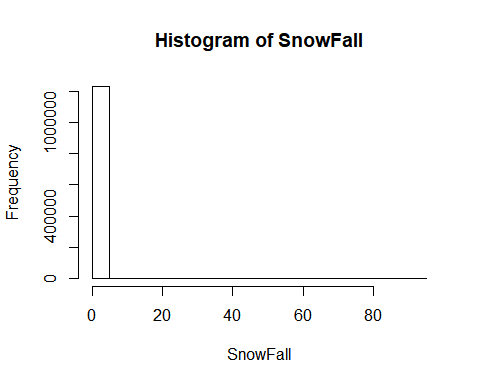
## Depth—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

0.0 0.0 0.0 0.4 0.0 114.0 2995265

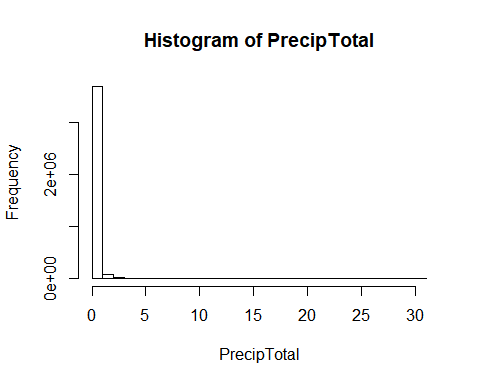
## SnowFall—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

0.0 0.0 0.0 0.1 0.0 93.1 3038165

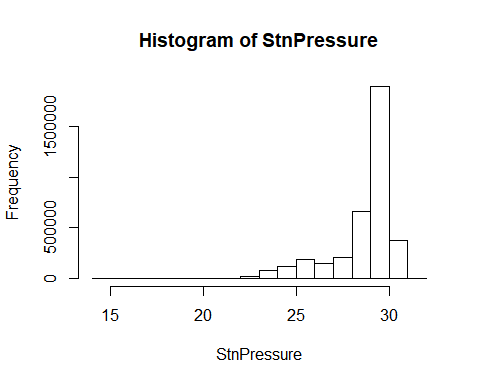
## PrecipTotal—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

0.0 0.0 0.0 0.1 0.0 30.0 483932

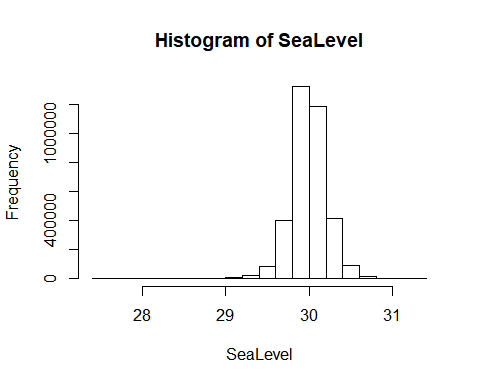
## StnPressure—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

14.7 28.4 29.3 28.7 29.8 31.1 571574

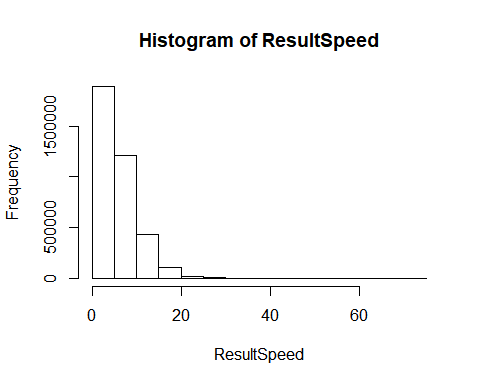
## SeaLevel—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

27.5 29.9 30.0 30.0 30.1 31.2 741768

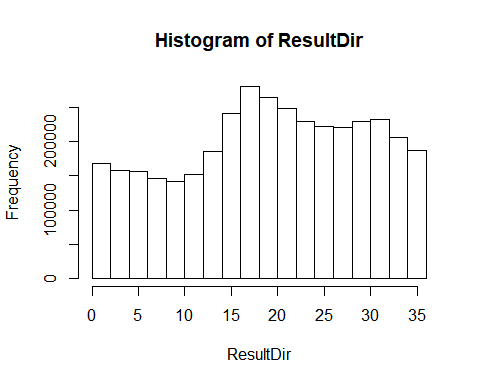
## ResultSpeed—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

0.0 2.5 4.9 5.8 8.1 74.4 599084

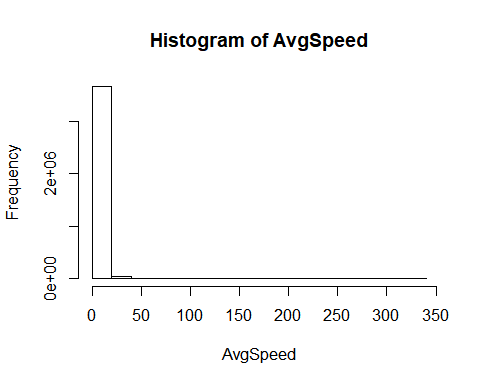
## ResultDir—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

1.0 12.0 20.0 19.7 28.0 36.0 599084

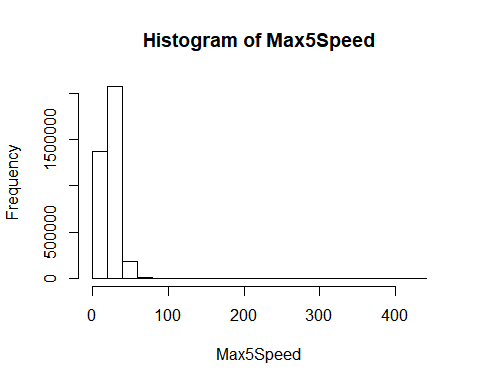
## AvgSpeed—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

0.0 4.4 6.7 7.4 9.6 325.0 566684

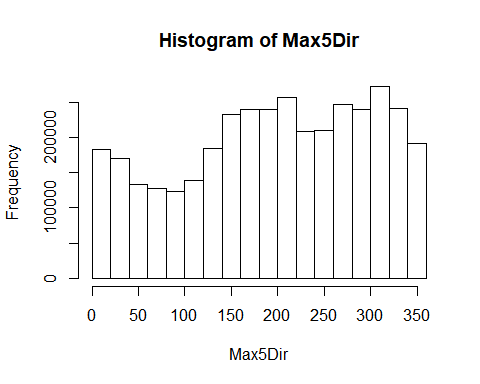
## Max5Speed—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

