server.R

2021-12-12

```
#For Shiny integration
library(shiny)
library(knitr)
#Libraries for prediction model
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(caret)

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift

library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

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

```
library(owmr)
```

```
## owmr 0.8.2
##    another crazy way to talk to OpenWeatherMap's API
##    Documentation: type ?owmr or https://crazycapivara.github.io/owmr/
##    Issues, notes and bleeding edge: https://github.com/crazycapivara/owmr/

##
## Attaching package: 'owmr'

## The following object is masked from 'package:purrr':
##
##     flatten
```

```
#Prediction using Random Forest

key = "71bc0174c0bc055562e89f563918d2a1"
owmr_settings(key)
```

```
## It is recommended that you store your api key in an environment variable
called OWM_API_KEY.
```

```
data_pred = read.csv("data/final_severity.csv")
data_pred = data_pred[,2:22]
#View(data_pred)
colnames(data_pred)
```

```
##  [1] "State..UT"                      "state_code"
##  [3] "Defective.brakes"               "Defective.Steering"
##  [5] "Punctured.burst.Tyres"          "Bald.Tyres"
##  [7] "Worn.out.Tyres"                 "Other.serious.mechanical.defect"
##  [9] "normal"                         "mist_fog"
## [11] "cloudy"                         "rain"
## [13] "flooding"                       "hail_sleet"
## [15] "snow"                           "dust_storm"
## [17] "Surfaced.Roads"                 "Metalled.Roads"
## [19] "Pucca.road..Normal.Road."       "Kutcha.Roads"
## [21] "Dry.road"
```

```
choices_col <- c()
```

```
road <- read.csv(file = "data/injured1.csv")
```

```
# Define server logic required to draw state.filter histogram
shinyServer(function(input, output, session) {
  observe({
    updateCheckboxGroupInput(
      session,
```

```
      'state.checkbox.filter',
      choiceNames = as.character(by_state_order$state_name),
      choiceValues = by_state_order$state_code,
     selected = if (input$all.none)
        by_state_order$state_code
    )
  })
  checkbox_state_filter <- reactive({
    state_group %>%
      filter(state_code %in% input$state.checkbox.filter)

  })


  df3 <- reactive({
    state_group %>%
      filter(state_code %in% input$state.checkbox.filter)

  })




  df2 <- reactive({
    cities_dataset %>%
      filter(state_code %in% input$state.checkbox.filter)

  })




##############################################################################
###
  # TAB1 graphs and outputs- Start

##############################################################################
###

  # Leaflet Map Plot Generation
  output$tab1_leaflet_map <- renderLeaflet({

    # If All States option is selected in the dropdown
    if (input$tab1_dropdown_states == "All") {

      leaflet.map <- leaflet()  %>% addTiles()

      names(spllitted_cities) %>%
```

```
        purrr::walk(function(city.data.frame) {
          leaflet.map <<- leaflet.map %>%
            # Marker HTML layout to show on the popup
            addMarkers(
              data = splitted_cities[[city.data.frame]],
              lng =  ~ lng,
              lat =  ~ lat,
              popup = paste(
                "<h4>",
                splitted_cities[[city.data.frame]]$state_name,
                "</h4>",
                "<b>persons_killed_2014:</b>",
                splitted_cities[[city.data.frame]]$persons_killed_2014,
                "<br>",
                "<b>persons_killed_2015:</b>",
                splitted_cities[[city.data.frame]]$persons_killed_2015,
                "<br>",
                "<b>persons_killed_2016:</b>",
                splitted_cities[[city.data.frame]]$persons_killed_2016,
                "<br>",
                "<b>persons_killed_2017:</b>",
                splitted_cities[[city.data.frame]]$persons_killed_2017

              ),
              group = city.data.frame,
              clusterOptions =
markerClusterOptions(removeOutsideVisibleBounds = F),
              labelOptions = labelOptions(noHide = F,
                                          direction = 'auto')
            )
        })

      leaflet.map

    } else{
      states_dataset_filtered <-
        filter(states_dataset,
               states_dataset$state_code == input$tab1_dropdown_states)
      leaflet.map <- leaflet()  %>% addTiles() %>%
        addMarkers(
          data = states_dataset_filtered,
          lng =  ~ lng,
          lat =  ~ lat,
          popup = paste(
            "<h4>",
            states_dataset_filtered$state_name,
            "</h4>",
            "<b>persons_killed_2014:</b>",
            states_dataset_filtered$persons_killed_2014,
            "<br>",
```

```
            "<b>persons_killed_2015:</b>",
            states_dataset_filtered$persons_killed_2015,
            "<br>",
            "<b>persons_killed_2016:</b>",
            states_dataset_filtered$persons_killed_2016,
            "<br>",
            "<b>persons_killed_2017:</b>",
            states_dataset_filtered$persons_killed_2017

          )
        )

      leaflet.map

    }
  })

  # Total Cities Value Box output generation
  output$tab1_valuebox_persons_killed_2014 <- renderValueBox({
    if (input$tab1_dropdown_states == "All") {
      valueBox(
        sum(states_dataset$persons_killed_2014),
        "Total Persons Killed in 2014",
        color = "purple",
        width = 6
      )
    } else{
      states_dataset_filtered <- filter(state_group, state_group$state_code
== input$tab1_dropdown_states)
      valueBox(states_dataset_filtered$persons_killed_2014,
               "Total Persons Killed in 2014",
               color = "purple",
               width = 6)
    }
  })

