1.

```
# load(url("weather2011.rda"))
load("C:/Documents/GSLIS/490 Introduction to Data Science/HW9/weather2011.rda")

monthNames = c("January", "February", "March", "April",
               "May", "June", "July", "August", "September",
               "October", "November", "December")
daysInMonth = c(31, 28, 31, 30, 31, 30, 31,
                31, 30, 31, 30, 31)
cumDays = cumsum(c(1, daysInMonth))


makePlotRegion = function(xlim, ylim, bgcolor, ylabels,
                margins, cityName, xtop = TRUE) {
  # This function is to produce a blank plot that has
  # the proper axes labels, background color, etc.
  # It is to be used for both the top and bottom plot.

  # The parameters are
  # xlim is a two element numeric vector used for the two
  #   end points of the x axis
  # ylim is the same as xlim, but for the y axis
  # ylabels is a numeric vector of labels for "tick marks"
  #   on the y axis
  # We don't need to x labels because they are Month names
  # margins specifies the size of the plot margins (see mar parameter in par)
  # cityName is a character string to use in the title
  # xtop indicates whether the month names are to appear
  # at the top of the plot or the bottom of the plot
  #
  # See the assignment for a pdf image of the plot that is
  # produced as a result of calling this function.
```

```
  par(bg = bgcolor, mar = margins, xaxs="i", yaxs="i") # set the axis to fit within the data range, no extra space
  plot(NULL, xlim = xlim, ylim =ylim, yaxt = "n", xaxt = "n", axes = FALSE)
  axis(side = 2, at = ylabels, tick = TRUE, col = "grey", col.ticks = "cornsilk2", lwd = 1, las = 1) # left axis
  axis(side = 4, at = ylabels, tick = TRUE, col = "grey", col.ticks = "cornsilk2", lwd = 1, las = 1) # right axis
  title(main = cityName, adj = 0) # set the tile left-justified
  side = 1 + 2*xtop # Month names appear whether above or below
  axis(side, at = cumDays[-13] + 15, tick = FALSE, labels = monthNames, cex.axis = 0.5, font.axis = 2)
  # cumDays is cumulative days in a year
  # make the month names appear basically in the middle of each interval
}


drawTempRegion = function(day, high, low, col){
  # This plot will produce 365 rectangles, one for each day
  # It will be used for the record temps, normal temps, and
  # observed temps

  # day - a numeric vector of 365 dates
  # high - a numeric vector of 365 high temperatures
  # low - a numeric vector of 365 low temperatures
  # col - color to fill the rectangles

  rect(xleft = c(0:364), xright = c(1:365), ybottom = low, ytop = high, col = col, border = NA)
  # Each high and low for each day is represented as a narrow rectangle

}

addGrid = function(location, col, ltype, vertical = TRUE) {
  # This function adds a set of parallel grid lines
```

```
  # It will be used to place vertical and horizontal lines
  # on both temp and precip plots

  # location is a numeric vector of locations for the lines
  # col - the color to make the lines
  # ltype - the type of line to make
  # vertical - indicates whether the lines are vertical or horizontal

  if (vertical){
    abline(v = location, col = col, lty = ltype)
  }
  else if (!vertical){
    abline(h = location, col = col, lty = ltype)
  }

}

monthPrecip = function(day, dailyprecip, normal){
  # This function adds one month's precipitation to the
  #   precipitation plot.
  # It will be called 12 times, once for each month
  # It creates the cumulative precipitation curve,
  # fills the area below with color, add the total
  # precipitation for the month, and adds a reference
  # line and text for the normal value for the month

  # day a numeric vector of dates for the month
  # dailyprecip a numeric vector of precipitation recorded
  # for the month (any NAs can be set to 0)
  # normal a single value, which is the normal total precip
  # for the month
```

```
  points(x = day, y = dailyprecip, col = "deepskyblue4", type = "l", lwd = 3)
  polygon(x = c(day, max(day), day[1]), y = c(dailyprecip, 0, 0), col = "cornsilk3", border = NA)
  points(x =c(day[1], max(day)), y = rep(normal,2), type = "l", col = "cadetblue3", lwd = 1)

}