0.0 17.0 23.0 24.4 29.0 431.0 630339

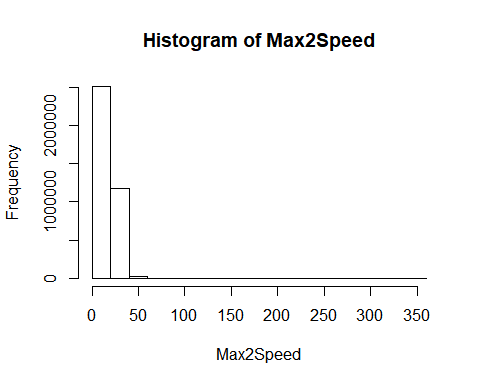
## Max5Dir—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

0.0 130.0 210.0 201.6 290.0 360.0 631809

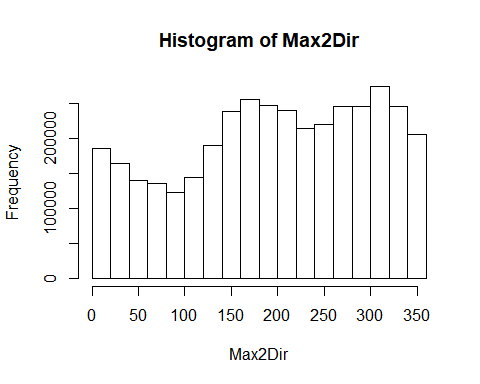
## Max2Speed—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

