HW5: assigned 4/26, due 5/1 by 11:59 PM

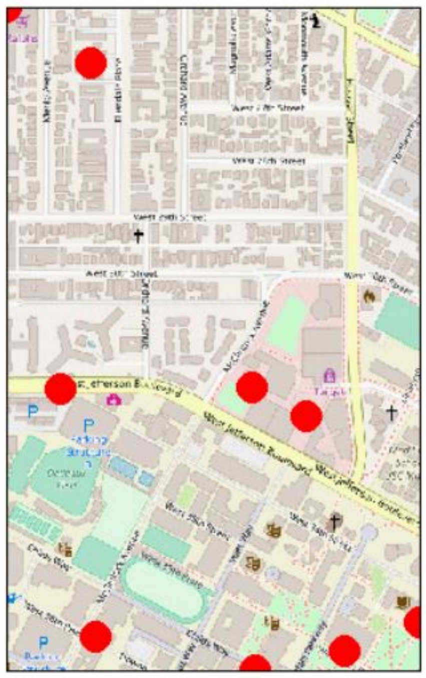
Total points: 6 [3+3]

With this HW, you get a chance to dip your toes into the 'R' ocean :) The assignment consists of renderi ('plotting'), using R functions that are in spatial-data-oriented packages ("libraries"), the (lat,long) locati data you collected for your HW3.

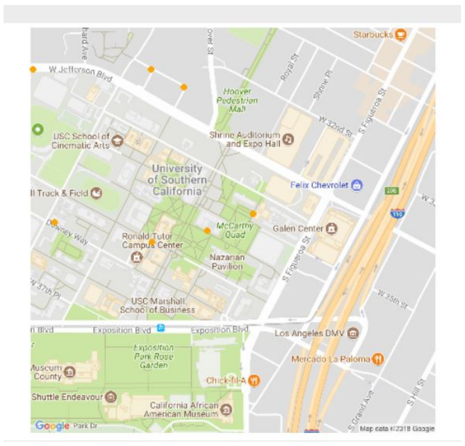
The 'point' (!) is to make you experience geospatial data handling in R. R's powerful, extensive capabiliti (via add-on packages) and simple syntax make it possible to query, analyze and display a variety of spati data.

You'd need to plot the points you collected for HW3, in two different ways, over a raster map.

The first way [3 points] uses the 'tmap, tmaptools, rgdal, sp, OpenStreetMap, osmdata' packages, and produces this kind of result:



The second technique [3 points] uses the 'PBSmapping, ggmap' packages, and generates such an outpu


Starbucks
W Jefferson Blvd
Hoover Pedestrian Mall
Shrine Auditorium and Expo Hall
USC School of Cinematic Arts
University of Southern California
Felix Chevrolet
Galen Center
McCarthy Quad
Ronald Tutor Campus Center
Nazarian Pavilion
USC Marshall School of Business
Exposition Blvd
Los Angeles DMV
Mercado La Paloma
Exposition Park Rose Garden
Chick-fil-A
Shuttle Endeavour
California African American Museum
Google
Map data ©2018 Google