  # Total Popuation Value Box output generation
  output$tab1_valuebox_persons_killed_2015 <- renderValueBox({
    if (input$tab1_dropdown_states == "All") {
      valueBox(
        sum(states_dataset$persons_killed_2015),
        "Total Persons Killed in 2015",
        color = "orange",
        width = 6
      )
    } else{
      states_dataset_filtered <- filter(state_group, state_group$state_code
== input$tab1_dropdown_states)
      valueBox(states_dataset_filtered$persons_killed_2015,
```

```
                "Total Persons Killed in 2015",
                color = "orange",
                width = 6)
    }
  })

  # Total Male Population Percent Value Box output generation
  output$tab1_valuebox_persons_killed_2016 <- renderValueBox({
    if (input$tab1_dropdown_states == "All") {
      valueBox(
        sum(states_dataset$persons_killed_2016),
        "Total Persons Killed in 2016",
        color = "green",
        width = 6
      )
    } else{
      states_dataset_filtered <- filter(state_group, state_group$state_code
== input$tab1_dropdown_states)
      valueBox(states_dataset_filtered$persons_killed_2016,
                "Total Persons Killed in 2016",
                color = "green",
                width = 6)
    }
  })

  output$tab1_valuebox_persons_killed_2017 <- renderValueBox({
    if (input$tab1_dropdown_states == "All") {
      valueBox(
        sum(states_dataset$persons_killed_2017),
        "Total Persons Killed in 2017",
        color = "blue",
        width = 6
      )
    } else{
      states_dataset_filtered <- filter(state_group, state_group$state_code
== input$tab1_dropdown_states)
      valueBox(states_dataset_filtered$persons_killed_2017,
                "Total Persons Killed in 2017",
                color = "blue",
                width = 6)
    }
  })



  # Polar Plot output generation using Highcharter Library
  output$tab1_polar_plot <- renderHighchart({
    if (input$tab1_dropdown_states == "All") {
      hc <- highchart() %>%
```

```
        hc_chart(polar = TRUE)  %>%
        hc_title(
          text = "<b>Number of Persons Injured in:</b>",align = "center")%>%
        hc_xAxis(
          categories = c(
            "2014",
            "2015",
            "2016",
            "2017"
          ),
          tickmarkPlacement = "on",
          lineWidth = 0
        ) %>%
        hc_yAxis(
          gridLineInterpolation = "polygon",
          lineWidth = 0,
          min = 0
        ) %>%
        hc_series(
          list(
            name =  "All States",
            data = c(
              sum(states_dataset$persons_injured_2014),
              sum(states_dataset$persons_injured_2015),
              sum(states_dataset$persons_injured_2016),
              sum(states_dataset$persons_injured_2017)
            ),
            pointPlacement = "on",
            colorByPoint = TRUE,
            type = "column",
            colors = c("#F00", "#0F0", "#00F", "#F0F")
          )
        )

      hc
    } else{
      states_dataset_filtered <- filter(state_group, state_group$state_code
== input$tab1_dropdown_states)
      hc <- highchart() %>%
        hc_chart(polar = TRUE)  %>%
        hc_title(
          text = "<b>Number of Persons Injured in:</b>",align = "center")%>%
        hc_xAxis(
          categories = c(
            "2014",
            "2015",
            "2016",
            "2017"
          ),
          tickmarkPlacement = "on",
```

```
          lineWidth = 0
        ) %>%
        hc_yAxis(
          gridLineInterpolation = "polygon",
          lineWidth = 0,
          min = 0
        ) %>%
        hc_series(
          list(
            name = states_dataset_filtered$state_name,
            data = c(
              states_dataset_filtered$persons_injured_2014,
              states_dataset_filtered$persons_injured_2015,
              states_dataset_filtered$persons_injured_2016,
              states_dataset_filtered$persons_injured_2017
            ),
            pointPlacement = "on",
            colorByPoint = TRUE,
            type = "column",
            colors = c("#F00", "#0F0", "#00F", "#F0F")
          )
        )
      hc
    }
  })


##############################################################################
###
  # TAB1 graphs and outputs- End

##############################################################################
###

##############################################################################
###
  # TAB2 graphs and outputs- Start
##############################################################################
###


##############################################################################
###
  # TAB2 graphs and outputs- End
##############################################################################
###


##############################################################################
```

```
###
  # TAB3 graphs and outputs- Start
################################################################################
###

  output$state_group <- DT::renderDataTable({
    action <- DT::dataTableAjax(session, state_group)
    DT::datatable(
      state_group,
      options = list(
        searching = T,
        pageLength = 15,
        scrollX = T
      ),
      escape = FALSE
    )
  })

################################################################################
###
  # TAB3 graphs and outputs- End
################################################################################
###

################################################################################
###
  # TAB4 graphs and outputs- Start

################################################################################
###
  output$tab4_column_chart1 <- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = persons_injured_2014, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>Persons injured in 2014</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
```