finalPlot = function(temp, precip){
  # The purpose of this function is to create the whole plot
  # Include here all of the set up that you need for
  # calling each of the above functions.
  # temp is the data frame sfoWeather or laxWeather
  # precip is the data frame sfoMonthlyPrecip or laxMonthlyPrecip


  # Here are some vectors that you might find handy

  # monthNames = c("January", "February", "March", "April",
  #              "May", "June", "July", "August", "September",
  #              "October", "November", "December")
  # daysInMonth = c(31, 28, 31, 30, 31, 30, 31,
  #                 31, 30, 31, 30, 31)
  # cumDays = cumsum(c(1, daysInMonth))

  # normPrecip = as.numeric(as.character(precip$normal))

  ### Fill in the various stages with your code


  ### Add any additional variables that you will need here


  ### Set up the graphics device to plot to pdf and layout
```

```
### the two plots on one canvas
### pdf("", width = , height = )
### layout( )

pdf("HuiLyu_LosAngeles.pdf", width = 8, height = 11)
layout(matrix(c(1,2), nrow = 2, ncol = 1, byrow = TRUE), heights = c(3,1))

### Call makePlotRegion to create the plotting region
### for the temperature plot

makePlotRegion(xlim = c(1,365), ylim = c(10,120), bgcolor = "cornsilk2", ylabels = seq(10,120,by = 10), margins = c(2,3,5,3), cityName = "Los Angeles's Weather in 2011", xtop = TRUE)

### Call drawTempRegion 3 times to add the rectangles for
### the record, normal, and observed temps

drawTempRegion(c(1:365), laxWeather$RecordHigh, laxWeather$RecordLow, col = "lemonchiffon3")
drawTempRegion(c(1:365), laxWeather$NormalHigh, laxWeather$NormalLow, col = "lemonchiffon4")
drawTempRegion(c(1:365), laxWeather$High, laxWeather$Low, col = "indianred4")

### Call addGrid to add the grid lines to the plot

addGrid(location = seq(10,120,by = 10), col = "cornsilk2", ltype = "solid", FALSE)
addGrid(location = cumDays, col = "black", ltype = "dotted", TRUE)

### Add the markers for the record breaking days

(laxWeather$Low - laxWeather$RecordLow)<0
minDiffer.low = min(abs(laxWeather$Low - laxWeather$RecordLow), na.rm = TRUE)
laxWeather[abs(laxWeather$Low - laxWeather$RecordLow)==minDiffer.low,]
segments(x0=279,y0=52-25,x1=279,y1=52)
```

```
segments(x0=280,y0=53-25,x1=280,y1=53)
text(x=282, y=53-28, labels = "RECORD LOW: 52", cex = .6, col = "black")
text(x=282, y=53-31, labels = "RECORD LOW: 53", cex = .6, col = "black")

(laxWeather$High - laxWeather$RecordHigh)>0
minDiffer.high = min(abs(laxWeather$High - laxWeather$RecordHigh), na.rm = TRUE)
laxWeather[abs(laxWeather$High - laxWeather$RecordHigh)==minDiffer.high,]
segments(x0=68,y0=81,x1=68,y1=81+25)
text(x=68, y=81+27, labels = "RECORD High: 81", cex = .6, col = "black")

### Add the titles

text(x=5, y = 118, labels = "Temperature", cex = 1.1, col = "black", font=2, adj = 0)
text(x=5, y = 114, labels = "Bars represent range between the daily high and low.", cex = .6, col =
"black", adj = 0)

rect(xleft=(364/2)-2, xright=(364/2)+2, ytop= 30+5, ybottom= 15, col="lemonchiffon3",border=NA)
rect(xleft=(364/2)-2, xright=(364/2)+2, ytop= 25+5, ybottom= 20, col="lemonchiffon4",border=NA)
rect(xleft=(364/2)-2, xright=(364/2), ytop= 28+5, ybottom= 23, col="indianred4",border=NA)

text(x=(364/2)-15, y = 37, labels = "RECORD HIGH", cex = .6, col = "black")
text(x=(364/2)-15, y = 13, labels = "RECORD LOW", cex = .6, col = "black")
text(x=(364/2)-40, y = 25, labels = "NORMAL RANGE", cex = .6, col = "black")
text(x=(364/2)-10, y = 25, labels = "{", cex = 4, col = "black")
text(x=(364/2)+25, y = 33, labels = "ACTUAL HIGH", cex = .6, col = "black")
text(x=(364/2)+25, y = 23, labels = "ACTUAL LOW", cex = .6, col = "black")

