Making Moving Bubble Charts[1]

Micah S. Woods

4 May 2012

[1] This is an almost step-by-step guide describing how I made a chart with climatological normal data for 52 world cities

The weather, specifically the temperature, precipitation, and sunshine, has a controlling influence on the growth and performance of turfgrass. Scatterplots can be used to visualize the data and in particular to compare and contrast the growing environments in different cities where turfgrasses are managed. Scatterplots that move as the data change with time are sometimes referred to as bubble charts or motion charts. I describe here the process I use to make these charts using climatological normal data from the World Meteorological Organization, R, and the googleVis package. The chart produced by this process can be seen at this page:

`http://climate.asianturfgrass.com/node/12`

What are we trying to do?

At the outset it is useful to consider what we are trying to achieve. The software we will use makes the creation of the chart a matter of only a few lines of code. The time-consuming part of this is getting the data and ensuring it is in the proper format.

We will use the googleVis package[2] to interface with the Google Visualization API to create the chart from the data that we provide. Once we have the data, loaded into R[3] as a data table, we will require only two lines of code to create the chart (Figure 1).

We need to produce the climate.52 data file, and we need it to contain the cities for which we will plot the data in one column (the **idvar**), another column will contain the date for which each climatological normal was measured (the **timevar**), and we will need columns that contain the associated temperature, precipitation, and sunshine hours data. The googleVis package makes it easy to generate the chart, because once we assign the **idvar** and the **timevar** to variables in the data table, the software automatically assigns the other columns of data to be those that are plotted.

Let's call the columns in our data table *variables*, and rows in our data table *observations*. We need to make a data table that has five variables (columns): city name, date, temperature, precipitation, and sunshine hours.[4] We can have as many observations as we want, but for the purposes of this chart let's pick 52 cities from around the world, and for each city go through an annual cycle (January to January) of monthly climatological normals.

[2] Markus Gesmann and Diego de Castillo. googleVis: Interface between R and the Google Visualisation API. *The R Journal*, 3 (2):40–44, December 2011. URL `http://journal.r-project.org/archive/2011-2/RJournal_2011-2_Gesmann+de~Castillo.pdf`

[3] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL `http://www.R-project.org`. ISBN 3-900051-07-0

```
17 w = gvisMotionChart(data=climate.52,
   idvar="city", timevar="date",
   chartid = "ATC_climate_chart_52")
18 plot(w)
```

Figure 1: Once the data are formatted, we require only these two lines of code to generate the chart

[4] I add an extra column to indicate which continent the city is on, which can be useful in color coding of the final chart

Step 1: getting the data

This is the most time-consuming part of the exercise. Not only do we need to consider which data we will include in the chart, but

we must find the data and get them into one file. The formatting of these data will be done later.

For this chart, I wanted to include areas where cool-season grasses are grown, with less of an emphasis on warm-season areas.[5] I'd recently traveled to Seoul, and Beijing, and wondered how the climatological normals in those cities compared with other cities in temperate climates, particularly cities in the Midwest and Northeast of the United States, where there are high concentrations of golf courses, other managed turfs, and a large amount of turfgrass research.

[5] I've made a number of static and moving charts showing data for warm-season areas, available here: http://climate.asianturfgrass.com

The Hong Kong Observatory (HKO) have conveniently placed much of the World Meteorological Organization (WMO) climatological normal information dataset on their website, in tables.[6] I make a new spreadsheet in Google Docs and use the **ImportHtml** function to read in the data, city by selected city. That populates the spreadsheet with the data for up to 50 cities[7] and for additional cities, I highlighted the data in the table and then cut-and-pasted it into the spreadsheet.

[6] http://www.hko.gov.hk/wxinfo/climat/world/eng/world_climat_e.htm

[7] The **ImportHtml** function is limited to 50 uses within one spreadsheet

Step 2: formatting the data

A BIT MORE WORK is required to make sure we have just the right data, formatted just the right way. For New York City and Los Angeles, the sunshine hours are in the WMO database but are not on the HKO website. I downloaded the sunshine data for those two cities from the WMO normals dataset[8] and converted it from monthly hours into daily hours to match the other 50 cities.

[8] http://data.un.org/Explorer.aspx?d=CLINO

Four of the cities (Cape Town, Melbourne, Milano, and Sydney) did not have a mean monthly temperature in the HKO website tables, so I calculated the mean temperature for those cities based on the maximum and minimum temperatures that were in the dataset.

