Storms Events: Assignment 3

Sowmya

2022-09-26

# Section 1

#Loading tidyverse package for loading the data  
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --  
## v ggplot2 3.3.6 v purrr 0.3.4   
## v tibble 3.1.8 v dplyr 1.0.10  
## v tidyr 1.2.1 v stringr 1.4.1   
## v readr 2.1.2 v forcats 0.5.2   
## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

#Loading Data  
StormEventsData97 <- read\_csv("StormEvents\_details-ftp\_v1.0\_d1997\_c20220425.csv.gz")

## Warning: One or more parsing issues, see `problems()` for details

## Rows: 41991 Columns: 51  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (20): STATE, MONTH\_NAME, EVENT\_TYPE, CZ\_TYPE, CZ\_NAME, WFO, BEGIN\_DATE\_T...  
## dbl (24): BEGIN\_YEARMONTH, BEGIN\_DAY, BEGIN\_TIME, END\_YEARMONTH, END\_DAY, EN...  
## lgl (7): MAGNITUDE\_TYPE, FLOOD\_CAUSE, CATEGORY, TOR\_OTHER\_WFO, TOR\_OTHER\_CZ...  
##   
## i Use `spec()` to retrieve the full column specification for this data.  
## i Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

#Getting Data overview  
head(StormEventsData97)

## # A tibble: 6 x 51  
## BEGIN\_~1 BEGIN~2 BEGIN~3 END\_Y~4 END\_DAY END\_T~5 EPISO~6 EVENT~7 STATE STATE~8  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 199704 21 1727 199704 21 1727 2402786 5592480 TENN~ 47  
## 2 199704 21 1730 199704 21 1730 2402786 5592481 TENN~ 47  
## 3 199704 21 1630 199704 21 1630 2402786 5592482 TENN~ 47  
## 4 199707 17 500 199707 17 500 2402790 5604831 KANS~ 20  
## 5 199707 16 2354 199707 16 2354 2402790 5604827 KANS~ 20  
## 6 199707 17 515 199707 17 515 2402790 5604828 KANS~ 20  
## # ... with 41 more variables: YEAR <dbl>, MONTH\_NAME <chr>, EVENT\_TYPE <chr>,  
## # CZ\_TYPE <chr>, CZ\_FIPS <dbl>, CZ\_NAME <chr>, WFO <chr>,  
## # BEGIN\_DATE\_TIME <chr>, CZ\_TIMEZONE <chr>, END\_DATE\_TIME <chr>,  
## # INJURIES\_DIRECT <dbl>, INJURIES\_INDIRECT <dbl>, DEATHS\_DIRECT <dbl>,  
## # DEATHS\_INDIRECT <dbl>, DAMAGE\_PROPERTY <chr>, DAMAGE\_CROPS <chr>,  
## # SOURCE <chr>, MAGNITUDE <dbl>, MAGNITUDE\_TYPE <lgl>, FLOOD\_CAUSE <lgl>,  
## # CATEGORY <lgl>, TOR\_F\_SCALE <chr>, TOR\_LENGTH <dbl>, TOR\_WIDTH <dbl>, ...

# Section 2

#Variable selection   
var\_selected <- c("BEGIN\_YEARMONTH", "BEGIN\_DAY", "BEGIN\_TIME", "BEGIN\_DATE\_TIME","END\_YEARMONTH", "END\_DAY", "END\_TIME",   
 "END\_DATE\_TIME", "EPISODE\_ID", "EVENT\_ID", "STATE", "STATE\_FIPS", "CZ\_NAME", "CZ\_TYPE", "CZ\_FIPS", "EVENT\_TYPE",   
 "SOURCE", "BEGIN\_LAT", "BEGIN\_LON", "END\_LAT", "END\_LON")  
#Sub-setting the data  
StormEventsData97 <- StormEventsData97[var\_selected]