### Call makePlotRegion to create the plotting region
### for the precipitation plot

makePlotRegion(xlim = c(1, 365), ylim = c(0, 4.5), bgcolor = "cornsilk2",
        ylabels = seq(0, 4.5, by = 0.5), margins = c(3,3,2,3), cityName = "", xtop = FALSE)
```

```
### Call monthPrecip 12 times to create each months
### cumulative precipitation plot. To do this use
### sapply(1:12, function(m) {
###         code
###         monthPrecip(XXXX)
###         })
### the anonymous function calls monthPrecip with the
### appropriate arguments

normPrecip = as.numeric(as.character(precip$normal))
actualPrecip = as.numeric(as.character(precip$precip))

sapply(1:12, function(m){
 monthPrecip(day = cumDays[m]+temp$Day[temp$Month==m], dailyprecip = cumsum(temp$Precip[temp$Month==m]),
        normal = normPrecip[m])
})

### Call addGrid to add the grid lines to the plot

addGrid(location = seq(0, 4.5, by = 0.5), col = "cornsilk2", ltype = "solid", vertical = FALSE)
addGrid(location = cumDays, col = "black", ltype = "dotted", TRUE)

### Add the titles

title(main = "Precipitation", cex.main = 1.1, font = 2, adj=0)
title(main = "Cumulative monthly precipitation in inches compared with normal monthly precipitation.", cex.main= 0.6, adj = 0.5)
text(x=16, y = 3.2, labels = "NORMAL", cex = 0.6)
text(x=16, y = 0.8, labels = "ACTUAL", cex = 0.6)

text(x=cumDays[1]+2, y=normPrecip[1]-0.2, labels = normPrecip[1], cex = 0.6, adj = 0)
text(x=cumDays[2]-2, y=actualPrecip[1]-0.3, labels = actualPrecip[1], cex = 0.6, adj = 1)
```

```
  text(x=cumDays[2]+2, y=normPrecip[2]+0.2, labels = normPrecip[2], cex = 0.6, adj = 0)
  text(x=cumDays[3]-2, y=actualPrecip[2]+0.2, labels = actualPrecip[2], cex = 0.6, adj = 1)
  text(x=cumDays[3]+2, y=normPrecip[3]+0.2, labels = normPrecip[3], cex = 0.6, adj = 0)
  text(x=cumDays[4]-2, y=actualPrecip[3]+0.2, labels = actualPrecip[3], cex = 0.6, adj = 1)
  text(x=cumDays[4]+2, y=normPrecip[4]+0.2, labels = normPrecip[4], cex = 0.6, adj = 0)
  text(x=cumDays[5]-2, y=actualPrecip[4]+0.2, labels = actualPrecip[4], cex = 0.6, adj = 1)
  text(x=cumDays[5]+2, y=normPrecip[5]+0.2, labels = normPrecip[5], cex = 0.6, adj = 0)
  text(x=cumDays[6]-2, y=actualPrecip[5]+0.2, labels = actualPrecip[5], cex = 0.6, adj = 1)
  text(x=cumDays[6]+2, y=normPrecip[6]+0.2, labels = normPrecip[6], cex = 0.6, adj = 0)
  text(x=cumDays[7]-2, y=actualPrecip[6]+0.2, labels = actualPrecip[6], cex = 0.6, adj = 1)
  text(x=cumDays[7]+2, y=normPrecip[7]+0.2, labels = normPrecip[7], cex = 0.6, adj = 0)
  text(x=cumDays[8]-2, y=actualPrecip[7]+0.2, labels = actualPrecip[7], cex = 0.6, adj = 1)
  text(x=cumDays[8]+2, y=normPrecip[8]+0.2, labels = normPrecip[8], cex = 0.6, adj = 0)
  text(x=cumDays[9]-2, y=actualPrecip[8]+0.2, labels = actualPrecip[8], cex = 0.6, adj = 1)
  text(x=cumDays[9]+2, y=normPrecip[9]+0.2, labels = normPrecip[9], cex = 0.6, adj = 0)
  text(x=cumDays[10]-2, y=actualPrecip[9]+0.2, labels = actualPrecip[9], cex = 0.6, adj = 1)
  text(x=cumDays[10]+2, y=normPrecip[10]+0.2, labels = normPrecip[10], cex = 0.6, adj = 0)
  text(x=cumDays[11]-2, y=actualPrecip[10]+0.2, labels = actualPrecip[10], cex = 0.6, adj = 1)
  text(x=cumDays[11]+2, y=normPrecip[11]+0.2, labels = normPrecip[11], cex = 0.6, adj = 0)
  text(x=cumDays[12]-2, y=actualPrecip[11]+0.2, labels = actualPrecip[11], cex = 0.6, adj = 1)
  text(x=cumDays[12]+2, y=normPrecip[12]+0.2, labels = normPrecip[12], cex = 0.6, adj = 0)
  text(x=cumDays[13]-2, y=actualPrecip[12]+0.2, labels = actualPrecip[12], cex = 0.6, adj = 1)


  ### Close the pdf device dev.off()
  dev.off()
}

### Call: finalPlot(temp = sfoWeather, precip = sfoMonthlyPrecip)

finalPlot(temp = laxWeather, precip = laxMonthlyPrecip)
```

2.

