Assignment 3

Shaik

2022-09-27

# Loading Packages

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()

library(stringr)

# Part 1

#Reading the Data into R   
StormEvents\_1993 <- read\_csv("StormEvents\_details-ftp\_v1.0\_d1993\_c20220425.csv.gz")

## Rows: 8664 Columns: 51  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (17): STATE, MONTH\_NAME, EVENT\_TYPE, CZ\_TYPE, CZ\_NAME, BEGIN\_DATE\_TIME, ...  
## dbl (23): BEGIN\_YEARMONTH, BEGIN\_DAY, BEGIN\_TIME, END\_YEARMONTH, END\_DAY, EN...  
## lgl (11): EPISODE\_ID, WFO, SOURCE, MAGNITUDE\_TYPE, FLOOD\_CAUSE, CATEGORY, TO...  
##   
## 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.

# Part 2

#Selecting columns for the data analysis  
StormEvents\_1993 <- StormEvents\_1993 %>% select(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)

# Part 3

#Arranging by beginning year and month   
StormEvents\_1993 <- StormEvents\_1993 %>% arrange(BEGIN\_YEARMONTH)

# Part 4

#Changing the State and County Names to title case  
StormEvents\_1993$STATE <- str\_to\_title(StormEvents\_1993$STATE)  
StormEvents\_1993$CZ\_NAME <- str\_to\_title(StormEvents\_1993$CZ\_NAME)

# Part 5

#Filtering for CZ Type C  
StormEvents\_1993 <- StormEvents\_1993 %>% filter(CZ\_TYPE == "C")  
#Removing CZ Type column  
StormEvents\_1993 <- StormEvents\_1993 %>% select(-CZ\_TYPE)

# Part 6

#Padding the State and County FIPS with 0  
StormEvents\_1993$STATE\_FIPS <- str\_pad(StormEvents\_1993$STATE\_FIPS, width = 3, side = "left", pad = "0")  
StormEvents\_1993$CZ\_FIPS <- str\_pad(StormEvents\_1993$CZ\_FIPS, width = 4, side = "left", pad = "0")  
#Uniting the two columns to form the fips column  
StormEvents\_1993 <- unite(StormEvents\_1993, col = "fips", c("STATE\_FIPS", "CZ\_FIPS"), sep = "", remove = F)

#Part 7

#Setting column names to lower case  
StormEvents\_1993 <- rename\_all(StormEvents\_1993, tolower)

# Part 8

#Calling the state data  
data("state")  
#Creating data frame with state name, area and region  
state.dataframe <- data.frame(state = state.name,  
 state.area = state.area,   
 state.region = state.region)

# Part 9

#Grouping and Summarizing for state and number of events  
events.per.state <- StormEvents\_1993 %>%   
 group\_by(state) %>%   
 summarise(events.no = n())  
#Checking which state in event per state is not in state data frame  
which(events.per.state$state %in% state.dataframe$state)

## [1] 1 2 3 4 5 6 7 8 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26  
## [26] 27 28 29 30 31 32 33 34 35 36 37 38 40 41 42 43 44 45 46 47 48 49 50 51

#Excluding rows 9 and 39 in events per state, not present in state data information  
events.per.state <- events.per.state[c(which(events.per.state$state %in% state.dataframe$state)),]  
#Merging with state.data frame data set  
state.storm.dataframe <- merge(x = state.dataframe, y = events.per.state,   
 by.x = "state", by.y = "state")

# Part 10

#Plotting state area and number of events, grouped by region  
ggplot(state.storm.dataframe) +  
 aes(x = state.area, y = events.no, colour = state.region) +  
 geom\_point(shape = "circle", size = 1.5) +  
 scale\_color\_hue(direction = 1) +  
 labs(  
 x = "Land area (square miles)",  
 y = "# of strom events in 1993",  
 color = "region"  
 ) +  
 theme\_minimal()