1 13 17 18 22 345 558340

## Max2Dir—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

1.0 130.0 210.0 201.9 290.0 360.0 562667

## WMO—Character

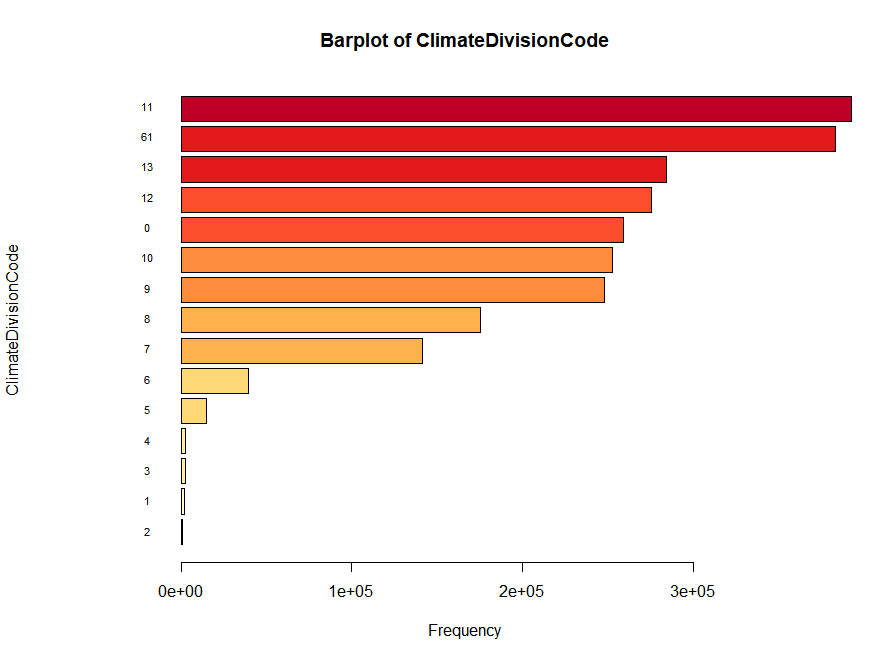
7-digit WMO Meteorological code

## CallSign—Character

Code with letters and digits

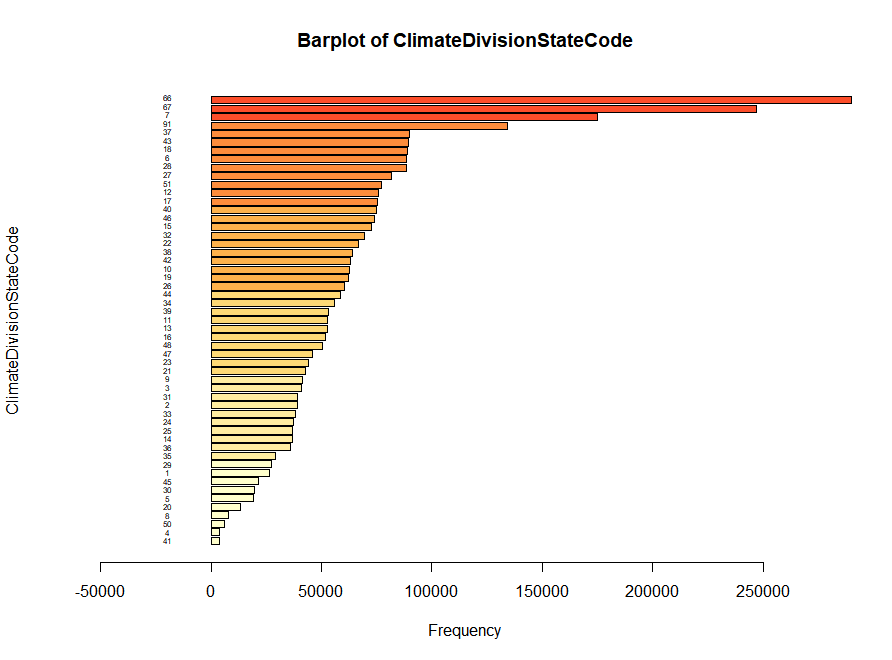
## ClimateDivisionCode—Character

From 0 to 13, and 61



## ClimateDivisionStateCode—Character

From 1 to 51, and 66, 67, 91



## ClimateDivisionStationCode—Character

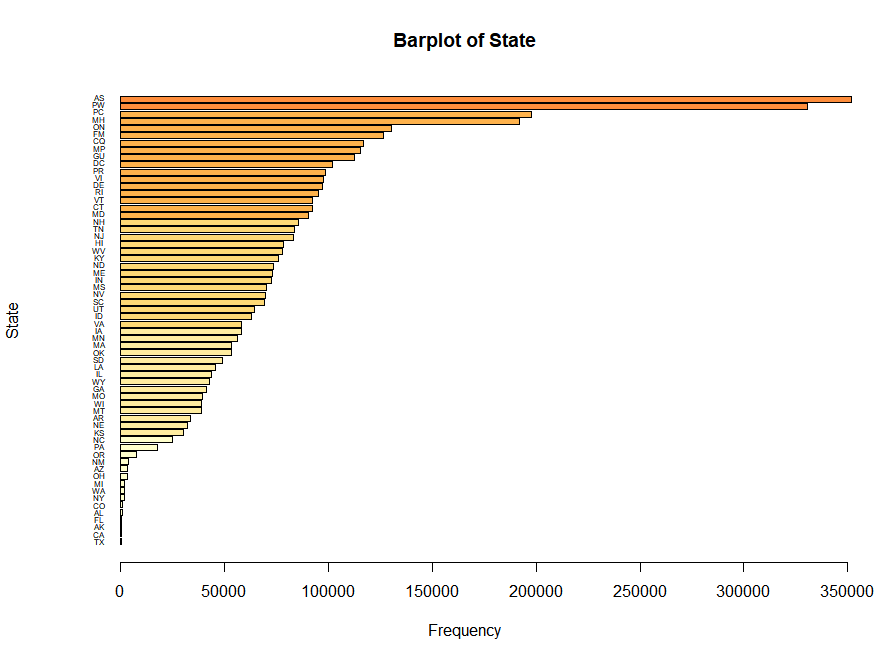
From 16 to 9941

## Name—Character

Station name

## State—Character

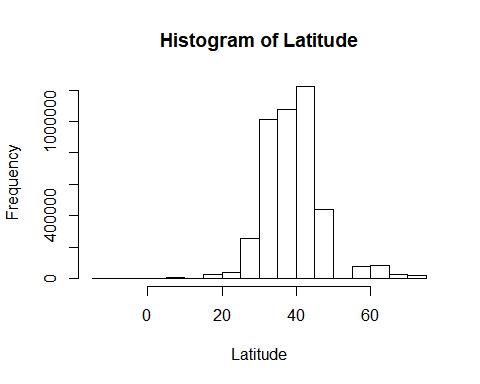
State abbreviation



## Location—Character

Detailed station location

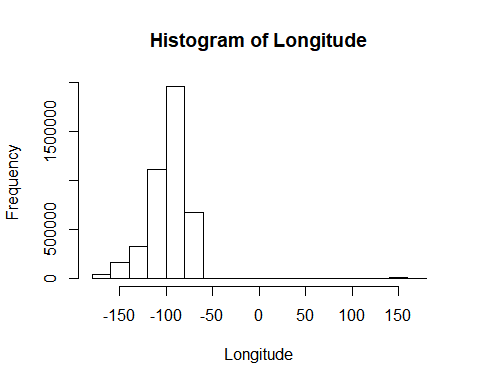
## Latitude—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max.

-14.33 33.94 38.91 39.00 42.80 71.32

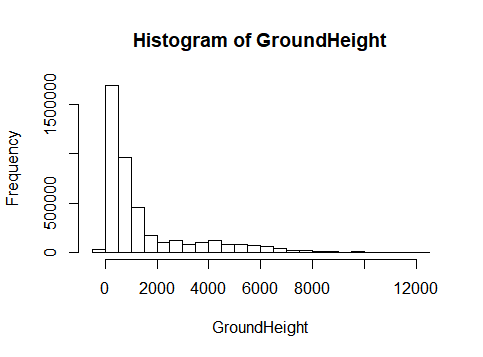
## Longitude—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max.

-176.65 -110.42 -95.28 -97.82 -83.57 174.10

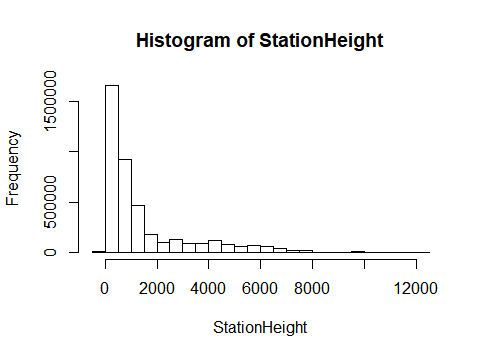
## GroundHeight—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