I will take the Digital Curation Centre (DCC) Lifecycle Model as the reference for my work on plot creation. The DCC Lifecycle Model consists of several circles from internal to external. The inner cycles are mainly disposal of data itself, while the outer cycles are collaboration work between researchers and curators.

The core is the statistical data, for the Temperature and Precipitation juxtaposed plot creation work, the core data is the numbers in the tables. And, the description and preprocessing data in DCC Lifecycle is the well-formatted data frames in R for my work. One thing that belongs to the DCC Lifecycle Model but does not reflect for my work is the preservation and curation circle part. I suppose there is no certain task for long-term preservation purpose. (But the pdf format of the final plot is a relatively good digital object format.) Then, the ingestion and transformation of the plot creation work could be the R script code for disposal. The final plot I created is the presentation circle part of the DCC Lifecycle Model. Besides, the whole plot recreation work itself actually is a reuse and reproduce of the original data. So generally my work on plot recreation coincides with the DCC Lifecycle Model.

3. Extra Credit

```
finalPlot2 = function(temp, precip){
  # The purpose of this function is to create the whole plot
  # Include here all of the set up that you need for
  # calling each of the above functions.
  # temp is the data frame sfoWeather or laxWeather
  # precip is the data frame sfoMonthlyPrecip or laxMonthlyPrecip



  # Here are some vectors that you might find handy


  # monthNames = c("January", "February", "March", "April",
  #             "May", "June", "July", "August", "September",
  #             "October", "November", "December")
  # daysInMonth = c(31, 28, 31, 30, 31, 30, 31,
  #                 31, 30, 31, 30, 31)
  # cumDays = cumsum(c(1, daysInMonth))


  # normPrecip = as.numeric(as.character(precip$normal))
```

```
### Fill in the various stages with your code


### Add any additional variables that you will need here


### Set up the graphics device to plot to pdf and layout
### the two plots on one canvas
### pdf("", width = , height = )
### layout(  )

pdf("HuiLyu_SanFrancisco.pdf", width = 8, height = 11)
layout(matrix(c(1,2), nrow = 2, ncol = 1, byrow = TRUE), heights = c(3,1))

### Call makePlotRegion to create the plotting region
### for the temperature plot

makePlotRegion(xlim = c(1,365), ylim = c(0,120), bgcolor = "cornsilk2", ylabels = seq(0,120,by = 10), margins = c(2,3,5,3), cityName = "San Francisco's Weather in 2011", xtop = TRUE)

### Call drawTempRegion 3 times to add the rectangles for
### the record, normal, and observed temps

drawTempRegion(c(1:365), sfoWeather$RecordHigh, sfoWeather$RecordLow, col = "lemonchiffon3")
drawTempRegion(c(1:365), sfoWeather$NormalHigh, sfoWeather$NormalLow, col = "lemonchiffon4")
drawTempRegion(c(1:365), sfoWeather$High, sfoWeather$Low, col = "indianred4")