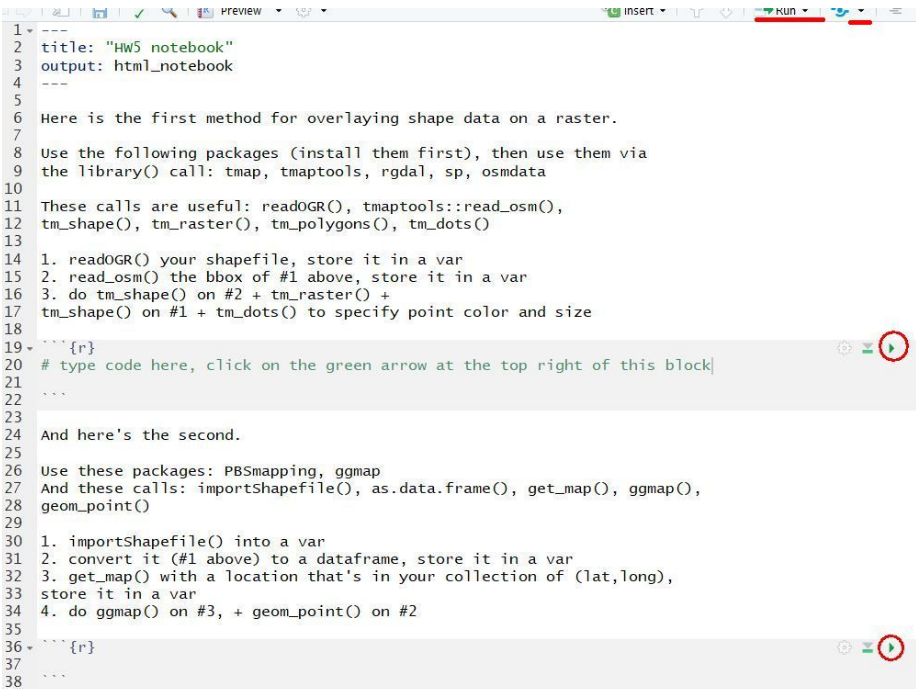
Don't worry about the blurriness of the resulting images (there are ways to fix it if you do this 'for real' i
the future).

Make sure you install R (https://cran.r-project.org/), then, RStudio (https://www.rstudio.com/) - you'll d
the HW using RStudio [RStudio is an IDE for the underlying R distribution]. Here (https://cran.r-project.org/doc/manuals/R-intro.pdf) is the 'official' intro to R - you do NOT read/know it in its entirety,
do this HW! Reading sections 1-7 of this shorter intro' (https://cran.r-project.org/doc/contrib/Torfs+Brau
Short-R-Intro.pdf) should be sufficient.

For both techniques, you'd need a 'shapefile' (which consists of .shp, .shx, .prj and .dbf files, all in the same directory): convert your HW3's kml file that contains just the points you surveyed (without convex hull and nearest neighbors etc.), into a shapefile, using this link: https://mygeodata.cloud/converter/kml
to-shp (https://mygeodata.cloud/converter/kml-to-shp) You'd upload your .kml, click on 'Convert', download the resulting .zip file and unpack it in a folder - that folder will contain a .shp, .shx, .prj and .c
set of files; of these, you'd use the .shp file in your function calls (the .shp file contents point to spatial d
and metadata stored in the other three files). Note: start with a simple filename such as hw5.kml, not something like hw3.txt.kml etc. [otherwise you'll run into problems when you use the resulting .shp file

Here (starter.Rmd) is a starter.Rmd "R notebook" (where R code as well as Markdown text can be kept, hence the .Rmd extension), with VERY valuable hints I'm providing - it tells you, **step by step, what to dc
for each of the two techniques (what packages to use, and what functions to call in them**!).

Please rename the starter.Rmd file to HW5_yourName.Rmd, and start to fill in code and notes. Load you
.Rmd into RStudio, and start adding code blocks (cells) plus annotations; you can run code in each cell block (marked by backticks at the top and at the bottom) by clicking on the little green arrow at the top right of the cell (that way you can develop and test just the code in that cell, as opposed to all the code
the notebook; that said, it is also possible to execute all the code in the notebook, from the topmost cell
the bottom - look in the options for the 'Run' button that is the top of the UI):