```
    #else{}
  })

  output$tab4_column_chart2 <- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
    hc <-
      hchart(state.filter,
             "column",
             hcaes(x = state_name, y = persons_injured_2015, color =
state_name))  %>%
      hc_add_theme(hc_theme_google()) %>%
      hc_tooltip(
        crosshairs = TRUE,
        backgroundColor = "#FCFFC5",
        shared = TRUE,
        borderWidth = 5,
        pointFormat = "<b>Persons injured in 2015</b>: {point.y}"
      ) %>%
      hc_xAxis(title = list(text = "States"))

    hc
    }
  })

  output$tab4_column_chart3 <- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
    hc <-
      hchart(state.filter,
             "column",
             hcaes(x = state_name, y = persons_injured_2016, color =
state_name))  %>%
      hc_add_theme(hc_theme_google()) %>%
      hc_tooltip(
        crosshairs = TRUE,
        backgroundColor = "#FCFFC5",
        shared = TRUE,
        borderWidth = 5,
        pointFormat = "<b>Persons injured in 2016</b>: {point.y}"
      ) %>%
      hc_xAxis(title = list(text = "States"))

    hc
    }
  })

  output$tab4_column_chart4 <- renderHighchart({
    state.filter <- checkbox_state_filter()
```

```
    if(length(state.filter$state_name)>0){
    hc <-
      hchart(state.filter,
             "column",
             hcaes(x = state_name, y = persons_injured_2017, color =
state_name))  %>%
      hc_add_theme(hc_theme_google()) %>%
      hc_tooltip(
        crosshairs = TRUE,
        backgroundColor = "#FCFFC5",
        shared = TRUE,
        borderWidth = 5,
        pointFormat = "<b>Persons injured in 2017</b>: {point.y}"
      ) %>%
      hc_xAxis(title = list(text = "States"))

    hc
    }
  })

  output$tab4_column_chart5 <- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = persons_killed_2014, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>Persons killed in 2014</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })



  output$tab4_column_chart6 <- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
```

```
              "column",
              hcaes(x = state_name, y = persons_killed_2015, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>Persons killed in 2015</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })

  output$tab4_column_chart7 <- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = persons_killed_2016, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>Persons killed in 2016</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })

  output$tab4_column_chart8 <- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = persons_killed_2017, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
```

```
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>Persons killed in 2017</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })

  output$tab4_column_chart21 <- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
    hc <-
      hchart(state.filter,
             "column",
             hcaes(x = state_name, y = weather_normal, color = state_name))
%>%
      hc_add_theme(hc_theme_google()) %>%
      hc_tooltip(
        crosshairs = TRUE,
        backgroundColor = "#FCFFC5",
        shared = TRUE,
        borderWidth = 5,
        pointFormat = "<b>Weather_condition_fine</b>: {point.y}"
      ) %>%
      hc_xAxis(title = list(text = "States"))

    hc
    }
  })



  output$tab4_column_chart22<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
    hc <-
      hchart(state.filter,
             "column",
             hcaes(x = state_name, y = weather_mist_fog, color = state_name))
%>%
      hc_add_theme(hc_theme_google()) %>%
      hc_tooltip(
        crosshairs = TRUE,
        backgroundColor = "#FCFFC5",
        shared = TRUE,
```

```
        borderWidth = 5,
        pointFormat = "<b>Weather_condition_Mist_Foggy</b>: {point.y}"
      ) %>%
      hc_xAxis(title = list(text = "States"))

    hc
    }
  })


  output$tab4_column_chart23<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
    hc <-
      hchart(state.filter,
             "column",
             hcaes(x = state_name, y = weather_cloudy, color = state_name))
%>%
      hc_add_theme(hc_theme_google()) %>%
      hc_tooltip(
        crosshairs = TRUE,
        backgroundColor = "#FCFFC5",
        shared = TRUE,
        borderWidth = 5,
        pointFormat = "<b>Weather_condition_cloudy</b>: {point.y}"
      ) %>%
      hc_xAxis(title = list(text = "States"))

    hc
    }
  })

   output$tab4_column_chart24<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = weather_rain, color = state_name))
%>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>Weather_condition_Rainy</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))
```

```
      hc
    }
  })

  output$tab4_column_chart25<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = weather_flooding, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>Weather_condition_flooding</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })

  output$tab4_column_chart26<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = weather_hail_sleet, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>Weather_condition_Hail/Sleet</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
```

```
  output$tab4_column_chart27<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = weather_snow, color = state_name))
%>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>Weather_condition_snow</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })

  output$tab4_column_chart28<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = weather_dust_storm, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>Weather_condition_dust_storm</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })

  output$tab4_column_chart29<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
    hc <-
      hchart(state.filter,
```

```
             "column",
             hcaes(x = state_name, y = weather_other_extreme_conditions,
color = state_name))  %>%
      hc_add_theme(hc_theme_google()) %>%
      hc_tooltip(
        crosshairs = TRUE,
        backgroundColor = "#FCFFC5",
        shared = TRUE,
        borderWidth = 5,
        pointFormat = "<b>weather_other_extreme_conditions</b>: {point.y}"
      ) %>%
      hc_xAxis(title = list(text = "States"))

    hc
    }
  })

  output$tab4_column_chart31<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_surfaced_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_surfaced_road</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })

  output$tab4_column_chart32<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_metalled_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
```