### Call addGrid to add the grid lines to the plot

addGrid(location = seq(0,120,by = 10), col = "cornsilk2", ltype = "solid", FALSE)
addGrid(location = cumDays, col = "black", ltype = "dotted", TRUE)
```

```
### Add the markers for the record breaking days

(sfoWeather$Low - sfoWeather$RecordLow)<0
minDiffer.low = min(abs(sfoWeather$Low - sfoWeather$RecordLow), na.rm = TRUE)
sfoWeather[abs(sfoWeather$Low - sfoWeather$RecordLow)==minDiffer.low,]
segments(x0=51,y0=37-25,x1=51,y1=37)
segments(x0=57,y0=35-25,x1=57,y1=35)
segments(x0=98,y0=41-25,x1=98,y1=41)
text(x=51, y=37-28, labels = "RECORD LOW: 37", cex = .6, col = "black")
text(x=57, y=35-28, labels = "RECORD LOW: 35", cex = .6, col = "black")
text(x=98, y=41-28, labels = "RECORD LOW: 41", cex = .6, col = "black")

(sfoWeather$High - sfoWeather$RecordHigh)>0
minDiffer.high = min(abs(sfoWeather$High - sfoWeather$RecordHigh), na.rm = TRUE)
sfoWeather[abs(sfoWeather$High - sfoWeather$RecordHigh)==minDiffer.high,]
segments(x0=25,y0=68,x1=25,y1=68+25)
text(x=25, y=68+27, labels = "RECORD High: 68", cex = .6, col = "black")
segments(x0=36,y0=73,x1=36,y1=73+25)
text(x=36, y=73+28, labels = "RECORD High: 73", cex = .6, col = "black")
segments(x0=37,y0=72,x1=37,y1=72+25)
text(x=37, y=72+27, labels = "RECORD High: 72", cex = .6, col = "black")
segments(x0=90,y0=80,x1=90,y1=80+25)
text(x=90, y=80+27, labels = "RECORD High: 80", cex = .6, col = "black")
segments(x0=124,y0=84,x1=124,y1=84+25)
text(x=124, y=84+27, labels = "RECORD High: 84", cex = .6, col = "black")

### Add the titles

text(x=5, y = 118, labels = "Temperature", cex = 1.1, col = "black", font=2, adj = 0)
text(x=5, y = 114, labels = "Bars represent range between the daily high and low.", cex = .6, col = "black", adj = 0)
```

```
rect(xleft=(364/2)-2, xright=(364/2)+2, ytop= 30+5, ybottom= 15, col="lemonchiffon3",border=NA)
rect(xleft=(364/2)-2, xright=(364/2)+2, ytop= 25+5, ybottom= 20, col="lemonchiffon4",border=NA)
rect(xleft=(364/2)-2, xright=(364/2), ytop= 28+5, ybottom= 23, col="indianred4",border=NA)

text(x=(364/2)-15, y = 37, labels = "RECORD HIGH", cex = .6, col = "black")
text(x=(364/2)-15, y = 13, labels = "RECORD LOW", cex = .6, col = "black")
text(x=(364/2)-40, y = 25, labels = "NORMAL RANGE", cex = .6, col = "black")
text(x=(364/2)-10, y = 25, labels = "{", cex = 4, col = "black")
text(x=(364/2)+25, y = 33, labels = "ACTUAL HIGH", cex = .6, col = "black")
text(x=(364/2)+25, y = 23, labels = "ACTUAL LOW", cex = .6, col = "black")

### Call makePlotRegion to create the plotting region
### for the precipitation plot

makePlotRegion(xlim = c(1, 365), ylim = c(0, 6.1), bgcolor = "cornsilk2",
        ylabels = seq(0, 6, by = 1), margins = c(3,3,2,3), cityName = "", xtop = FALSE)

### Call monthPrecip 12 times to create each months
### cumulative precipitation plot. To do this use
### sapply(1:12, function(m) {
###         code
###         monthPrecip(XXXX)
###         })
### the anonymous function calls monthPrecip with the
### appropriate arguments

normPrecip = as.numeric(as.character(precip$normal))
actualPrecip = as.numeric(as.character(precip$precip))