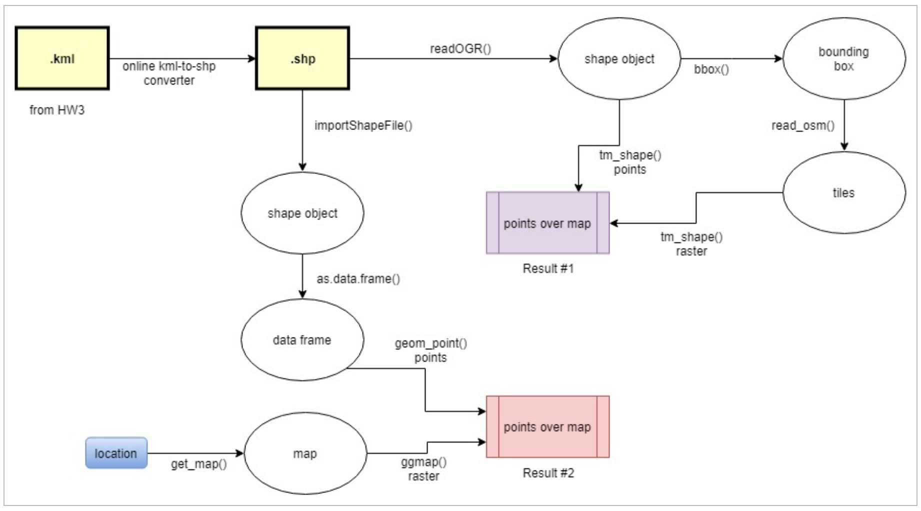


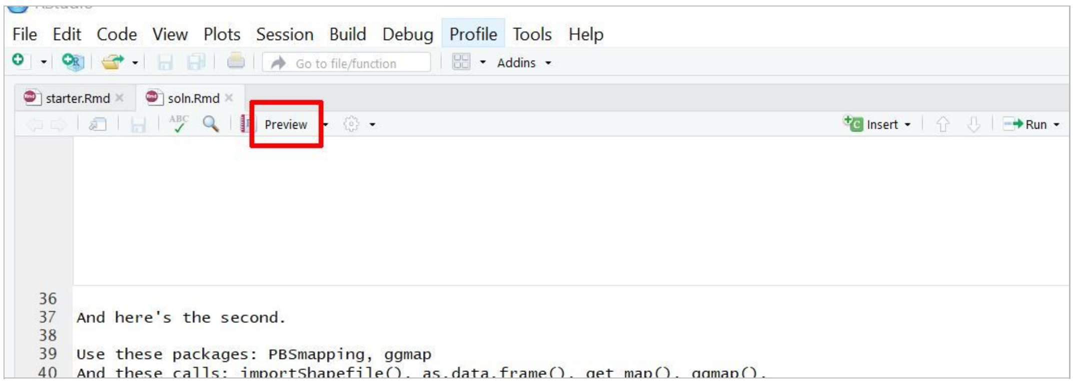
~~~
---
title: "HW5 notebook"
output: html_notebook
---

Here is the first method for overlaying shape data on a raster.

Use the following packages (install them first), then use them via
the library() call: tmap, tmaptools, rgdal, sp, osmdata

These calls are useful: readOGR(), tmaptools::read_osm(),
tm_shape(), tm_raster(), tm_polygons(), tm_dots()

1. readOGR() your shapefile, store it in a var
2. read_osm() the bbox of #1 above, store it in a var
3. do tm_shape() on #2 + tm_raster() +
tm_shape() on #1 + tm_dots() to specify point color and size

```{r}
# type code here, click on the green arrow at the top right of this block

~~~

And here's the second.

Use these packages: PBSmapping, ggmap
And these calls: importShapefile(), as.data.frame(), get_map(), ggmap(),
geom_point()

1. importShapefile() into a var
2. convert it (#1 above) to a dataframe, store it in a var
3. get_map() with a location that's in your collection of (lat,long),
store it in a var
4. do ggmap() on #3, + geom_point() on #2

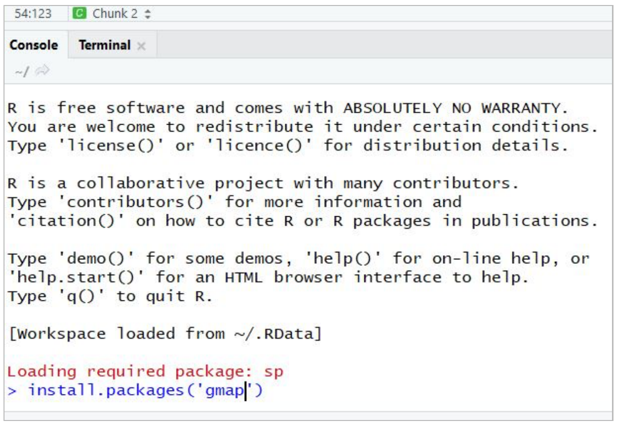
```{r}

```
```