```
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_metalled_road</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart33<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_normalpucca_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_normalpucca_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart34<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_Kutcha_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_Kutcha_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))
```

```
      hc
    }
  })
  output$tab4_column_chart35<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_dry_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_dry_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart36<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_wet_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_wet_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart37<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
```

```
              "column",
              hcaes(x = state_name, y = road_goodsurface_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_goodsurface_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart38<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
              "column",
              hcaes(x = state_name, y = road_loosesurface_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_loosesurface_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart39<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
              "column",
              hcaes(x = state_name, y = road_under_repair_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
```

```
          borderWidth = 5,
          pointFormat = "<b>road_under_repair_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart40<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_corrugated_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_corrugated_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart41<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_slippery_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_slippery_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
```

```
  output$tab4_column_chart42<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_snowy_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_snowy_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart43<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_muddy_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_muddy_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart44<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_oily_road_acc, color =
state_name))  %>%
```

```
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_oily_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart45<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_straight_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_straight_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart46<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_slightcurve_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_slightcurve_road_acc</b>: {point.y}"
        ) %>%
```

```
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart47<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_flat_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_flat_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart48<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_gentleincline_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_gentleincline_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart49<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
```

```
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_hump_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_hump_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart50<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_dip_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_dip_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart51<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_pothole_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
```

```
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_pothole_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart52<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_speedbreaker_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_speedbreaker_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart53<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_steepincline_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_steepincline_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
```

```
    }
  })
  output$tab4_column_chart54<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = road_sharpcurve_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_sharpcurve_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart55<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y =
road_earthernshoulderedgedrop_road_acc, color = state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_earthernshoulderedgedrop_road_acc</b>:
{point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart56<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
```

```
               "column",
               hcaes(x = state_name, y = road_other_road_acc, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>road_other_road_acc</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart61<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = vehicle_defect_brakes, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>vehicle_defect_brakes</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart62<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = vehicle_defect_steering, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
```

```
          borderWidth = 5,
          pointFormat = "<b>vehicle_defect_steering</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart63<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = vehicle_defect_puncturedbursttyres,
color = state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>vehicle_defect_puncturedbursttyres</b>:
{point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })
  output$tab4_column_chart64<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = vehicle_defect_baldtyres, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>vehicle_defect_baldtyres</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
```

```
  })
  output$tab4_column_chart65<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = vehicle_defect_wornouttyres, color =
state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>vehicle_defect_wornouttyres</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })

  output$tab4_column_chart66<- renderHighchart({
    state.filter <- checkbox_state_filter()
    if(length(state.filter$state_name)>0){
      hc <-
        hchart(state.filter,
               "column",
               hcaes(x = state_name, y = vehicle_defect_othermechanical,
color = state_name))  %>%
        hc_add_theme(hc_theme_google()) %>%
        hc_tooltip(
          crosshairs = TRUE,
          backgroundColor = "#FCFFC5",
          shared = TRUE,
          borderWidth = 5,
          pointFormat = "<b>vehicle_defect_othermechanical</b>: {point.y}"
        ) %>%
        hc_xAxis(title = list(text = "States"))

      hc
    }
  })



  output$tab4_heatmap_chart1 <- renderHighchart({
    state.filter <- df3()
```

```
  if(length(state.filter$state_name)>0){
    hchart(state.filter,
           type = "treemap",
           hcaes(
             x = state_name,
             value = persons_killed_2014,
             color = persons_killed_2014
           ))  %>% hc_add_theme(hc_theme_538()) %>%
      hc_colorAxis(minColor = "#FF0000", maxColor = "#008000")
  }
})

output$tab4_heatmap_chart2 <- renderHighchart({
  state.filter <- df3()
  if(length(state.filter$state_name)>0){
    hchart(state.filter,
           type = "treemap",
           hcaes(
             x = state_name,
             value = persons_killed_2015,
             color = persons_killed_2015
           ))  %>% hc_add_theme(hc_theme_538()) %>%
      hc_colorAxis(minColor = "#FFFF00", maxColor = "#FF0000")
  }
})

output$tab4_heatmap_chart3 <- renderHighchart({
  state.filter <- df3()
  if(length(state.filter$state_name)>0){
    hchart(state.filter,
           type = "treemap",
           hcaes(
             x = state_name,
             value = persons_killed_2016,
             color = persons_killed_2016
           ))  %>% hc_add_theme(hc_theme_538()) %>%
      hc_colorAxis(minColor = "#0000FF", maxColor = "#008000")
  }
})

output$tab4_heatmap_chart4 <- renderHighchart({
  state.filter <- df3()
  if(length(state.filter$state_name)>0){
    hchart(state.filter,
           type = "treemap",
           hcaes(
             x = state_name,
             value = persons_killed_2017,
             color = persons_killed_2017
           ))  %>% hc_add_theme(hc_theme_538()) %>%
```