sapply(1:12, function(m){
  monthPrecip(day = cumDays[m]+temp$Day[temp$Month==m], dailyprecip =
cumsum(temp$Precip[temp$Month==m]),
        normal = normPrecip[m])
```

```
})

### Call addGrid to add the grid lines to the plot

addGrid(location = seq(0, 6, by = 1), col = "cornsilk2", ltype = "solid", vertical = FALSE)
addGrid(location = cumDays, col = "black", ltype = "dotted", TRUE)

### Add the titles

title(main = "Precipitation", cex.main = 1.1, font = 2, adj=0)
title(main = "Cumulative monthly precipitation in inches compared with normal monthly precipitation.", cex.main= 0.6, adj = 0.5)
text(x=16, y = 4.4, labels = "NORMAL", cex = 0.6)
text(x=16, y = 0.8, labels = "ACTUAL", cex = 0.6)

text(x=cumDays[1]+2, y=normPrecip[1]-0.2, labels = normPrecip[1], cex = 0.6, adj = 0)
text(x=cumDays[2]-2, y=actualPrecip[1]+0.2, labels = actualPrecip[1], cex = 0.6, adj = 1)
text(x=cumDays[2]+2, y=normPrecip[2]+0.2, labels = normPrecip[2], cex = 0.6, adj = 0)
text(x=cumDays[3]-2, y=actualPrecip[2]+0.2, labels = actualPrecip[2], cex = 0.6, adj = 1)
text(x=cumDays[3]+2, y=normPrecip[3]+0.2, labels = normPrecip[3], cex = 0.6, adj = 0)
text(x=cumDays[4]-2, y=actualPrecip[3]+0.2, labels = actualPrecip[3], cex = 0.6, adj = 1)
text(x=cumDays[4]+2, y=normPrecip[4]+0.2, labels = normPrecip[4], cex = 0.6, adj = 0)
text(x=cumDays[5]-2, y=actualPrecip[4]+0.2, labels = actualPrecip[4], cex = 0.6, adj = 1)
text(x=cumDays[5]+2, y=normPrecip[5]+0.2, labels = normPrecip[5], cex = 0.6, adj = 0)
text(x=cumDays[6]-2, y=actualPrecip[5]+0.2, labels = actualPrecip[5], cex = 0.6, adj = 1)
text(x=cumDays[6]+2, y=normPrecip[6]+0.2, labels = normPrecip[6], cex = 0.6, adj = 0)
text(x=cumDays[7]-2, y=actualPrecip[6]+0.2, labels = actualPrecip[6], cex = 0.6, adj = 1)
text(x=cumDays[7]+2, y=normPrecip[7]+0.2, labels = normPrecip[7], cex = 0.6, adj = 0)
text(x=cumDays[8]-2, y=actualPrecip[7]+0.2, labels = actualPrecip[7], cex = 0.6, adj = 1)
text(x=cumDays[8]+2, y=normPrecip[8]+0.2, labels = normPrecip[8], cex = 0.6, adj = 0)
text(x=cumDays[9]-2, y=actualPrecip[8]+0.2, labels = actualPrecip[8], cex = 0.6, adj = 1)
text(x=cumDays[9]+2, y=normPrecip[9]+0.2, labels = normPrecip[9], cex = 0.6, adj = 0)
text(x=cumDays[10]-2, y=actualPrecip[9]+0.2, labels = actualPrecip[9], cex = 0.6, adj = 1)
```

```
  text(x=cumDays[10]+2, y=normPrecip[10]+0.2, labels = normPrecip[10], cex = 0.6, adj = 0)
  text(x=cumDays[11]-2, y=actualPrecip[10]+0.2, labels = actualPrecip[10], cex = 0.6, adj = 1)
  text(x=cumDays[11]+2, y=normPrecip[11]+0.2, labels = normPrecip[11], cex = 0.6, adj = 0)
  text(x=cumDays[12]-2, y=actualPrecip[11]+0.2, labels = actualPrecip[11], cex = 0.6, adj = 1)
  text(x=cumDays[12]+2, y=normPrecip[12]+0.2, labels = normPrecip[12], cex = 0.6, adj = 0)
  text(x=cumDays[13]-2, y=actualPrecip[12]+0.2, labels = actualPrecip[12], cex = 0.6, adj = 1)


  ### Close the pdf device dev.off()
  dev.off()
}

### Call: finalPlot(temp = sfoWeather, precip = sfoMonthlyPrecip)

finalPlot2(temp = sfoWeather, precip = sfoMonthlyPrecip)









