CS614 – Prof. Berardi
Homework #4 (Lecture 6)

In this assignment, you will use `ggplot` to recreate figures from a paper I published in the *Journal of Biomedical Informatics*. As part of the publishing process, I created an R package consisting of functions and data relevant to the manuscript that is hosted on my Github page and easily accessible via the `devtools` package in R. For each of the figures I request below, I will list the relevant R code to produce the requisite data. Please note that this code is also located in an appendix of the paper, but the journal editors mistakenly changed some of the syntax during their "correction" process, so I'd recommend using the scripts provided here.

Before reproducing any figures, we must load the data as follows:

```
library(devtools)
install_github("vancebee/MarkovSCD")
library(MarkovSCD)

#Load Baseline and Treatment Phase data for one home
BL = HM2$MassAve[HM2$Phase == "BL"]
TX = HM2$MassAve[HM2$Phase == "TX"]

#Define state boundaries
sb = seq(30,90,10)
```

1. Reproduce Figure 6, top center panel using the following code:

```
cv = dynamicsconv(tseries1 = BL, tseries2 = TX, nitvl = 10,statebounds = sb,lag = 6)
il1 = cv$ilength1[7]
il2 = cv$ilength2[7]

vv = validitycheck(tseries1 = BL, tseries2 = TX, ilength1 = il1,ilength2 = il2,
statebounds = sb,lag = 6)
```

`vv$norm` will give you the y-values for the required figure. The x-values must be entered manually into a data frame that also contains the y-value.

2. Reproduce Figure 6, bottom center panel. `vv$diagconfig`, from the code for Figure 1, will give you the data needed for plotting. *super* corresponds to *+1*, *diag* corresponds to *0*, and *sub* corresponds to *-1*. I'd recommend beginning by converting the matrix to a data frame and changing the column names to - 4:2. Then add a new column called *Pos* consisting of c(1,0,-1). Now you can run `melt()` using *Pos* as the id variable to get things into a useable format.

3. Reproduce the center panel of Figure B.8. using the following code to get the data:

```
le = lageval(tseries = TX,statebounds = sb, lagrange = c(1,2,seq(3,60,3)))
```

The output will be a list of vectors. The first element of this list corresponds to State 1, second to State 2, and so forth. Don't forget the vertical dashed line at x=1.

4. Reproduce the center figure of the second row of Figure 4 using the following code:

```
B = transmat(tseries = TX,statebounds = sb,lag = 6)
```

`B$prob` contains the appropriate data. You'll need to use `geom_tile()` where the x and y aesthetics are the Destination and Source bins, respectively, and the fill aesthetic is based on a cell's value. Similar to Problem 2, a clever introduction of a new variable plus the use of the `melt()` command can get things into the correct format. Also, `geom_text()` should be used to insert the text into each cell. You'll have to use `scale_fill_gradient()` to get the fill color right, where the limits are 0 to 1 and a square root transformation is used to move from a low value of *white* to a high value of *orange*. Opine on why the square root transformation was needed.