```
      hc_colorAxis(minColor = "#FFFF00", maxColor = "#008000")
  }
})

output$tab4_heatmap_chart5 <- renderHighchart({
  state.filter <- df3()
  if(length(state.filter$state_name)>0){
    hchart(state.filter,
           type = "treemap",
           hcaes(
             x = state_name,
             value = persons_injured_2014,
             color = persons_injured_2014
           ))  %>% hc_add_theme(hc_theme_538()) %>%
      hc_colorAxis(minColor = "#FF0000", maxColor = "#008000")
  }
})

output$tab4_heatmap_chart6 <- renderHighchart({
  state.filter <- df3()
  if(length(state.filter$state_name)>0){
    hchart(state.filter,
           type = "treemap",
           hcaes(
             x = state_name,
             value = persons_injured_2015,
             color = persons_injured_2015
           ))  %>% hc_add_theme(hc_theme_538()) %>%
      hc_colorAxis(minColor = "#FFFF00", maxColor = "#FF0000")
  }
})

output$tab4_heatmap_chart7 <- renderHighchart({
  state.filter <- df3()
  if(length(state.filter$state_name)>0){
    hchart(state.filter,
           type = "treemap",
           hcaes(
             x = state_name,
             value = persons_injured_2016,
             color = persons_injured_2016
           ))  %>% hc_add_theme(hc_theme_538()) %>%
      hc_colorAxis(minColor = "#0000FF", maxColor = "#008000")
  }
})

output$tab4_heatmap_chart8 <- renderHighchart({
  state.filter <- df3()
  if(length(state.filter$state_name)>0){
```

```
    hchart(state.filter,
           type = "treemap",
           hcaes(
             x = state_name,
             value = persons_injured_2017,
             color = persons_injured_2017
           ))  %>% hc_add_theme(hc_theme_538()) %>%
      hc_colorAxis(minColor = "#FFFF00", maxColor = "#008000")
  }
})



output$tab4_line_chart1 <- renderHighchart({
  state.filter <- checkbox_state_filter()
  state.filter <- state.filter[order(state.filter$state_name), ]
  highchart() %>%
    hc_xAxis(categories = state.filter$state_name) %>%
    hc_add_series(name = "persons_injured_2014",
                  data = state.filter$persons_injured_2014,
                  color = "green") %>%
    hc_add_series(name = "persons_injured_2015",
                  data = state.filter$persons_injured_2015,
                  color = "red") %>%
    hc_add_series(name = "persons_injured_2016",
                  data = state.filter$persons_injured_2016,
                  color = "blue") %>%
    hc_add_series(name = "persons_injured_2017",
                  data = state.filter$persons_injured_2017,
                  color = "blue")

})

output$tab4_line_chart2 <- renderHighchart({
  state.filter <- checkbox_state_filter()
  state.filter <- state.filter[order(state.filter$state_name), ]
  highchart() %>%
    hc_xAxis(categories = state.filter$state_name) %>%
    hc_add_series(name = "persons_killed_2014",
                  data = state.filter$persons_killed_2014,
                  color = "green") %>%
    hc_add_series(name = "persons_killed_2015",
                  data = state.filter$persons_killed_2015,
                  color = "red") %>%
    hc_add_series(name = "persons_killed_2016",
                  data = state.filter$persons_killed_2016,
                  color = "blue") %>%
    hc_add_series(name = "persons_killed_2017",
                  data = state.filter$persons_killed_2017,
```

```
                    color = "blue")

})


output$tab4_line_chart3 <- renderHighchart({
  state.filter <- checkbox_state_filter()
  state.filter <- state.filter[order(state.filter$state_name), ]
  highchart()  %>%
    hc_xAxis(categories = state.filter$state_name) %>%
    hc_add_series(name = "weather_normal",
                  data = state.filter$weather_normal,
                  color = "green") %>%
    hc_add_series(name = "weather_mist_fog",
                  data = state.filter$weather_mist_fog,
                  color = "red") %>%
    hc_add_series(name = "weather_cloudy",
                  data = state.filter$weather_cloudy,
                  color = "blue") %>%
    hc_add_series(name = "weather_rain",
                  data = state.filter$weather_rain,
                  color = "yellow")
})

output$tab4_line_chart4 <- renderHighchart({
  state.filter <- checkbox_state_filter()
  state.filter <- state.filter[order(state.filter$state_name), ]
  highchart()  %>%
    hc_xAxis(categories = state.filter$state_name) %>%
    hc_add_series(name = "weather_flooding",
                  data = state.filter$weather_flooding,
                  color = "green") %>%
    hc_add_series(name = "weather_hail_sleet",
                  data = state.filter$weather_hail_sleet,
                  color = "red") %>%
    hc_add_series(name = "weather_snow",
                  data = state.filter$weather_snow,
                  color = "blue") %>%
    hc_add_series(name = "weather_dust_storm",
                  data = state.filter$weather_dust_storm,
                  color = "yellow") %>%
    hc_add_series(name = "weather_other_extreme_conditions",
                  data = state.filter$weather_other_extreme_conditions,
                  color = "black")
})

output$tab4_line_chart5 <- renderHighchart({
  state.filter <- checkbox_state_filter()
  state.filter <- state.filter[order(state.filter$state_name), ]
```