Save your .Rmd (R Markdown) notebook file often, as you continue working on the two techniques. Consider creating little cells for experimenting, and annotating each with notes. Don't throw any code away (by overwriting it), in case you want to refer to it again later; instead, create a new cell, transfer co from existing cells if needed, and continue exploring. Here (https://rmarkdown.rstudio.com/articles_intro.html) is a nice intro' to R Markdown. Also, just fyi - simila 'notebooks' are used in Python (https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter Notebooks) as well.

Here is a **very helpful** diagram that shows how input location data is transformed via various calls, to produce the two required maps:



Notes:

- if a .nb.html doesn't appear automagically when you save your .Rmd, do 'File -> Knit Document' to generate it
- if your .nb.html isn't 'self-contained' (doesn't seem to upload to the Dropbox submission folder properly), do a 'Preview' (click on the Prev button) instead of doing 'Knit Document':
```


File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
starter.Rmd
soln.Rmd
Preview
Insert
Run

```
And here's the second.

Use these packages: PBSmapping, ggmap
And these calls: importShapefile(), as.data.frame(), get_map(), ggmap(),
```

• if Preview doesn't generate a self-contained .nb.html either, create a .zip file of the entire directory that contains the .nb.html, and uploac .zip file - do this ONLY if your .nb.html doesn't upload properly

• it is better to do 'install.packages()' via the shell/console, rather than in a code block [in other words, type install.packages('gmap'), for example, into the shell]:



```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

Loading required package: sp
> install.packages('gmap')
```

• if you're on Ubuntu, you might need to search around online a bit, for possible help with installing packages; eg. for rgdal, see https://philmikejones.wordpress.com/2014/07/14/installing-rgdal-in-r-on-linux/ (https://philmikejones.wordpress.com/2014/07/14/installing rgdal-in-r-on-linux/)

Here are R-specific pages to inspire/inform you [many contain code, illustrating function calls you'd use both parts of your HW (look at the starred ones in particular)]:

• https://cran.r-project.org/web/views/Spatial.html (https://cran.r-project.org/web/views/Spatial.html)

• http://spatial.ly/ (http://spatial.ly/)

• http://oscarperpinan.github.io/spacetime-vis/ (http://oscarperpinan.github.io/spacetime-vis/)

• https://geocompr.robinlovelace.net/spatial-class.html (https://geocompr.robinlovelace.net/spatial-class.html)

• https://github.com/Robinlovelace/Creating-maps-in-R (https://github.com/Robinlovelace/Creating-maps-in-R)

* https://cran.r-project.org/web/packages/tmap/vignettes/tmap-nutshell.html#plotting-with-tmap-elements (https://cran.r-project.org/web/packages/tmap/vignettes/tmap-nutshell.html#plotting-with-tmap-elements)

* https://stackoverflow.com/questions/2078468/plotting-shapefiles-on-top-of-google-map-tiles (https://stackoverflow.com/questions/2078468/plotting-shapefiles-on-top-of-google-map-tiles)

* https://www.nceas.ucsb.edu/~frazier/RSpatialGuides/ggmap/ggmapCheatsheet.pdf (https://www.nceas.ucsb.edu/~frazier/RSpatialGuides/ggmap/ggmapCheatsheet.pdf)

* https://www.nceas.ucsb.edu/scicomp/usecases/shapeFileToKML (https://www.nceas.ucsb.edu/scicomp/usecases/shapeFileToKML)

What to submit: your R notebook (.Rmd), and, a "rendered", publishable browser page (.nb.html) [or alternately, a .zip file instead of .nb.html].
