DIsaster_1982_2022.R

Mukesh Rajmohan

2023-10-16

```
# load libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)

# load the data from csv file
events <- read.csv('events-US-1982-2022.csv',as.is=FALSE,skip=1)
state <- read.csv('state-cost-data.csv',as.is=FALSE,skip=1)

# change the date formats and get the Year values
disaster <- select(events,-End.Date,-Deaths)
disaster <- rename(disaster,Date=Begin.Date,`Cost(In Million)`=Total.CPI.Adjusted.Cost..Millions.of.Doll
disaster <- mutate(disaster,formatted_date=as.Date(as.character(Date),format = "%Y%m%d")) |>
    mutate(disaster, Year=year(formatted_date))
disaster <- disaster[order(disaster$Year,-disaster$`Cost(In Million)`),]

# Filter the highest cost disaster each year
disaster_max <- disaster |>
    group_by(Year) |>
    filter(`Cost(In Million)` == max(`Cost(In Million)`))

# Filter to get the top 15 high costing disasters
disaster_high <- disaster |>
    arrange(desc(`Cost(In Million)`)) |>
    head(15)

# Convert Cost from Millions to Billions
disaster_max <- rename(disaster_max, Cost=`Cost(In Million)`)
disaster_max$Cost <- disaster_max$Cost / 1000
disaster_max$Cost <- round(disaster_max$Cost, 1)
```

```r
# Convert Cost from Millions to Billions
disaster_high <- rename(disaster_high, Cost=`Cost(In Million)`)
disaster_high$Cost <- disaster_high$Cost / 1000
disaster_high$Cost <- round(disaster_high$Cost, 1)

# pivot the states table to get disaster and costs
states_long <- state |>
    pivot_longer(cols=-state, names_to="disaster", values_to="cost")

# Filter the highest costing disaster in each state
states_max <- states_long |>
    group_by(state) |>
    filter(cost == max(cost))

# remove US data and converting cost to billions
states_max <- states_max[states_max$state != "US",]
states_max <- states_max[states_max$cost != "0",]
states_max$cost <- states_max$cost / 1000
states_max$cost <- round(states_max$cost, 1)

# write the new files into local folder
write.csv(disaster_max, file = "disaster.csv", row.names = FALSE)
write.csv(disaster_high, file = "disaster_top15.csv", row.names = FALSE)
write.csv(states_max, file = "states_disaster.csv", row.names = FALSE)
```