global.R

shaun

2021-12-12

```
#####################################
# Add all the required Libraries
#####################################

library(shiny) # Shiny Library
library(plotly) # Plotly graphing librarry
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(highcharter) # Highcharter Libarry
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(shinydashboard) # Shiny Dashboard LIbrary
```

```
##
## Attaching package: 'shinydashboard'
```

```
## The following object is masked from 'package:graphics':
##
##     box
```

```
library(data.tree) # For data tables
library(treemap)
library(leaflet) # For maps and Choropleth
library(stringr)
library(shinyWidgets) # Shiny Widgets
library(dplyr) # Data Manipulation Library
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(shinythemes) # Shiny App themes


#######################################
# Read the dataset
#######################################


states_dataset <-
  read.csv("data/injured1.csv")

state_group <- states_dataset%>%
  group_by(state_name)%>%
  summarize(
    state_code=state_code,
    persons_killed_2014=persons_killed_2014,
    persons_killed_2015=persons_killed_2015,
    persons_killed_2016=persons_killed_2016,
    persons_killed_2017=persons_killed_2017,
    persons_injured_2014=persons_injured_2014,
    persons_injured_2015=persons_injured_2015,
    persons_injured_2016=persons_injured_2016,
    persons_injured_2017=persons_injured_2017,
    weather_normal=weather_normal,
    weather_mist_fog=weather_mist_fog,
    weather_cloudy=weather_cloudy,
    weather_rain=weather_rain,
    weather_flooding=weather_flooding,
    weather_hail_sleet=weather_hail_sleet,
    weather_snow=weather_snow,
    weather_dust_storm=weather_dust_storm,
    weather_other_extreme_conditions=weather_other_extreme_conditions,
    road_surfaced_road_acc=road_surfaced_road_acc,
    road_metalled_road_acc=road_metalled_road_acc,
```

```
    road_normalpucca_road_acc=road_normalpucca_road_acc,
    road_Kutcha_road_acc=road_Kutcha_road_acc,
    road_dry_road_acc=road_dry_road_acc,
    road_wet_road_acc=road_wet_road_acc,
    road_goodsurface_road_acc=road_goodsurface_road_acc,
    road_loosesurface_road_acc=road_loosesurface_road_acc,
    road_under_repair_road_acc=road_under_repair_road_acc,
    road_corrugated_road_acc=road_corrugated_road_acc,
    road_slippery_road_acc=road_slippery_road_acc,
    road_snowy_road_acc=road_snowy_road_acc,
    road_muddy_road_acc=road_muddy_road_acc,
    road_oily_road_acc=road_oily_road_acc,
    road_straight_road_acc=road_straight_road_acc,
    road_slightcurve_road_acc=road_slightcurve_road_acc,
    road_flat_road_acc=road_flat_road_acc,
    road_gentleincline_road_acc=road_gentleincline_road_acc,
    road_hump_road_acc=road_hump_road_acc,
    road_dip_road_acc=road_dip_road_acc,
    road_pothole_road_acc=road_pothole_road_acc,
    road_speedbreaker_road_acc=road_speedbreaker_road_acc,
    road_steepincline_road_acc=road_steepincline_road_acc,
    road_sharpcurve_road_acc=road_sharpcurve_road_acc,
    road_earthernshoulderedgedrop_road_acc=road_earthernshoulderedgedrop_road_acc,
    road_other_road_acc=road_other_road_acc,
    vehicle_defect_brakes=vehicle_defect_brakes,
    vehicle_defect_steering=vehicle_defect_steering,
    vehicle_defect_puncturedbursttyres=vehicle_defect_puncturedbursttyres,
    vehicle_defect_baldtyres=vehicle_defect_baldtyres,
    vehicle_defect_wornouttyres=vehicle_defect_wornouttyres,
    vehicle_defect_othermechanical=vehicle_defect_othermechanical,
    lat=lat,
    lng=lng
  )


state_group$state_name <-
  as.character(state_group$state_name)
#####################################
# Arrange and group by state
#####################################

#####################################
# Download India Map GeoJson file
#####################################
mapdata <-
  get_data_from_map(download_map_data("countries/in/custom/in-all-andaman-and-nicobar"))

#####################################
# Correcting the data to match the data frames
#####################################


state_group$state_name <- as.factor(state_group$state_name)
```

```r
# Get the codes for all the states
hcmap.state_codes <-
  dplyr::select(filter(
    mapdata,
    tolower(mapdata$name) %in% tolower(state_group$state_name)
  ), c("hc-a2", "name"))
hcmap.state_codes$name <- toupper(hcmap.state_codes$name)
state_group$state_name <- toupper(state_group$state_name)
# Merge the codes with the cities dataset
states_dataset.merge <-
  merge(state_group,
        hcmap.state_codes,
        by.x = "state_name",
        by.y = "name")

states_dataset.merge$state_name <-
  as.factor(states_dataset.merge$state_name)

state_group$state_name <- as.factor(state_group$state_name)
state_group$lng <- as.numeric(state_group$lng)
state_group$lat <- as.numeric(state_group$lat)

spllitted_cities <- split(states_dataset, states_dataset$state_code)

by_state_order <-
  state_group[order(state_group$state_name), ]

state_group$state_name <- as.factor(state_group$state_name)
```