### app.R

### azabache

2019-12-05

```
# 1) Please create a a shiny app to showcase:
# * Create a user interface with fluidPage, titlePanel and sidebarLayout
# * Create an HTML element with one of Shiny's tag functions
# * Add images by placing your image in a folder labeled www within your
  Shiny app directory and then calling the img function
# * Create a widget and connect widget to reactive output, an object that
  updates itself whenever your user changes the widget.
# For part 1 my widget is a fileinput element. I use fluidPage, titlePanel
# and sidebarLayout and my shiny tag is a tags$h5 element.
# I added an image ("cutecat.jpg") to my side bar.
# 2) Use a time series data to numerically and graphically showcase the trend
    component, random component and seasonal component (if it exists).
# After uploading my dummy time series file, I display a decomposed plot to
# showcase the trend, random, and seasonal components.
library(shiny)
library(forecast)
## Registered S3 method overwritten by 'xts':
##
    method
              from
##
    as.zoo.xts zoo
## Registered S3 method overwritten by 'quantmod':
##
    method
##
    as.zoo.data.frame zoo
## Registered S3 methods overwritten by 'forecast':
    method
                     from
##
    fitted.fracdiff
                     fracdiff
    residuals.fracdiff fracdiff
ui <- fluidPage(
   tags$h5("Author: Erika Azabache"),
   titlePanel("Simple Time Series App"),
   tabPanel("Data Summary",
           sidebarLayout(
               sidebarPanel(
                   fileInput('dummyfile', h4(strong('Choose csv File')),
                                    accept=c('text/csv',
                                             '.csv')),
```

```
img(src='cutecat.jpg', align = "center", height = '300px', width = '200px')
                 ),
                 mainPanel(fluidRow(
                     column(10,h3("Plot"),plotOutput("MyPlot"))
                 )
             ))
)
server <- function(input,output,session){</pre>
    MyData <- reactive({</pre>
            inFile<-input$dummyfile</pre>
            if (is.null(inFile))
                return(NULL)
            read.csv(inFile$datapath, header=TRUE, sep=',',
                     quote='')
    })
    output$summary<-renderPrint({</pre>
        summary(MyData()[,Col()])
    output$MyPlot<-renderPlot({</pre>
        if(is.null(MyData())!=T){
            ds_ts <-ts(MyData(), frequency=12)</pre>
            f <- decompose(ds_ts)</pre>
            plot(f)
        }
    })
}
shinyApp(ui, server)
```

Shiny applications not supported in static R Markdown documents

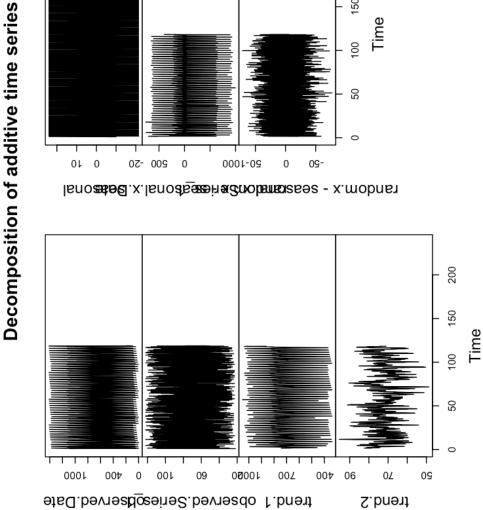
## Author: Erika Azabache

# Simple Time Series App

### dummy2.csv Choose csv File Browse...

Upload complete

## Plot



200

150

100

20