-122 165 700 1422 1624 12074 13630

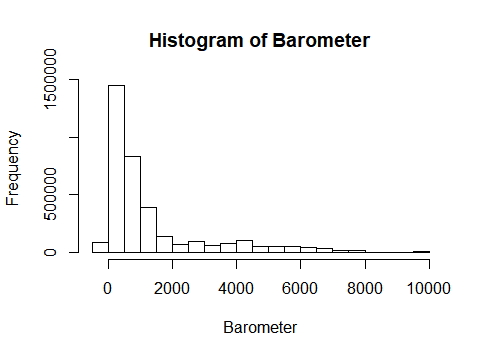
## StationHeight—Numeric



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

-118 172 704 1367 1540 12074 164953

## Barometer—Numeric

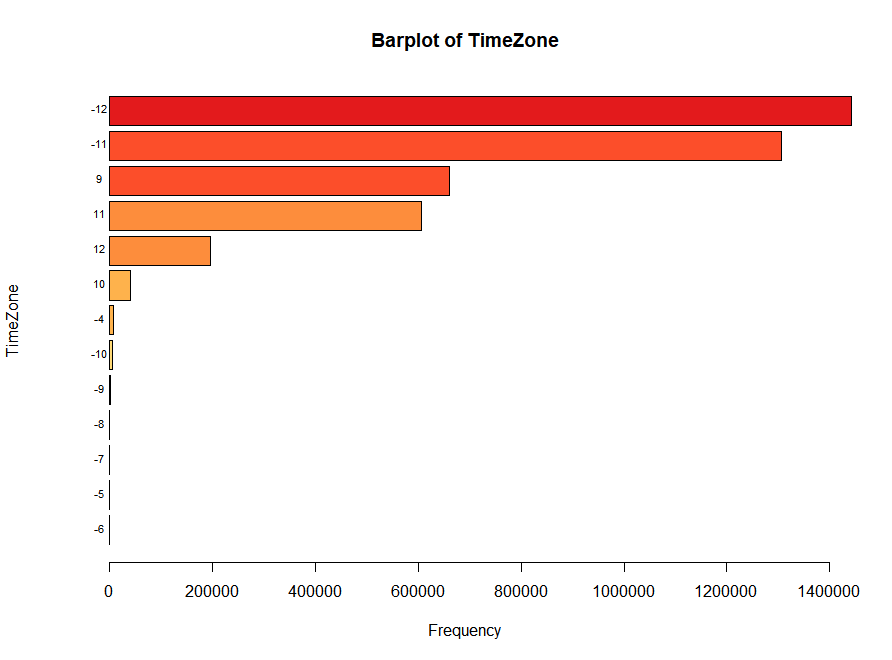


Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

-116 129 653 1247 1326 9938 701626

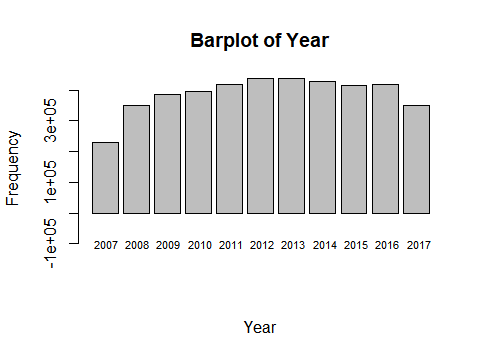
## TimeZone—Character

From -12 to -4 and 9 to 12



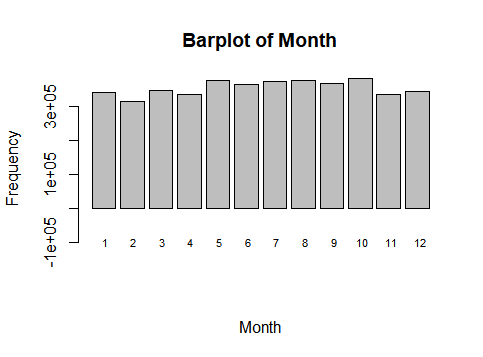
## Year—Integer

From 2007 to 2017.



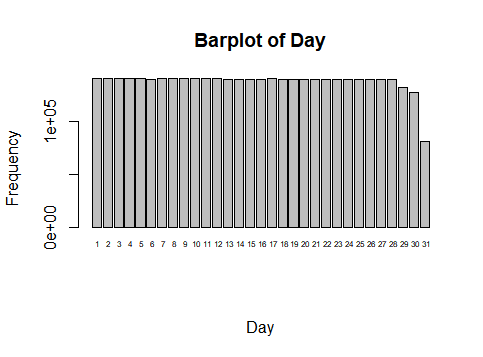
## Month—Integer

From 1 to 12.



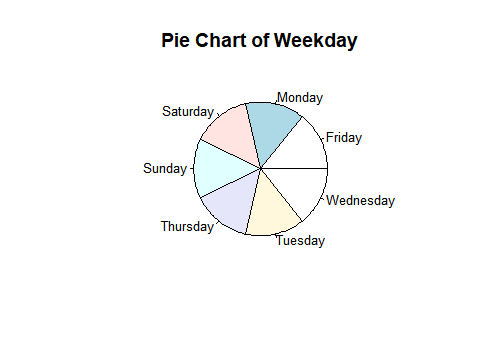
## Day-- Integer

From 1 to 31.



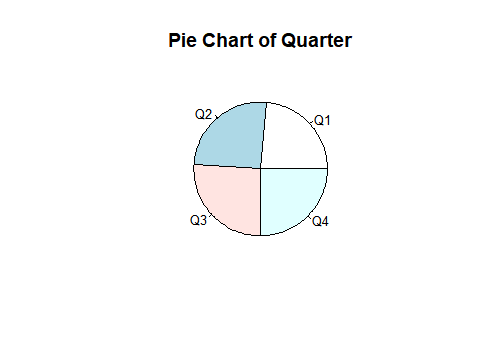
## Weekday—Character

From Monday to Sunday.

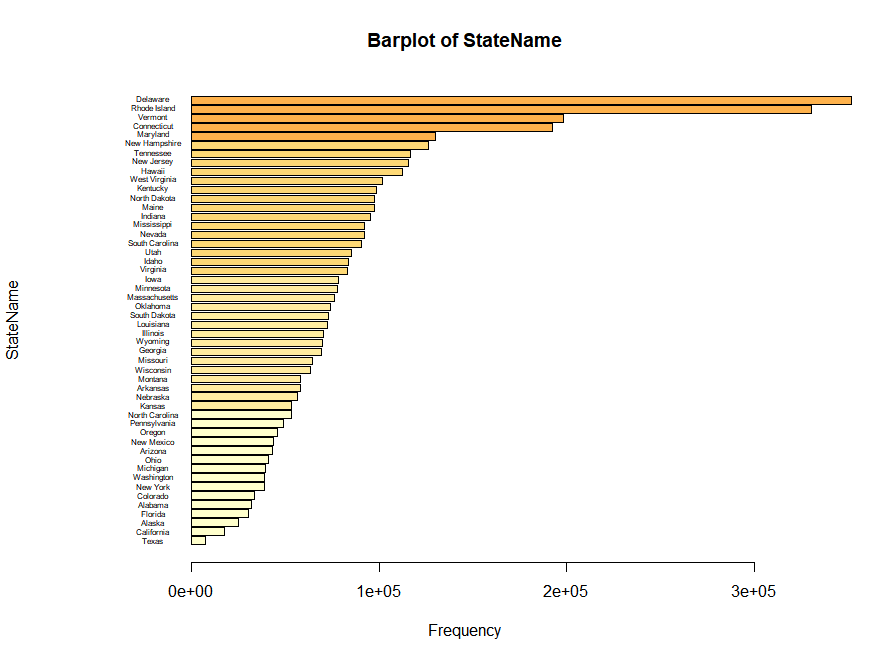


## Quarter—Character

From Q1 to Q4.

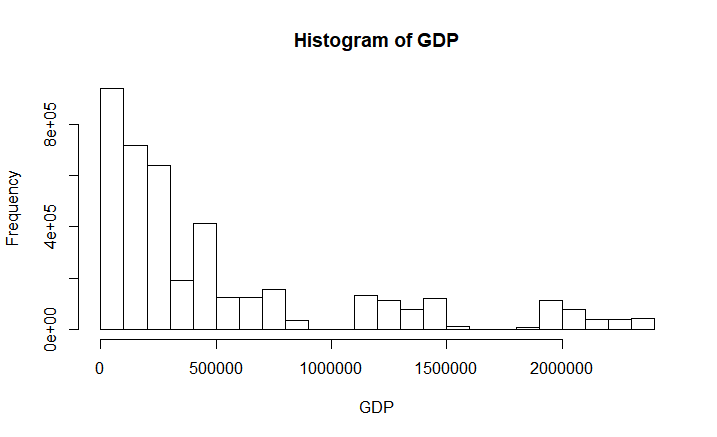


## StateName-- Character



## All industry total—Numeric

Quarterly real GDP by state (Millions of chained 2009 dollars)



Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

25219 109902 261887 516013 649490 2368261 162595

# method

## Climate data creation

### Climate data collecting

1. Download one climate monthly summary .zip file using download.file() function from the LCD source.

zipn<-paste("QCLCD",j,i,".zip",sep="")  
 url<-paste("https://www.ncdc.noaa.gov/orders/qclcd/",zipn,sep="")  
 trydownload<-try(download.file(url,zipn),silent=T)