```
  highchart()  %>%
    hc_xAxis(categories = state.filter$state_name) %>%
    hc_add_series(name = "road_surfaced_road_acc",
                  data = state.filter$road_surfaced_road_acc,
                  color = "green") %>%
    hc_add_series(name = "road_metalled_road_acc",
                  data = state.filter$road_metalled_road_acc,
                  color = "red") %>%
    hc_add_series(name = "road_normalpucca_road_acc",
                  data = state.filter$road_normalpucca_road_acc,
                  color = "blue") %>%
    hc_add_series(name = "road_Kutcha_road_acc",
                  data = state.filter$road_Kutcha_road_acc,
                  color = "yellow") %>%
    hc_add_series(name = "road_dry_road_acc",
                  data = state.filter$road_dry_road_acc,
                  color = "black")
})

output$tab4_line_chart6 <- renderHighchart({
  state.filter <- checkbox_state_filter()
  state.filter <- state.filter[order(state.filter$state_name), ]
  highchart()  %>%
    hc_xAxis(categories = state.filter$state_name) %>%
    hc_add_series(name = "road_wet_road_acc",
                  data = state.filter$road_wet_road_acc,
                  color = "green") %>%
    hc_add_series(name = "road_goodsurface_road_acc",
                  data = state.filter$road_goodsurface_road_acc,
                  color = "red") %>%
    hc_add_series(name = "road_loosesurface_road_acc",
                  data = state.filter$road_loosesurface_road_acc,
                  color = "blue") %>%
    hc_add_series(name = "road_under_repair_road_acc",
                  data = state.filter$road_under_repair_road_acc,
                  color = "yellow") %>%
    hc_add_series(name = "road_corrugated_road_acc",
                  data = state.filter$road_corrugated_road_acc,
                  color = "black")
})

output$tab4_line_chart7 <- renderHighchart({
  state.filter <- checkbox_state_filter()
  state.filter <- state.filter[order(state.filter$state_name), ]
  highchart()  %>%
    hc_xAxis(categories = state.filter$state_name) %>%
    hc_add_series(name = "road_slippery_road_acc",
                  data = state.filter$road_slippery_road_acc,
                  color = "green") %>%
    hc_add_series(name = "road_snowy_road_acc",
```

```
                    data = state.filter$road_snowy_road_acc,
                    color = "red") %>%
    hc_add_series(name = "road_muddy_road_acc",
                    data = state.filter$road_muddy_road_acc,
                    color = "blue") %>%
    hc_add_series(name = "road_oily_road_acc",
                    data = state.filter$road_oily_road_acc,
                    color = "yellow") %>%
    hc_add_series(name = "road_straight_road_acc",
                    data = state.filter$road_straight_road_acc,
                    color = "black")
})

output$tab4_line_chart8 <- renderHighchart({
  state.filter <- checkbox_state_filter()
  state.filter <- state.filter[order(state.filter$state_name), ]
  highchart()  %>%
    hc_xAxis(categories = state.filter$state_name) %>%
    hc_add_series(name = "road_slightcurve_road_acc",
                    data = state.filter$road_slightcurve_road_acc,
                    color = "green") %>%
    hc_add_series(name = "road_flat_road_acc",
                    data = state.filter$road_flat_road_acc,
                    color = "red") %>%
    hc_add_series(name = "road_gentleincline_road_acc",
                    data = state.filter$road_gentleincline_road_acc,
                    color = "blue") %>%
    hc_add_series(name = "road_hump_road_acc",
                    data = state.filter$road_hump_road_acc,
                    color = "yellow") %>%
    hc_add_series(name = "road_dip_road_acc",
                    data = state.filter$road_dip_road_acc,
                    color = "black")
})


output$tab4_line_chart9 <- renderHighchart({
  state.filter <- checkbox_state_filter()
  state.filter <- state.filter[order(state.filter$state_name), ]
  highchart()  %>%
    hc_xAxis(categories = state.filter$state_name) %>%
    hc_add_series(name = "road_pothole_road_acc",
                    data = state.filter$road_pothole_road_acc,
                    color = "green") %>%
    hc_add_series(name = "road_speedbreaker_road_acc",
                    data = state.filter$road_speedbreaker_road_acc,
                    color = "red") %>%
    hc_add_series(name = "road_steepincline_road_acc",
                    data = state.filter$road_steepincline_road_acc,
                    color = "blue") %>%
```

```
      hc_add_series(name = "road_sharpcurve_road_acc",
                    data = state.filter$road_sharpcurve_road_acc,
                    color = "yellow") %>%
      hc_add_series(name = "road_earthernshoulderedgedrop_road_acc",
                    data =
state.filter$road_earthernshoulderedgedrop_road_acc,
                    color = "black") %>%
      hc_add_series(name = "road_other_road_acc",
                    data = state.filter$road_other_road_acc,
                    color = "black")
  })



  output$tab4_line_chart10 <- renderHighchart({
    state.filter <- checkbox_state_filter()
    state.filter <- state.filter[order(state.filter$state_name), ]
    highchart()  %>%
      hc_xAxis(categories = state.filter$state_name) %>%
      hc_add_series(name = "vehicle_defect_brakes",
                    data = state.filter$vehicle_defect_brakes,
                    color = "green") %>%
      hc_add_series(name = "vehicle_defect_steering",
                    data = state.filter$vehicle_defect_steering,
                    color = "red") %>%
      hc_add_series(name = "vehicle_defect_puncturedbursttyres",
                    data = state.filter$vehicle_defect_puncturedbursttyres,
                    color = "blue")
  })


