Assignment 03 Getting and Cleaning Data

Erica Peng

2023-09-21

## Step 1: Move this into a good local directory for your current working directory and read and save it

getwd()

## [1] "/Users/owner/Desktop/OneDrive - nyu.edu/McDaniel/R Fall Semester"

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.3 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.3 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

# Set the working directory  
setwd("/Users/owner/Downloads")  
  
# Import the CSV file into a data frame  
data <- read.csv("StormEvents\_details-ftp\_v1.0\_d1977\_c20220425.csv.csv")  
  
# View the first few rows of the data frame  
head(data)

## BEGIN\_YEARMONTH BEGIN\_DAY BEGIN\_TIME END\_YEARMONTH END\_DAY END\_TIME  
## 1 197705 1 2000 197705 1 2000  
## 2 197705 1 2000 197705 1 2000  
## 3 197705 7 1740 197705 7 1740  
## 4 197705 7 1932 197705 7 1932  
## 5 197705 10 1555 197705 10 1555  
## 6 197705 10 1615 197705 10 1615  
## EPISODE\_ID EVENT\_ID STATE STATE\_FIPS YEAR MONTH\_NAME EVENT\_TYPE  
## 1 NA 9987488 COLORADO 8 1977 May Tornado  
## 2 NA 9987489 COLORADO 8 1977 May Tornado  
## 3 NA 9987490 COLORADO 8 1977 May Hail  
## 4 NA 9987492 COLORADO 8 1977 May Hail  
## 5 NA 9987493 COLORADO 8 1977 May Thunderstorm Wind  
## 6 NA 9987494 COLORADO 8 1977 May Hail  
## CZ\_TYPE CZ\_FIPS CZ\_NAME WFO BEGIN\_DATE\_TIME CZ\_TIMEZONE  
## 1 C 99 PROWERS 01-May-77 20:00:00 CST  
## 2 C 99 PROWERS 01-May-77 20:00:00 CST  
## 3 C 59 JEFFERSON T 07-May-77 17:40:00 CST  
## 4 C 59 JEFFERSON ATA 07-May-77 19:32:00 CST  
## 5 C 69 LARIMER ATA 10-May-77 15:55:00 CST  
## 6 C 69 LARIMER ATA 10-May-77 16:15:00 CST  
## END\_DATE\_TIME INJURIES\_DIRECT INJURIES\_INDIRECT DEATHS\_DIRECT  
## 1 01-May-77 20:00:00 0 0 0  
## 2 01-May-77 20:00:00 0 0 0  
## 3 07-May-77 17:40:00 0 0 0  
## 4 07-May-77 19:32:00 0 0 0  
## 5 10-May-77 15:55:00 0 0 0  
## 6 10-May-77 16:15:00 0 0 0  
## DEATHS\_INDIRECT DAMAGE\_PROPERTY DAMAGE\_CROPS SOURCE MAGNITUDE MAGNITUDE\_TYPE  
## 1 0 0K 0 NA 0.00 NA  
## 2 0 0K 0 NA 0.00 NA  
## 3 0 0 0 NA 1.00 NA  
## 4 0 0 0 NA 2.75 NA  
## 5 0 0 0 NA 63.00 NA  
## 6 0 0 0 NA 0.75 NA  
## FLOOD\_CAUSE CATEGORY TOR\_F\_SCALE TOR\_LENGTH TOR\_WIDTH TOR\_OTHER\_WFO  
## 1 NA NA F1 0.1 80 NA  
## 2 NA NA F1 0.1 80 NA  
## 3 NA NA 0.0 0 NA  
## 4 NA NA 0.0 0 NA  
## 5 NA NA 0.0 0 NA  
## 6 NA NA 0.0 0 NA  
## TOR\_OTHER\_CZ\_STATE TOR\_OTHER\_CZ\_FIPS TOR\_OTHER\_CZ\_NAME BEGIN\_RANGE  
## 1 NA NA NA 0  
## 2 NA NA NA 0  
## 3 NA NA NA 0  
## 4 NA NA NA 0  
## 5 NA NA NA 0  
## 6 NA NA NA 0  
## BEGIN\_AZIMUTH BEGIN\_LOCATION END\_RANGE END\_AZIMUTH END\_LOCATION BEGIN\_LAT  
## 1 NA NA 0 NA NA 38.08  
## 2 NA NA 0 NA NA 38.08  
## 3 NA NA 0 NA NA 39.65  
## 4 NA NA 0 NA NA 39.75  
## 5 NA NA 0 NA NA 40.58  
## 6 NA NA 0 NA NA 40.75  
## BEGIN\_LON END\_LAT END\_LON EPISODE\_NARRATIVE EVENT\_NARRATIVE DATA\_SOURCE  
## 1 -102.53 NA NA NA NA PUB  
## 2 -102.53 NA NA NA NA PUB  
## 3 -105.30 NA NA NA NA PUB  
## 4 -105.08 NA NA NA NA PUB  
## 5 -105.08 NA NA NA NA PUB  
## 6 -105.23 NA NA NA NA PUB