finalPlot3 = function(temp, precip){
  # The purpose of this function is to create the whole plot
  # Include here all of the set up that you need for
  # calling each of the above functions.
  # temp is the data frame sfoWeather or laxWeather
  # precip is the data frame sfoMonthlyPrecip or laxMonthlyPrecip


  # Here are some vectors that you might find handy
```

```
# monthNames = c("January", "February", "March", "April",
#             "May", "June", "July", "August", "September",
#             "October", "November", "December")
# daysInMonth = c(31, 28, 31, 30, 31, 30, 31,
#               31, 30, 31, 30, 31)
# cumDays = cumsum(c(1, daysInMonth))

# normPrecip = as.numeric(as.character(precip$normal))

### Fill in the various stages with your code



### Add any additional variables that you will need here



### Set up the graphics device to plot to pdf and layout
### the two plots on one canvas
### pdf("", width = , height = )
### layout(  )

pdf("HuiLyu_LosAngeles_for_compare.pdf", width = 8, height = 11)
layout(matrix(c(1,2), nrow = 2, ncol = 1, byrow = TRUE), heights = c(3,1))

### Call makePlotRegion to create the plotting region
### for the temperature plot

makePlotRegion(xlim = c(1,365), ylim = c(0,120), bgcolor = "cornsilk2", ylabels = seq(0,120,by =
10), margins = c(2,3,5,3), cityName = "Los Angeles's Weather in 2011", xtop = TRUE)

### Call drawTempRegion 3 times to add the rectangles for
### the record, normal, and observed temps
```

```
drawTempRegion(c(1:365), laxWeather$RecordHigh, laxWeather$RecordLow, col = "lemonchiffon3")
drawTempRegion(c(1:365), laxWeather$NormalHigh, laxWeather$NormalLow, col = "lemonchiffon4")
drawTempRegion(c(1:365), laxWeather$High, laxWeather$Low, col = "indianred4")

### Call addGrid to add the grid lines to the plot

addGrid(location = seq(0,120,by = 10), col = "cornsilk2", ltype = "solid", FALSE)
addGrid(location = cumDays, col = "black", ltype = "dotted", TRUE)

### Add the markers for the record breaking days

(laxWeather$Low - laxWeather$RecordLow)<0
minDiffer.low = min(abs(laxWeather$Low - laxWeather$RecordLow), na.rm = TRUE)
laxWeather[abs(laxWeather$Low - laxWeather$RecordLow)==minDiffer.low,]
segments(x0=279,y0=52-25,x1=279,y1=52)
segments(x0=280,y0=53-25,x1=280,y1=53)
text(x=282, y=53-28, labels = "RECORD LOW: 52", cex = .6, col = "black")
text(x=282, y=53-31, labels = "RECORD LOW: 53", cex = .6, col = "black")

(laxWeather$High - laxWeather$RecordHigh)>0
minDiffer.high = min(abs(laxWeather$High - laxWeather$RecordHigh), na.rm = TRUE)
laxWeather[abs(laxWeather$High - laxWeather$RecordHigh)==minDiffer.high,]
segments(x0=68,y0=81,x1=68,y1=81+25)
text(x=68, y=81+27, labels = "RECORD High: 81", cex = .6, col = "black")

### Add the titles

text(x=5, y = 118, labels = "Temperature", cex = 1.1, col = "black", font=2, adj = 0)
text(x=5, y = 114, labels = "Bars represent range between the daily high and low.", cex = .6, col = "black", adj = 0)
```

```
rect(xleft=(364/2)-2, xright=(364/2)+2, ytop= 30+5, ybottom= 15, col="lemonchiffon3",border=NA)
rect(xleft=(364/2)-2, xright=(364/2)+2, ytop= 25+5, ybottom= 20, col="lemonchiffon4",border=NA)
rect(xleft=(364/2)-2, xright=(364/2), ytop= 28+5, ybottom= 23, col="indianred4",border=NA)

text(x=(364/2)-15, y = 37, labels = "RECORD HIGH", cex = .6, col = "black")
text(x=(364/2)-15, y = 13, labels = "RECORD LOW", cex = .6, col = "black")
text(x=(364/2)-40, y = 25, labels = "NORMAL RANGE", cex = .6, col = "black")
text(x=(364/2)-10, y = 25, labels = "{", cex = 4, col = "black")
text(x=(364/2)+25, y = 33, labels = "ACTUAL HIGH", cex = .6, col = "black")
text(x=(364/2)+25, y = 23, labels = "ACTUAL LOW", cex = .6, col = "black")