  output$tab4_line_chart11 <- renderHighchart({
    state.filter <- checkbox_state_filter()
    state.filter <- state.filter[order(state.filter$state_name), ]
    highchart()  %>%
      hc_xAxis(categories = state.filter$state_name) %>%
      hc_add_series(name = "vehicle_defect_baldtyres",
                    data = state.filter$vehicle_defect_baldtyres,
                    color = "green") %>%
      hc_add_series(name = "vehicle_defect_wornouttyres",
                    data = state.filter$vehicle_defect_wornouttyres,
                    color = "red") %>%
      hc_add_series(name = "vehicle_defect_othermechanical",
                    data = state.filter$vehicle_defect_othermechanical,
                    color = "blue")
  })

##############################################################################
###
```

```
  # TAB4 graphs and outputs- End

################################################################################
###

  output$value <- renderPrint({ input$select })



  output$mymap <- renderLeaflet({


    if(as.numeric(input$state)==1){
      df <- road

    }

    if(input$selectMap == 'p10')
    {
      y <- df$persons_killed_2014
      content <- paste("<b>",df$state_name,"</b></br>","<b>Persons Killed in
Road Accidents: 2014</b>",y)
      radius <- sqrt(df$persons_killed_2014)*800
      pal <- colorNumeric(
        palette = "Dark2",
        domain = df$persons_killed_2014
      )
      stat <- "Persons Killed in Road Accidents: 2014"
    }
    else if(input$selectMap == 'p11')
    {       y <- df$persons_killed_2015
    content <- paste("<b>",df$state_name,"</b></br>","<b>Persons Killed in
Road Accidents: 2015</b>",y)
    radius <- sqrt(df$persons_killed_2015)*800
    pal <- colorNumeric(
      palette = "Dark2",
      domain = df$persons_killed_2015
    )
    stat <- "Persons Killed in Road Accidents: 2015"
    }
    else if(input$selectMap == 'p12')
    {       y <- df$persons_killed_2016
    content <- paste("<b>",df$state_name,"</b></br>","<b>Persons Killed in
Road Accidents: 2016</b>",y)
    radius <- sqrt(df$persons_killed_2016)*800
    pal <- colorNumeric(
      palette = "Dark2",
      domain = df$persons_killed_2015
    )
```

```
    stat <- "Persons Killed in Road Accidents: 2016"
    }
    else if(input$selectMap == 'p13')
    {       y <- df$persons_killed_2017
    content <- paste("<b>",df$state_name,"</b></br>","<b>Persons Killed in
Road Accidents: 2017</b>",y)
    radius <- sqrt(df$persons_killed_2017)*800
    pal <- colorNumeric(
      palette = "Dark2",
      domain = df$persons_killed_2017
    )
    stat <- "Persons Killed in Road Accidents: 2017"
    }
    else if(input$selectMap == 'p14')
    {       y <- df$persons_injured_2014
    content <- paste("<b>",df$state_name,"</b></br>","<b>Persons Injured in
Road Accidents: 2014</b>",y)
    radius <- sqrt(df$persons_injured_2014)*800
    pal <- colorNumeric(
      palette = "Dark2",
      domain = df$persons_injured_2014
    )
    stat <- "Persons Injured in Road Accidents: 2014"
    }
    else if(input$selectMap == 'p15')
    {       y <- df$persons_injured_2015
    content <- paste("<b>",df$state_name,"</b></br>","<b>Persons Injured in
Road Accidents: 2015</b>",y)
    radius <- sqrt(df$persons_injured_2015)*800
    pal <- colorNumeric(
      palette = "Dark2",
      domain = df$persons_injured_2015
    )
    stat <- "Persons Injured in Road Accidents: 2015"
    }
    else if(input$selectMap == 'p16')
    {       y <- df$persons_injured_2016
    content <- paste("<b>",df$state_name,"</b></br>","<b>Persons Injured in
Road Accidents: 2016</b>",y)
    radius <- sqrt(df$persons_injured_2016)*800
    pal <- colorNumeric(
      palette = "Dark2",
      domain = df$persons_injured_2016
    )
    stat <- "Persons Injured in Road Accidents: 2016"
    }
    else if(input$selectMap == 'p17')
    {       y <- df$persons_injured_2017
    content <- paste("<b>",df$state_name,"</b></br>","<b>Persons Injured in
Road Accidents: 2017</b>",y)
```

```
    radius <- sqrt(df$persons_injured_2017)*800
    pal <- colorNumeric(
      palette = "Dark2",
      domain = df$persons_injured_2017
    )
    stat <- "Persons Injured in Road Accidents: 2017"
    }




    m <- leaflet(data =df ,height = 400) %>%setView(lng= 78.0419,lat =
17.1750,zoom=5)%>%
      addTiles()%>%
      addCircles(~lng,~lat,weight =1,radius =
radius,color=pal(y),popup=content)%>%
      addLegend(pal =pal,values = y,title =stat , opacity
=1,position="bottomright")


    m
  })