I deleted all extraneous data, leaving me with a file containing a header row and 156 rows of data, each of the 52 selected cities having three rows: one for mean monthly temperature, one for mean monthly precipitation, and one for mean daily sunshine hours. The file had 14 columns of data: the city name, the name for the parameter being measured (temperature, precipitation, or sunshine), and then 12 values for each parameter, from January through to December, as these are mean monthly normals.

This doesn't work for the chart we are trying to create. We need to have 5 columns of data, as mentioned earlier: city name, date, temperature, precipitation, and sunshine hours. Fortunately, the **reshape** package[9] in R will allow us to easily reformat the data.

[9] Hadley Wickham. Reshaping data with the reshape package. *Journal of Statistical Software*, 21(12), 2007. URL http://www.jstatsoft.org/v21/i12/paper

Step 3: preparing the data for plotting

We now need to use R.[10] We will be using the **reshape** and **googleVis** packages.[11] The code to generate the chart is given in Figure 2. Note that any text beginning with **#** is a comment and will not be executed. One must adjust the file location path to its location on your computer in order for this code to work.

[10] if you don't have R, download it here: `http://www.r-project.org/`

[11] if you do not have those packages, you can get them by executing **install.packages("reshape")** and **install.packages("googleVis")** from the R command line

```
library(googleVis) # loads the googleVis package
library(reshape) # loads the reshape package

# read in the fulll dataset that needs reformatting or columns and rows
data.52 <- read.table("/Users/woods/R/cities52-3.csv", header=TRUE, sep=",")
# melt the data so that observations and variables can be formatted properly
 for plotting
melted.52 <- melt(data.52)
# casts the data into the desired format of observations as rows, variables
 as columns, variable is DATE, measure is parameter of sun, temp, or precip
shaped.52 <- cast(melted.52, city + variable ~ measure)
#export, to make few minor changes, format date, etc
write.table (shaped.52, "cities52-4.csv", sep=",")

#now ready to read in the data for plotting
# this reads in the formatted data set with extra January data to make full
 annual cycle, column headers spelled exactly as they should appear in the
 chart as parameter names
climate.52 <- read.table("/Users/woods/R/cities52-5.csv", header=TRUE,
 sep=",")
#formats date variable as a date
climate.52$Date <- as.Date(climate.52$Date)
# creates the HTML code for the chart
w = gvisMotionChart(data=climate.52, idvar="City", timevar="Date", chartid =
 "ATC_climate_chart_52")
# loads the chart, in a browser
plot(w)
```

Figure 2: This code will reformat (reshape) the data so it can be plotted and generates the motion chart with 52 cities

You may download the datasets for this chart if you wish to try this yourself. The datasets cities52-3.csv, cities52-4.csv, and cities52-5.csv are available for download here:

`http://calendar.asianturfgrass.com/cities52-3.csv`
`http://calendar.asianturfgrass.com/cities52-4.csv`
`http://calendar.asianturfgrass.com/cities52-5.csv`

The data in these files, and the code shown in Figure 2 creates an interactive motion chart that shows average climatological data for 52 world cities throughout a 366 day period. The HTML code can be used to embed the chart on a website. These types of charts would be useful, perhaps, in looking at growing degree days, disease pressure, growth potential, soil test data, fertilizer applications, cumulative work done, golfer rounds, labor hours, and many other

types of data that change through time.

How long does it take to create these charts? For this one of 52 world cities, it took me about three hours to get the data organized into one file and to check that all was in order with the data. Then it took an hour or so to reshape the data and to write the code to generate the chart. It would normally take only ten minutes to reshape the data and generate the chart, but something seems to have changed in the **googleVis** package since the last time I had used it, so there was some difficulty in getting the data to display properly on the chart.

References

Markus Gesmann and Diego de Castillo. googleVis: Interface between R and the Google Visualisation API. *The R Journal*, 3(2): 40–44, December 2011. URL `http://journal.r-project.org/archive/2011-2/RJournal_2011-2_Gesmann+de~Castillo.pdf`.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

Hadley Wickham. Reshaping data with the reshape package. *Journal of Statistical Software*, 21(12), 2007. URL `http://www.jstatsoft.org/v21/i12/paper`.