### Call makePlotRegion to create the plotting region
### for the precipitation plot

makePlotRegion(xlim = c(1, 365), ylim = c(0, 6.1), bgcolor = "cornsilk2",
        ylabels = seq(0, 6, by = 1), margins = c(3,3,2,3), cityName = "", xtop = FALSE)

### Call monthPrecip 12 times to create each months
### cumulative precipitation plot. To do this use
### sapply(1:12, function(m) {
###         code
###         monthPrecip(XXXX)
###         })
### the anonymous function calls monthPrecip with the
### appropriate arguments

normPrecip = as.numeric(as.character(precip$normal))
actualPrecip = as.numeric(as.character(precip$precip))

sapply(1:12, function(m){
  monthPrecip(day = cumDays[m]+temp$Day[temp$Month==m], dailyprecip =
cumsum(temp$Precip[temp$Month==m]),
        normal = normPrecip[m])
```

```
})

### Call addGrid to add the grid lines to the plot

addGrid(location = seq(0, 6, by = 1), col = "cornsilk2", ltype = "solid", vertical = FALSE)
addGrid(location = cumDays, col = "black", ltype = "dotted", TRUE)

### Add the titles

title(main = "Precipitation", cex.main = 1.1, font = 2, adj=0)
title(main = "Cumulative monthly precipitation in inches compared with normal monthly precipitation.", cex.main= 0.6, adj = 0.5)
text(x=16, y = 3.2, labels = "NORMAL", cex = 0.6)
text(x=16, y = 0.8, labels = "ACTUAL", cex = 0.6)

text(x=cumDays[1]+2, y=normPrecip[1]-0.2, labels = normPrecip[1], cex = 0.6, adj = 0)
text(x=cumDays[2]-2, y=actualPrecip[1]-0.3, labels = actualPrecip[1], cex = 0.6, adj = 1)
text(x=cumDays[2]+2, y=normPrecip[2]+0.2, labels = normPrecip[2], cex = 0.6, adj = 0)
text(x=cumDays[3]-2, y=actualPrecip[2]+0.2, labels = actualPrecip[2], cex = 0.6, adj = 1)
text(x=cumDays[3]+2, y=normPrecip[3]+0.2, labels = normPrecip[3], cex = 0.6, adj = 0)
text(x=cumDays[4]-2, y=actualPrecip[3]+0.2, labels = actualPrecip[3], cex = 0.6, adj = 1)
text(x=cumDays[4]+2, y=normPrecip[4]+0.2, labels = normPrecip[4], cex = 0.6, adj = 0)
text(x=cumDays[5]-2, y=actualPrecip[4]+0.2, labels = actualPrecip[4], cex = 0.6, adj = 1)
text(x=cumDays[5]+2, y=normPrecip[5]+0.2, labels = normPrecip[5], cex = 0.6, adj = 0)
text(x=cumDays[6]-2, y=actualPrecip[5]+0.2, labels = actualPrecip[5], cex = 0.6, adj = 1)
text(x=cumDays[6]+2, y=normPrecip[6]+0.2, labels = normPrecip[6], cex = 0.6, adj = 0)
text(x=cumDays[7]-2, y=actualPrecip[6]+0.2, labels = actualPrecip[6], cex = 0.6, adj = 1)
text(x=cumDays[7]+2, y=normPrecip[7]+0.2, labels = normPrecip[7], cex = 0.6, adj = 0)
text(x=cumDays[8]-2, y=actualPrecip[7]+0.2, labels = actualPrecip[7], cex = 0.6, adj = 1)
text(x=cumDays[8]+2, y=normPrecip[8]+0.2, labels = normPrecip[8], cex = 0.6, adj = 0)
text(x=cumDays[9]-2, y=actualPrecip[8]+0.2, labels = actualPrecip[8], cex = 0.6, adj = 1)
text(x=cumDays[9]+2, y=normPrecip[9]+0.2, labels = normPrecip[9], cex = 0.6, adj = 0)
text(x=cumDays[10]-2, y=actualPrecip[9]+0.2, labels = actualPrecip[9], cex = 0.6, adj = 1)
```

```
  text(x=cumDays[10]+2, y=normPrecip[10]+0.2, labels = normPrecip[10], cex = 0.6, adj = 0)
  text(x=cumDays[11]-2, y=actualPrecip[10]+0.2, labels = actualPrecip[10], cex = 0.6, adj = 1)
  text(x=cumDays[11]+2, y=normPrecip[11]+0.2, labels = normPrecip[11], cex = 0.6, adj = 0)
  text(x=cumDays[12]-2, y=actualPrecip[11]+0.2, labels = actualPrecip[11], cex = 0.6, adj = 1)
  text(x=cumDays[12]+2, y=normPrecip[12]+0.2, labels = normPrecip[12], cex = 0.6, adj = 0)
  text(x=cumDays[13]-2, y=actualPrecip[12]+0.2, labels = actualPrecip[12], cex = 0.6, adj = 1)


  ### Close the pdf device dev.off()
  dev.off()
}

### Call: finalPlot(temp = sfoWeather, precip = sfoMonthlyPrecip)

finalPlot3(temp = laxWeather, precip = laxMonthlyPrecip)
```