1. Unzip the file and merge “daily.txt” with “station.txt” to get a “subclimate” dataset.

dailyn<-paste(j,i,"daily.txt",sep="")  
 stationn<-paste(j,i,"station.txt",sep="")  
 unzip(zipn,c(dailyn,stationn),overwrite = T)  
 daily<-read.csv(dailyn,header=T,stringsAsFactors = F)  
 station<-read.csv(stationn,sep="|",header=T,stringsAsFactors = F)  
 daily<-daily[,-b]  
 daily <- daily[order(daily$WBAN),]  
 station <- station[order(station$WBAN),]  
 subclimate<-merge(daily,station,by.x="WBAN",by.y="WBAN",all.x=T,all.y=F,sort=T)

1. Iterate step 1 and step 2. Merge all the “subclimate” datasets to get a “climate” dataset from May 1st 2007 till October 31st 2017.

if (j!=y[1] | i!=m[5]) {  
 climate\_test<-rbind(climate\_test,subclimate)  
 }else{  
 climate\_test<-subclimate  
 }

1. Remove the useless dataset in each loop.

rm(daily,station,subclimate,url,zipn,dailyn,stationn,trydownload)

### Deal with missing data and invalid data

For each variable

1. Using table() function to see if there exists invalid or missing data.
2. Using Regular expressions to transform the invalid data and missing data.

sub=gsub("\\\*|[M]|\\s","",cbind(climate$Tmax,climate$Tmin,climate$Tavg,climate$Depart))  
climate$Tmax<-sub[,1]  
climate$Tmin<-sub[,2]  
climate$Tavg<-sub[,3]  
climate$Depart<-sub[,4]  
rm(sub)  
  
mark=which(climate$Tmax=='')  
climate<-climate[-mark,]  
  
mark=which(climate$Tmin=='')  
climate<-climate[-mark,]  
  
climate$DewPoint[climate$DewPoint %in% c('-','M')]<-''  
climate$WetBulb[climate$WetBulb %in% c('-','M')]<-''  
  
climate$Heat=gsub("[M]|\\s","",climate$Heat)  
  
climate$Cool=gsub("[M]|\\s","",climate$Cool)  
  
climate$CodeSum[climate$CodeSum=='-']<-''  
  
climate$Depth=gsub("\\-|[MT]|\\s","",climate$Depth)  
  
climate<-climate[,-15]#all values of variable Water1 are missing  
  
climate$SnowFall=gsub("\\-|[A-z]|\\s|\\\*","",climate$SnowFall)  
  
climate$PrecipTotal=gsub("\\-|[A-z]|\\s|\\\*","",climate$PrecipTotal)  
  
climate$StnPressure=gsub("\\-|[A-z]|\\s|\\\*","",climate$StnPressure)  
  
climate$SeaLevel=gsub("\\-|[A-z]|\\s|\\\*","",climate$SeaLevel)  
  
climate$ResultSpeed=gsub("\\-|[A-z]|\\s|\\\*","",climate$ResultSpeed)  
  
climate$ResultDir=gsub("\\-|[A-z]|\\s|\\\*","",climate$ResultDir)  
  
climate$AvgSpeed=gsub("\\-|[A-z]|\\s|\\\*","",climate$AvgSpeed)  
  
climate$Max5Speed=gsub("\\-|[A-z]|\\s|\\\*","",climate$Max5Speed)  
  
climate$Max5Dir=gsub("\\-|[A-z]|\\s|\\\*","",climate$Max5Dir)  
  
climate$Max2Speed=gsub("\\-|[A-z]|\\s|\\\*","",climate$Max2Speed)  
  
climate$Max2Dir=gsub("\\-|[A-z]|\\s|\\\*","",climate$Max2Dir)  
  
climate$WMO=gsub("\\-|[A-z]|\\s|\\\*","",climate$WMO)  
  
climate$State=gsub("\\s","",climate$State)  
mark=which(climate$State=='')  
climate<-climate[-mark,]  
  
climate$GroundHeight[climate$GroundHeight==99999]<-NA  
climate$StationHeight[climate$StationHeight==99999]<-NA  
climate$Barometer[climate$Barometer==99999]<-NA  
climate$Barometer[climate$Barometer==9999]<-NA

### Add date related variables library(lubridate)

climate$YearMonthDay<-as.Date(as.character(climate$YearMonthDay),"%Y%m%d")  
climate$Year<-year(climate$YearMonthDay)  
climate$Month<-month(climate$YearMonthDay)  
climate$Day<-day(climate$YearMonthDay)  
climate$Weekday<-weekdays(climate$YearMonthDay)  
climate$Quarter<-quarters(climate$YearMonthDay)

### Add Sunrise and Sunset

library(RAtmosphere)  
sunrise<-suncalc(yday(climate$YearMonthDay),climate$Latitude, climate$Longitude,UTC=F)$sunrise

climate$Sunrise<-paste(sprintf("%02d",trunc(sunrise)),":",sprintf("%02d",trunc((sunrise-trunc(sunrise))\*60)),sep="")  
climate$Sunrise[climate$Sunrise=="NA:NA"]<-"NA" #NaNs produced cause some places don't have sunrise  
  
sunset<-suncalc(yday(climate$YearMonthDay),climate$Latitude, climate$Longitude,UTC=F)$sunset

climate$Sunset<-paste(sprintf("%02d",trunc(sunset)),":",sprintf("%02d",trunc((sunset-trunc(sunset))\*60)),sep="")  
climate$Sunset[climate$Sunset=="NA:NA"]<-"NA" #NaNs produced cause some places don't have sunset

### Add state names

climate$StateName<-state.name[match(climate$State,state.abb)]

### Transform the format

According to each variable’s meaning, transform their format to the suitable data type