  ##########################################
  #    TAB 5- Statewise distribution Map  #
  ##########################################

  output$tab2_bubble_map <- renderHighchart({
    if (input$map.type.filter == "all_view") {
      hc <-
        hcmap(
          "countries/in/custom/in-all-andaman-and-nicobar",
          data = states_dataset.merge,
          value = input$attribute.filters,
          joinBy = c("hc-a2", "hc-a2"),
          name = input$attribute.filters,
          dataLabels = list(enabled = TRUE, format = "{point.name}"),
          borderColor = "#FAFAFA",
          borderWidth = 0.1,
          tooltip = list(valueDecimals = 2)
        ) %>% hc_colorAxis(minColor = "blue",
                           maxColor = "red",
                           stops = color_stops(n = 5))
%>%hc_mapNavigation(enabled = TRUE)

      hc
```

```
  }
  else{
    if (input$attribute.filters == "persons_injured_2014") {
      top.states<- state_group %>% arrange(desc(persons_injured_2014))
      top.states.20<- top.states[1:20, ]
      sel_attr <- top.states.20$persons_injured_2014
    }else{
      top.states <- state_group  %>% mutate(desc(persons_injured_2014))
      top.states.20 <- top.states[1:20, ]
      sel_attr <- top.states.20$persons_injured_2014
    }


    cities_20 <- data_frame(
      name= top.states.20$state_name,
      lat = top.states.20$lat,
      lng = top.states.20$lng,
      z = sel_attr,
      color = colorize(z)
    )

    hcmap(
      "countries/in/custom/in-all-andaman-and-nicobar",
      showInLegend = FALSE,
      borderColor = "black",
      borderWidth = 1
    ) %>%
      hc_add_series(
        data = cities_20,
        type = "mapbubble",
        name = "States",
        maxSize = '10%',
        dataLabels = list(enabled = TRUE, format = '{point.name}'),
        showInLegend = FALSE
      )
  }
})

########################################
#       TAB5 State-wise Map - END      #
########################################


########################################
#       TAB6 PREDICTION - Start        #
########################################
```

```
  df_choice <- reactive({
    state <- input$tab5_dropdown_states
    choice <- input$choice.checkbox.filter
    if(!is_null(choice)) {
      choices_col <- append(choices_col, choice)
    }
    set.seed(635)
    train_split <- createDataPartition(data_pred$Defective.brakes, p = 0.80,
list = FALSE)
    train_data <- data_pred[train_split,]
    test_data <- data_pred[-train_split,]

    randomForest_model <- randomForest(Defective.brakes~., data=train_data,
mtry=sqrt(12))

    Control_randomforest <- trainControl(method="repeatedcv", number=10,
repeats = 10)

    fd <- subset(data_pred, state_code == state)
    state_name <- fd$State..UT

    weather = (res <- get_current(state_name, units = "metric") %>%
owmr_as_tibble())
    weather_id <- weather$weather_id
    if(weather_id>=200 & weather_id<=232)
    {
      weather_severity = 7;
    }
    else if(weather_id>=300 & weather_id<=321)
    {
      weather_severity = 3;
    }
    else if(weather_id>=500 & weather_id<=531)
    {
      weather_severity = 4;
    }
    else if(weather_id>=600 & weather_id<=622)
    {
      weather_severity = 5;
    }
    else if(weather_id==731)
    {
      weather_severity = 2;
    }
    else if(weather_id==741)
    {
      weather_severity = 5;
```

```
    }
    else if(weather_id>=800 & weather_id<=804)
    {
      weather_severity = 1;
    }
    else if(weather_id>=700 & weather_id<800)
    {
      weather_severity = 3;
    }
    input_choices <- c("Defective.brakes", "Defective.Steering",
"Punctured.burst.Tyres","Bald.Tyres", "Other.serious.mechanical.defect",
"Metalled.Roads", "Pucca.road..Normal.Road.", "Kutcha.Roads",
"Loose.Surface", "Road.under.repair.construction", "Corrugated.Wavy.road",
"Snowy", "Muddy", "Slight.Curve", "Flat.Road", "Gentle.Incline", "Pot.Holes",
"Speed.Breaker", "Steep.Incline", "Sharp.Curve", "Others.road.conditions")

    check_list = vector("logical", 21)
    for(i in choices_col) {
      index <- which(input_choices == i)
      check_list[index] = TRUE;
    }
    state_data = subset(data_pred, state_code == state)
    total = weather_severity
    count = 1
    for(i in seq(3:21))
    {
      if(check_list[i]==TRUE)
      {
        total= total + state_data[,i]
        count= count+1
      }
    }
    final_val = ceiling(total/count)
    ans <- NULL
    if(final_val == 1) {
      ans <- "Very low risk"
    }
    else if(final_val == 2) {
      ans<- "Low risk"
    }
    else if(final_val == 3) {
      ans<- "Low to Moderate risk"
    }
    else if(final_val == 4) {
      ans<- "Moderate risk"
    }
    else if(final_val == 5) {
      ans<- "Moderate to High risk"
    }
    else if(final_val == 6) {
```

```
      ans<- "High risk"
    }
    else if(final_val == 7) {
      ans<- "Very high risk"
    }
    ans
  })

  output$prediction <- renderText({
    df_choice()
  })

  ########################################
  # TAB6 PREDICTION - END                #
  ########################################

})
```