## Step 2: Limit the dataframe to the following column

newdata <- data[, c("BEGIN\_YEARMONTH", "BEGIN\_DAY",   
 "BEGIN\_TIME", "END\_YEARMONTH", "END\_DAY", "END\_TIME", "EPISODE\_ID",   
 "EVENT\_ID", "STATE", "EVENT\_TYPE", "CZ\_TYPE", "CZ\_FIPS", "CZ\_NAME", "CZ\_TIMEZONE", "SOURCE",   
 "TOR\_OTHER\_CZ\_STATE", "TOR\_OTHER\_CZ\_FIPS", "TOR\_OTHER\_CZ\_NAME", "BEGIN\_LAT", "BEGIN\_LON",  
 "END\_LAT", "END\_LON", "STATE\_FIPS")]  
  
head(newdata)

## BEGIN\_YEARMONTH BEGIN\_DAY BEGIN\_TIME END\_YEARMONTH END\_DAY END\_TIME  
## 1 197705 1 2000 197705 1 2000  
## 2 197705 1 2000 197705 1 2000  
## 3 197705 7 1740 197705 7 1740  
## 4 197705 7 1932 197705 7 1932  
## 5 197705 10 1555 197705 10 1555  
## 6 197705 10 1615 197705 10 1615  
## EPISODE\_ID EVENT\_ID STATE EVENT\_TYPE CZ\_TYPE CZ\_FIPS CZ\_NAME  
## 1 NA 9987488 COLORADO Tornado C 99 PROWERS  
## 2 NA 9987489 COLORADO Tornado C 99 PROWERS  
## 3 NA 9987490 COLORADO Hail C 59 JEFFERSON  
## 4 NA 9987492 COLORADO Hail C 59 JEFFERSON  
## 5 NA 9987493 COLORADO Thunderstorm Wind C 69 LARIMER  
## 6 NA 9987494 COLORADO Hail C 69 LARIMER  
## CZ\_TIMEZONE SOURCE TOR\_OTHER\_CZ\_STATE TOR\_OTHER\_CZ\_FIPS TOR\_OTHER\_CZ\_NAME  
## 1 CST NA NA NA NA  
## 2 CST NA NA NA NA  
## 3 CST NA NA NA NA  
## 4 CST NA NA NA NA  
## 5 CST NA NA NA NA  
## 6 CST NA NA NA NA  
## BEGIN\_LAT BEGIN\_LON END\_LAT END\_LON STATE\_FIPS  
## 1 38.08 -102.53 NA NA 8  
## 2 38.08 -102.53 NA NA 8  
## 3 39.65 -105.30 NA NA 8  
## 4 39.75 -105.08 NA NA 8  
## 5 40.58 -105.08 NA NA 8  
## 6 40.75 -105.23 NA NA 8