##transform characters to numerics  
climate$Tmax<-as.numeric(climate$Tmax)  
climate$Tmin<-as.numeric(climate$Tmin)  
climate$Tavg<-as.numeric(climate$Tavg)  
climate$Depart<-as.numeric(climate$Depart)  
climate$DewPoint<-as.numeric(climate$DewPoint)  
climate$WetBulb<-as.numeric(climate$WetBulb)  
climate$Heat<-as.numeric(climate$Heat)  
climate$Cool<-as.numeric(climate$Cool)  
climate$Depth<-as.numeric(climate$Depth)  
climate$SnowFall<-as.numeric(climate$SnowFall)  
climate$PrecipTotal<-as.numeric(climate$PrecipTotal)  
climate$StnPressure<-as.numeric(climate$StnPressure)  
climate$SeaLevel<-as.numeric(climate$SeaLevel)  
climate$ResultSpeed<-as.numeric(climate$ResultSpeed)  
climate$ResultDir=as.numeric(climate$ResultDir)  
climate$AvgSpeed=as.numeric(climate$AvgSpeed)  
climate$Max5Speed=as.numeric(climate$Max5Speed)  
climate$Max5Dir=as.numeric(climate$Max5Dir)  
climate$Max2Speed=as.numeric(climate$Max2Speed)  
climate$Max2Dir=as.numeric(climate$Max2Dir)  
climate$Barometer<-as.numeric(climate$Barometer)  
climate$StationHeight<-as.numeric(climate$StationHeight)  
##transform numerics to characters   
climate$WBAN<-sprintf("%05d",climate$WBAN)  
climate$WMO<-sprintf("%07d",as.integer(climate$WMO))  
climate$ClimateDivisionCode<-as.character(climate$ClimateDivisionCode)  
climate$ClimateDivisionStateCode<-as.character(climate$ClimateDivisionStateCode)  
climate$ClimateDivisionStationCode<-as.character(climate$ClimateDivisionStationCode)  
climate$TimeZone<-as.character(climate$TimeZone)  
##transform numerics to integers   
climate$Year<-as.integer(climate$Year)  
climate$Month<-as.integer(climate$Month)

### Create the id for later merging

paste(climate$StateName,climate$Year,climate$Quarter,sep='')

### Order the data by id

climate <- climate[order(climate$id),]

### Check the dataset structure

str(climate)

## GDP data creation

### GDP data collecting

1. unzip and read the quarterly state gdp data file

unzip("qgdpstate\_all\_R.zip","qgdpstate\_all\_R.csv",overwrite = T)  
qgdpstate<-read.csv("qgdpstate\_all\_R.csv",header=T,stringsAsFactors = F)

1. get the industry id and their description, and saved them into dataset industry

mark=which(qgdpstate$GeoFIPS=="00000")  
industry<-qgdpstate[mark,c(6,8)]

1. eliminate nonrelated data

mark=which(qgdpstate$GeoName %in% state.name)  
qgdpstate<-qgdpstate[mark,]  
  
mark1=grep("2005",names(qgdpstate))  
mark2=grep("2006",names(qgdpstate))  
qgdpstate<-qgdpstate[,-c(1,3,4,5,mark1,mark2)]

IndustryId Description

1 1 All industry total

2 2 Private industries

3 3 Agriculture, forestry, fishing, and hunting

4 6 Mining

5 10 Utilities

6 11 Construction

7 12 Manufacturing

8 13 Durable goods manufacturing

9 25 Nondurable goods manufacturing

10 34 Wholesale trade

11 35 Retail trade

12 36 Transportation and warehousing

13 45 Information

14 51 Finance and insurance

15 56 Real estate and rental and leasing

16 60 Professional, scientific, and technical services

17 64 Management of companies and enterprises

18 65 Administrative and waste management services

19 69 Educational services

20 70 Health care and social assistance

21 75 Arts, entertainment, and recreation

22 78 Accommodation and food services

23 81 Other services, except government

24 82 Government

### Merge climate dataset with GDP

1. Eliminate nonrelated data

mark=which(qgdpstate$IndustryId==industry$IndustryId[i])  
 total<-qgdpstate[mark,]  
 total<-total[,-c(2:5)]

1. stack the data

stotal<-data.frame(total[,1], stack(total[,2:ncol(total)]),stringsAsFactors = F)  
 names(stotal)<-c("StateName",industry$Description[i],"time")  
 stotal$Year<-as.integer(substr(stotal$time,2,5))  
 stotal$Quarter<-substr(stotal$time,7,8)  
 stotal<-stotal[,-3]]

1. create the id used for merging

stotal$id<-paste(stotal$StateName,stotal$Year,stotal$Quarter,sep='')  
 stotal<-stotal[,-c(1,3,4)]

1. merge the climate with the stotal

if (i==1){  
 #merge the climate with the stotal  
 final<-merge(climate,stotal,by.x="id",by.y="id",all.x=T,all.y=F,sort=T)  
 #transform GDP into numeric  
 n=ncol(final)  
 final[,n]<-as.numeric(final[,n])  
 #remove useless climate dataset  
 rm(climate)  
 }else{  
 #merge the final with the stotal  
 final<-merge(final,stotal,by.x="id",by.y="id",all.x=T,all.y=F,sort=T)  
 #transform GDP into numeric  
 final[,n]<-as.numeric(final[,n])  
 }

### Iterate for different industry id

# Appendix

#download daily weather from 200705 to 201710

y<-as.character(2007:2017)

m<-c(paste("0",as.character(1:9),sep=""),"10","11","12")

a<-c(2:25)

b<-2\*a

for (j in y[1:11]) {

for (i in m[1:12]) {

if (j==y[11] & i==m[11]){

break

}else{}

if (j==y[1] & i<m[5]){

next

}else{}

zipn<-paste("QCLCD",j,i,".zip",sep="")

url<-paste("https://www.ncdc.noaa.gov/orders/qclcd/",zipn,sep="")

trydownload<-try(download.file(url,zipn),silent=T)

if ('try-error' %in% class(trydownload)) {

file.remove(zipn)

rm(zipn,url,trydownload)

next

}else{

dailyn<-paste(j,i,"daily.txt",sep="")

stationn<-paste(j,i,"station.txt",sep="")

unzip(zipn,c(dailyn,stationn),overwrite = T)

daily<-read.csv(dailyn,header=T,stringsAsFactors = F)

station<-read.csv(stationn,sep="|",header=T,stringsAsFactors = F)

daily<-daily[,-b]

daily <- daily[order(daily$WBAN),]

station <- station[order(station$WBAN),]

subclimate<-merge(daily,station,by.x="WBAN",by.y="WBAN",all.x=T,all.y=F,sort=T)

if (j!=y[1] | i!=m[5]) {

climate<-rbind(climate,subclimate)

}else{

climate<-subclimate

}

file.remove(zipn,dailyn,stationn)

rm(daily,station,subclimate,url,zipn,dailyn,stationn,trydownload)

}

}

}

#save(climate,file="climate200705-201710.RData")

###########################################################

#load("climate200705-201710.RData")