# Section 3

#Sorting data using BEGIN\_YEARMONTH variable  
StormEventsData97 <- StormEventsData97 %>% arrange(BEGIN\_YEARMONTH)  
#Getting Data overview after sorting  
head(StormEventsData97)

## # A tibble: 6 x 21  
## BEGIN\_YEARMO~1 BEGIN~2 BEGIN~3 BEGIN~4 END\_Y~5 END\_DAY END\_T~6 END\_D~7 EPISO~8  
## <dbl> <dbl> <dbl> <chr> <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 199601 17 1700 17-JAN~ 199701 16 1200 16-JAN~ 2058066  
## 2 199601 9 1200 09-JAN~ 199701 1 0 01-JAN~ 2058061  
## 3 199601 9 1200 09-JAN~ 199701 1 0 01-JAN~ 2058061  
## 4 199601 9 1200 09-JAN~ 199701 1 0 01-JAN~ 2058061  
## 5 199601 9 1200 09-JAN~ 199701 1 0 01-JAN~ 2058061  
## 6 199601 9 1200 09-JAN~ 199701 1 0 01-JAN~ 2058061  
## # ... with 12 more variables: EVENT\_ID <dbl>, STATE <chr>, STATE\_FIPS <dbl>,  
## # CZ\_NAME <chr>, CZ\_TYPE <chr>, CZ\_FIPS <dbl>, EVENT\_TYPE <chr>,  
## # SOURCE <chr>, BEGIN\_LAT <dbl>, BEGIN\_LON <dbl>, END\_LAT <dbl>,  
## # END\_LON <dbl>, and abbreviated variable names 1: BEGIN\_YEARMONTH,  
## # 2: BEGIN\_DAY, 3: BEGIN\_TIME, 4: BEGIN\_DATE\_TIME, 5: END\_YEARMONTH,  
## # 6: END\_TIME, 7: END\_DATE\_TIME, 8: EPISODE\_ID

# Section 4

#Loading Stringr Package  
library(stringr)  
#Setting the State variable entries to title case  
StormEventsData97$STATE <- str\_to\_title(StormEventsData97$STATE)  
#Setting the county variable entries to title case  
StormEventsData97$CZ\_NAME <- str\_to\_title(StormEventsData97$CZ\_NAME)

# Section 5

#Retaining entries with CZ Type C and removing the CZ Type column  
StormEventsData97 <- StormEventsData97 %>% filter(CZ\_TYPE == "C")  
StormEventsData97 <- StormEventsData97[, -which(names(StormEventsData97) == "CZ\_TYPE")]

# Section 6

#State FIPS padding using 0   
StormEventsData97$STATE\_FIPS <- str\_pad(StormEventsData97$STATE\_FIPS, width = 3, side = "left", pad = "0")  
#Country FIPS padding using 0   
StormEventsData97$CZ\_FIPS <- str\_pad(StormEventsData97$CZ\_FIPS, width = 4, side = "left", pad = "0")  
#Forming one fips column  
StormEventsData97 <- unite(StormEventsData97, col = "fips", c("STATE\_FIPS", "CZ\_FIPS"), sep = "")

# Section 7

#Setting column names in the data to lower case  
StormEventsData97 <- rename\_all(StormEventsData97, tolower)

# Section 8

#Loading state data  
data("state")  
#Forming data frame from state data  
stateInfoData <- data.frame(state.name, state.area, state.region)

# Section 9

#Getting number of events per state  
state\_events <- data.frame(table(StormEventsData97$state))  
#Renaming State events data columns  
colnames(state\_events) <- c("state", "events")  
#Merging with state.data frame data set  
storm\_merge <- merge(x = stateInfoData, y = state\_events, by.x = "state.name", by.y = "state")

# Section 10

#Land area vs number of events plot  
storm\_merge %>% ggplot() + aes(x = state.area, y = events, colour = state.region) +   
 geom\_point() + scale\_color\_hue(direction = 1) +  
 labs(x = "Land area (square miles)", y = "# of strom events in 1997", color = "region") +  
 theme\_minimal()