# Now, 'selected\_columns' contains only the specified columns.

## Step 3: Arrange the data by beginning year and month (BEGIN\_YEARMONTH)

# install.packages("dplyr")   
library(dplyr)  
  
# Arrange the DataFrame by YEARMONTH in ascending order  
arrange\_data <- newdata %>%  
 arrange(BEGIN\_YEARMONTH)  
  
head(arrange\_data)

## BEGIN\_YEARMONTH BEGIN\_DAY BEGIN\_TIME END\_YEARMONTH END\_DAY END\_TIME  
## 1 197701 9 820 197701 9 820  
## 2 197701 9 1100 197701 9 1100  
## 3 197701 6 2030 197701 6 2030  
## 4 197701 10 550 197701 10 550  
## 5 197701 3 15 197701 3 15  
## 6 197701 28 1330 197701 28 1330  
## EPISODE\_ID EVENT\_ID STATE EVENT\_TYPE CZ\_TYPE CZ\_FIPS CZ\_NAME  
## 1 NA 10049963 MISSISSIPPI Tornado C 89 MADISON  
## 2 NA 10049964 MISSISSIPPI Tornado C 129 SMITH  
## 3 NA 10001804 FLORIDA Tornado C 91 OKALOOSA  
## 4 NA 10001805 FLORIDA Tornado C 97 OSCEOLA  
## 5 NA 9990656 CALIFORNIA Tornado C 83 SANTA BARBARA  
## 6 NA 9995210 GEORGIA Hail C 127 GLYNN  
## CZ\_TIMEZONE SOURCE TOR\_OTHER\_CZ\_STATE TOR\_OTHER\_CZ\_FIPS TOR\_OTHER\_CZ\_NAME  
## 1 CST NA NA NA NA  
## 2 CST NA NA NA NA  
## 3 CST NA NA NA NA  
## 4 CST NA NA NA NA  
## 5 CST NA NA NA NA  
## 6 CST NA NA NA NA  
## BEGIN\_LAT BEGIN\_LON END\_LAT END\_LON STATE\_FIPS  
## 1 32.65 -90.03 NA NA 28  
## 2 31.92 -89.67 NA NA 28  
## 3 30.53 -86.50 NA NA 12  
## 4 28.30 -81.40 NA NA 12  
## 5 34.43 -119.82 NA NA 6  
## 6 31.18 -81.50 NA NA 13

## Step 4: Change state and county names to title case (e.g., “New Jersey” instead of “NEW JERSEY”)

# install.packages("tools")   
library(tools)  
  
# Change the "STATE" and "CZ\_NAME" columns to title case  
newdata$STATE <- toTitleCase(arrange\_data$STATE)  
newdata$CZ\_NAME <- toTitleCase(arrange\_data$CZ\_NAME)

## Step 5: Limit to the events listed by county FIPS (CZ\_TYPE of “C”) and then remove the CZ\_TYPE column

# install.packages("dplyr")   
library(dplyr)  
  
# Limit to events listed by county FIPS (CZ\_TYPE of "C")  
filtered\_data <- subset(arrange\_data, CZ\_TYPE == "C")  
  
# Remove the CZ\_TYPE column  
filtered\_data\_rcz <- select(filtered\_data, -CZ\_TYPE)

## Stpe 6: Pad the state and county FIPS with a “0” at the beginning (hint: there’s a function in stringr to do this) and then unite the two columns to make one fips column with the 5 or 6-digit county FIPS code

# install.packages("stringr")  
library(stringr)  
  
# Pad the STATE\_FIPS and CZ\_FIPS columns with "0" at the beginning  
filtered\_data$STATE\_FIPS <- str\_pad(filtered\_data\_rcz$STATE\_FIPS, width = 3, side="left", pad = "0")  
filtered\_data$CZ\_FIPS <- str\_pad(filtered\_data\_rcz$CZ\_FIPS, width = 3, pad = "0")  
  