#Deal with missing data and invalid data

sub=gsub("\\\*|[M]|\\s","",cbind(climate$Tmax,climate$Tmin,climate$Tavg,climate$Depart))

climate$Tmax<-sub[,1]

climate$Tmin<-sub[,2]

climate$Tavg<-sub[,3]

climate$Depart<-sub[,4]

rm(sub)

mark=which(climate$Tmax=='')

climate<-climate[-mark,]

mark=which(climate$Tmin=='')

climate<-climate[-mark,]

climate$DewPoint[climate$DewPoint %in% c('-','M')]<-''

climate$WetBulb[climate$WetBulb %in% c('-','M')]<-''

climate$Heat=gsub("[M]|\\s","",climate$Heat)

climate$Cool=gsub("[M]|\\s","",climate$Cool)

climate$CodeSum[climate$CodeSum=='-']<-''

climate$Depth=gsub("\\-|[MT]|\\s","",climate$Depth)

climate<-climate[,-15]#all values of variable Water1 are missing

climate$SnowFall=gsub("\\-|[A-z]|\\s|\\\*","",climate$SnowFall)

climate$PrecipTotal=gsub("\\-|[A-z]|\\s|\\\*","",climate$PrecipTotal)

climate$StnPressure=gsub("\\-|[A-z]|\\s|\\\*","",climate$StnPressure)

climate$SeaLevel=gsub("\\-|[A-z]|\\s|\\\*","",climate$SeaLevel)

climate$ResultSpeed=gsub("\\-|[A-z]|\\s|\\\*","",climate$ResultSpeed)

climate$ResultDir=gsub("\\-|[A-z]|\\s|\\\*","",climate$ResultDir)

climate$AvgSpeed=gsub("\\-|[A-z]|\\s|\\\*","",climate$AvgSpeed)

climate$Max5Speed=gsub("\\-|[A-z]|\\s|\\\*","",climate$Max5Speed)

climate$Max5Dir=gsub("\\-|[A-z]|\\s|\\\*","",climate$Max5Dir)

climate$Max2Speed=gsub("\\-|[A-z]|\\s|\\\*","",climate$Max2Speed)

climate$Max2Dir=gsub("\\-|[A-z]|\\s|\\\*","",climate$Max2Dir)

climate$WMO=gsub("\\-|[A-z]|\\s|\\\*","",climate$WMO)

climate$State=gsub("\\s","",climate$State)

mark=which(climate$State=='')

climate<-climate[-mark,]

climate$GroundHeight[climate$GroundHeight==99999]<-NA

climate$StationHeight[climate$StationHeight==99999]<-NA

climate$Barometer[climate$Barometer==99999]<-NA

climate$Barometer[climate$Barometer==9999]<-NA

#Add date related variables

library(lubridate)

climate$YearMonthDay<-as.Date(as.character(climate$YearMonthDay),"%Y%m%d")

climate$Year<-year(climate$YearMonthDay)

climate$Month<-month(climate$YearMonthDay)

climate$Day<-day(climate$YearMonthDay)

climate$Weekday<-weekdays(climate$YearMonthDay)

climate$Quarter<-quarters(climate$YearMonthDay)

#Add Sunrise and Sunset

library(RAtmosphere)

sunrise<-suncalc(yday(climate$YearMonthDay),climate$Latitude, climate$Longitude,UTC=F)$sunrise

climate$Sunrise<-paste(sprintf("%02d",trunc(sunrise)),":",sprintf("%02d",trunc((sunrise-trunc(sunrise))\*60)),sep="")

climate$Sunrise[climate$Sunrise=="NA:NA"]<-"NA" #NaNs produced cause some places don't have sunrise

sunset<-suncalc(yday(climate$YearMonthDay),climate$Latitude, climate$Longitude,UTC=F)$sunset

climate$Sunset<-paste(sprintf("%02d",trunc(sunset)),":",sprintf("%02d",trunc((sunset-trunc(sunset))\*60)),sep="")

climate$Sunset[climate$Sunset=="NA:NA"]<-"NA" #NaNs produced cause some places don't have sunset

#add state names

climate$StateName<-state.name[match(climate$State,state.abb)]

#transform formats

##transform characters to numerics

climate$Tmax<-as.numeric(climate$Tmax)

climate$Tmin<-as.numeric(climate$Tmin)

climate$Tavg<-as.numeric(climate$Tavg)

climate$Depart<-as.numeric(climate$Depart)

climate$DewPoint<-as.numeric(climate$DewPoint)

climate$WetBulb<-as.numeric(climate$WetBulb)

climate$Heat<-as.numeric(climate$Heat)

climate$Cool<-as.numeric(climate$Cool)

climate$Depth<-as.numeric(climate$Depth)

climate$SnowFall<-as.numeric(climate$SnowFall)

climate$PrecipTotal<-as.numeric(climate$PrecipTotal)

climate$StnPressure<-as.numeric(climate$StnPressure)

climate$SeaLevel<-as.numeric(climate$SeaLevel)

climate$ResultSpeed<-as.numeric(climate$ResultSpeed)

climate$ResultDir=as.numeric(climate$ResultDir)

climate$AvgSpeed=as.numeric(climate$AvgSpeed)

climate$Max5Speed=as.numeric(climate$Max5Speed)

climate$Max5Dir=as.numeric(climate$Max5Dir)

climate$Max2Speed=as.numeric(climate$Max2Speed)

climate$Max2Dir=as.numeric(climate$Max2Dir)

climate$Barometer<-as.numeric(climate$Barometer)

climate$StationHeight<-as.numeric(climate$StationHeight)

##transform numerics to characters

climate$WBAN<-sprintf("%05d",climate$WBAN)

climate$WMO<-sprintf("%07d",as.integer(climate$WMO))

climate$ClimateDivisionCode<-as.character(climate$ClimateDivisionCode)

climate$ClimateDivisionStateCode<-as.character(climate$ClimateDivisionStateCode)

climate$ClimateDivisionStationCode<-as.character(climate$ClimateDivisionStationCode)

climate$TimeZone<-as.character(climate$TimeZone)

##transform numerics to integers

climate$Year<-as.integer(climate$Year)

climate$Month<-as.integer(climate$Month)

#order the data by StateName

climate <- climate[order(climate$StateName),]

#check the dataset structure

str(climate)

#save(climate,file="final\_climate.RData")

#####################################################################

#add GDP

#load("final\_climate.RData")

#unzip and read the quarterly state gdp data file

unzip("qgdpstate\_all\_R.zip","qgdpstate\_all\_R.csv",overwrite = T)

qgdpstate<-read.csv("qgdpstate\_all\_R.csv",header=T,stringsAsFactors = F)

#only keep the total state GDP of each state

mark=which(is.na(qgdpstate$Region))

qgdpstate<-qgdpstate[-mark,]

mark=which(qgdpstate$GeoName %in% state.name)

qgdpstate<-qgdpstate[mark,]

mark1=grep("2005",names(qgdpstate))

mark2=grep("2006",names(qgdpstate))

qgdpstate<-qgdpstate[,-c(1,3,4,5,mark1,mark2)]

mark=which(qgdpstate$IndustryId==1)

total<-qgdpstate[mark,]

total<-total[,-c(2:5)]