# Unite the two columns into one FIPS column  
filtered\_data$FIPS <- paste0(filtered\_data$STATE\_FIPS, filtered\_data$CZ\_FIPS)  
  
# Now, 'filtered\_data' contains the FIPS column with 5 or 6-digit county FIPS codes  
head(filtered\_data\_rcz)

## BEGIN\_YEARMONTH BEGIN\_DAY BEGIN\_TIME END\_YEARMONTH END\_DAY END\_TIME  
## 1 197701 9 820 197701 9 820  
## 2 197701 9 1100 197701 9 1100  
## 3 197701 6 2030 197701 6 2030  
## 4 197701 10 550 197701 10 550  
## 5 197701 3 15 197701 3 15  
## 6 197701 28 1330 197701 28 1330  
## EPISODE\_ID EVENT\_ID STATE EVENT\_TYPE CZ\_FIPS CZ\_NAME CZ\_TIMEZONE  
## 1 NA 10049963 MISSISSIPPI Tornado 89 MADISON CST  
## 2 NA 10049964 MISSISSIPPI Tornado 129 SMITH CST  
## 3 NA 10001804 FLORIDA Tornado 91 OKALOOSA CST  
## 4 NA 10001805 FLORIDA Tornado 97 OSCEOLA CST  
## 5 NA 9990656 CALIFORNIA Tornado 83 SANTA BARBARA CST  
## 6 NA 9995210 GEORGIA Hail 127 GLYNN CST  
## SOURCE TOR\_OTHER\_CZ\_STATE TOR\_OTHER\_CZ\_FIPS TOR\_OTHER\_CZ\_NAME BEGIN\_LAT  
## 1 NA NA NA NA 32.65  
## 2 NA NA NA NA 31.92  
## 3 NA NA NA NA 30.53  
## 4 NA NA NA NA 28.30  
## 5 NA NA NA NA 34.43  
## 6 NA NA NA NA 31.18  
## BEGIN\_LON END\_LAT END\_LON STATE\_FIPS  
## 1 -90.03 NA NA 28  
## 2 -89.67 NA NA 28  
## 3 -86.50 NA NA 12  
## 4 -81.40 NA NA 12  
## 5 -119.82 NA NA 6  
## 6 -81.50 NA NA 13

## Step 7: Change all the column names to lower case (you may want to try the rename\_all function for this)

# install.packages("dplyr")   
library(dplyr)  
  
# Change all column names to lowercase using rename\_all  
newname\_data <- filtered\_data\_rcz %>%  
 rename\_all(tolower)  
  
head(newname\_data)

## begin\_yearmonth begin\_day begin\_time end\_yearmonth end\_day end\_time  
## 1 197701 9 820 197701 9 820  
## 2 197701 9 1100 197701 9 1100  
## 3 197701 6 2030 197701 6 2030  
## 4 197701 10 550 197701 10 550  
## 5 197701 3 15 197701 3 15  
## 6 197701 28 1330 197701 28 1330  
## episode\_id event\_id state event\_type cz\_fips cz\_name cz\_timezone  
## 1 NA 10049963 MISSISSIPPI Tornado 89 MADISON CST  
## 2 NA 10049964 MISSISSIPPI Tornado 129 SMITH CST  
## 3 NA 10001804 FLORIDA Tornado 91 OKALOOSA CST  
## 4 NA 10001805 FLORIDA Tornado 97 OSCEOLA CST  
## 5 NA 9990656 CALIFORNIA Tornado 83 SANTA BARBARA CST  
## 6 NA 9995210 GEORGIA Hail 127 GLYNN CST  
## source tor\_other\_cz\_state tor\_other\_cz\_fips tor\_other\_cz\_name begin\_lat  
## 1 NA NA NA NA 32.65  
## 2 NA NA NA NA 31.92  
## 3 NA NA NA NA 30.53  
## 4 NA NA NA NA 28.30  
## 5 NA NA NA NA 34.43  
## 6 NA NA NA NA 31.18  
## begin\_lon end\_lat end\_lon state\_fips  
## 1 -90.03 NA NA 28  
## 2 -89.67 NA NA 28  
## 3 -86.50 NA NA 12  
## 4 -81.40 NA NA 12  
## 5 -119.82 NA NA 6  
## 6 -81.50 NA NA 13