#stack the data

stotal<-data.frame(total[,1], stack(total[,2:ncol(total)]),stringsAsFactors = F)

names(stotal)<-c("StateName","GDP","time")

stotal$Year<-as.integer(substr(stotal$time,2,5))

stotal$Quarter<-substr(stotal$time,7,8)

stotal<-stotal[,-3]

#create the id used for merging

stotal$id<-paste(stotal$StateName,stotal$Year,stotal$Quarter,sep='')

stotal<-stotal[,-c(1,3,4)]

climate$id<-paste(climate$StateName,climate$Year,climate$Quarter,sep='')

#merge the climate with the stotal

final<-merge(climate,stotal,by.x="id",by.y="id",all.x=T,all.y=F,sort=T)

#transform GDP into numeric

final$GDP<-as.numeric(final$GDP)

#remove useless climate dataset

rm(climate)

#elimate variable id

final<-final[,-1]

########################################################################

#save the final dataset into .csv file

write.csv(final,file="final.csv")

########################################################################

#Describe variables

n=length(final[,1])

k=length(final[1,])

str(final)

num=int=char=0

for (i in 1:k){

if(class(final[,i])=="numeric") num=num+1

else if(class(final[,i])=="integer") int=int+1

else if(class(final[,i])=="character") char=char+1

}

c(n,k,num,int,char)

##deatiled description

library(RColorBrewer)

library(knitr)

###numeric variables

for (i in 1:k ){

if(class(final[,i])=="numeric"){

hist(final[,i],xlab=paste(names(final[i])),main=paste("Histogram of",names(final[i])))

}

}

summary(final[,3])

summary(final[,4])

summary(final[,5])

summary(final[,6])

summary(final[,7])

summary(final[,8])

summary(final[,9])

summary(final[,10])

summary(final[,14])

summary(final[,15])

summary(final[,16])

summary(final[,17])

summary(final[,18])

summary(final[,19])

summary(final[,20])

summary(final[,21])

summary(final[,22])

summary(final[,23])

summary(final[,24])

summary(final[,25])

summary(final[,34])

summary(final[,35])

summary(final[,36])

summary(final[,37])

summary(final[,38])

summary(final[,47])

###integer variables

####Year

freq=table(final$Year)

freq=as.data.frame(freq)

freq<-freq[order(freq$Var1),]

barplot(freq$Freq,ylim=c(-150000,max(freq$Freq)),width=5.8,space=.2,main="Barplot of Year",ylab="Frequency",xlab="Year")

text(seq(4.3,length.out=11,by=7),rep(-100000,11),label=freq$Var1,cex=.7)

####Month

freq=table(final$Month)

freq=as.data.frame(freq)

freq<-freq[order(freq$Var1),]

barplot(freq$Freq,ylim=c(-150000,max(freq$Freq)),width=5.8,space=.2,main="Barplot of Month",ylab="Frequency",xlab="Month")

text(seq(4.3,length.out=12,by=7),-100000,label=freq$Var1,cex=.7)

####Day

freq=table(final$Day)

freq=as.data.frame(freq)

freq<-freq[order(freq$Var1),]

freq

barplot(freq$Freq,ylim=c(-30000,max(freq$Freq)),width=5.8,space=.2,main="Barplot of Day",ylab="Frequency",xlab="Day")

text(seq(4.3,length.out=31,by=7),-15000,label=freq$Var1,cex=.5)

###charactor variables

####Sunrise and Sunset

hist(sunrise)

summary(sunrise)

hist(sunset)

summary(sunset)

rm(sunrise,sunset) #remove unused values

####Weekday and Quarter

for (i in 43:44){

freq=table(final[,i])

freq=as.data.frame(freq)

pie(freq$Freq,labels=freq$Var1,main=paste("Pie Chart of",names(final[i])),cex=.8)

}

####State

freq=table(final$State)

freq=as.data.frame(freq)

freq<-freq[order(freq$Freq,decreasing=T),]

l=length(freq$Var1)

barplot(rev(freq$Freq),horiz=T,xlim=c(-20000,max(freq$Freq)),width=5.8,space=.2,main="Barplot of State",xlab="Frequency",ylab="State",col=rep(brewer.pal(9,'YlOrRd'),each=15))

text(x=-10000,seq(4.3,length.out=l,by=7),label=freq$Var1,cex=.5)

####StateName

freq=table(final$StateName)

freq=as.data.frame(freq)

freq<-freq[order(freq$Freq,decreasing=T),]

l=length(freq$Var1)

barplot(rev(freq$Freq),horiz=T,xlim=c(-60000,max(freq$Freq)),width=5.8,space=.2,main="Barplot of StateName",xlab="Frequency",ylab="StateName",col=rep(brewer.pal(9,'YlOrRd'),each=15))

text(x=-20000,seq(4.3,length.out=l,by=7),label=freq$Var1,cex=.5)

####Timezone

freq=table(final$TimeZone)

freq=as.data.frame(freq)

freq<-freq[order(freq$Freq,decreasing=T),]

l=length(freq$Var1)

barplot(rev(freq$Freq),horiz=T,xlim=c(-60000,max(freq$Freq)),width=5.8,space=.2,main="Barplot of TimeZone",xlab="Frequency",ylab="TimeZone",col=rep(brewer.pal(9,'YlOrRd'),each=2))

text(x=-20000,seq(4.3,length.out=l,by=7),label=freq$Var1,cex=.7)

####ClimateDivisionCode

freq=table(final$ClimateDivisionCode)

freq=as.data.frame(freq)

freq<-freq[order(freq$Freq,decreasing=T),]

l=length(freq$Var1)

barplot(rev(freq$Freq),horiz=T,xlim=c(-60000,max(freq$Freq)),width=5.8,space=.2,main="Barplot of ClimateDivisionCode",xlab="Frequency",ylab="ClimateDivisionCode",col=rep(brewer.pal(9,'YlOrRd'),each=2))

text(x=-20000,seq(4.3,length.out=l,by=7),label=freq$Var1,cex=.7)

####ClimateDivisionStateCode

freq=table(final$ClimateDivisionStateCode)

freq=as.data.frame(freq)

freq<-freq[order(freq$Freq,decreasing=T),]

l=length(freq$Var1)

barplot(rev(freq$Freq),horiz=T,xlim=c(-60000,max(freq$Freq)),width=5.8,space=.2,main="Barplot of ClimateDivisionStateCode",xlab="Frequency",ylab="ClimateDivisionStateCode",col=rep(brewer.pal(9,'YlOrRd'),each=10))

text(x=-20000,seq(4.3,length.out=l,by=7),label=freq$Var1,cex=.5)