## Step 8: There is data that comes with base R on U.S. states (data(“state”)). Use that to create a dataframe with these three columns: state name, area, and region

# Load the "state" dataset  
data("state")  
  
# Create a new DataFrame with state name, area, and region  
us\_state\_info<-data.frame(state=state.name, region=state.region, area=state.area)  
  
# Print the first few rows of the new DataFrame  
head(us\_state\_info)

## state region area  
## 1 Alabama South 51609  
## 2 Alaska West 589757  
## 3 Arizona West 113909  
## 4 Arkansas South 53104  
## 5 California West 158693  
## 6 Colorado West 104247

us\_state\_info$state <- toupper(us\_state\_info$state)

## Step 9: Create a dataframe with the number of events per state in the year of your birth. Merge in the state information dataframe you just created in step 8. Remove any states that are not in the state information dataframe.

# Create a frequency table for the specified column  
frequency\_table<-table(newname\_data$state)  
  
# Print the frequency table  
print(frequency\_table)

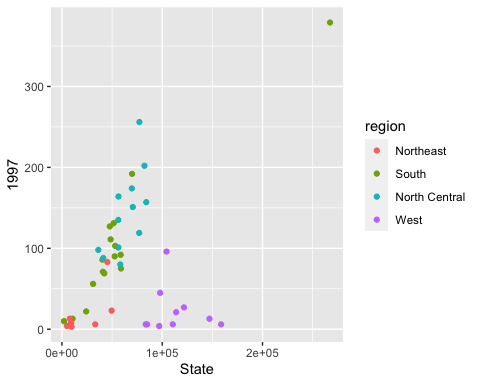
##   
## ALABAMA ARIZONA ARKANSAS CALIFORNIA COLORADO   
## 131 21 103 6 96   
## CONNECTICUT DELAWARE FLORIDA GEORGIA IDAHO   
## 4 10 92 75 6   
## ILLINOIS INDIANA IOWA KANSAS KENTUCKY   
## 164 98 101 202 86   
## LOUISIANA MAINE MARYLAND MASSACHUSETTS MICHIGAN   
## 111 6 13 6 80   
## MINNESOTA MISSISSIPPI MISSOURI MONTANA NEBRASKA   
## 157 127 174 13 256   
## NEVADA NEW HAMPSHIRE NEW JERSEY NEW MEXICO NEW YORK   
## 6 8 13 27 23   
## NORTH CAROLINA NORTH DAKOTA OHIO OKLAHOMA OREGON   
## 90 151 88 192 4   
## PENNSYLVANIA SOUTH CAROLINA SOUTH DAKOTA TENNESSEE TEXAS   
## 83 56 119 69 379   
## UTAH VERMONT VIRGINIA WEST VIRGINIA WISCONSIN   
## 6 3 71 22 135   
## WYOMING   
## 45

newset<-data.frame(table(newname\_data$state))  
  
newset1<-rename(newset, c("state"="Var1"))  
  
merged <- merge(x=newset1,y=us\_state\_info,by.x="state", by.y="state")  
  
head(merged)

## state Freq region area  
## 1 ALABAMA 131 South 51609  
## 2 ARIZONA 21 West 113909  
## 3 ARKANSAS 103 South 53104  
## 4 CALIFORNIA 6 West 158693  
## 5 COLORADO 96 West 104247  
## 6 CONNECTICUT 4 Northeast 5009

## Step 10: Create the following plot

library(ggplot2)  
  
stateyear\_plot <- ggplot(merged, aes(x=area, y=Freq))+  
 geom\_point(aes(color=region)) +  
 labs(x="State",  
 y="1997")  
  
show(stateyear\_plot)



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.