Lab 5. Visionmaking

Lab in Brief:
This lab will have you going public with your own datasets – choosing how to visualize them and how to use those datasets to support a particular argument in an online environment. You'll then create a number of different websites – using both R and an online program called visionmaker – to communicate these visualizations to a broader audience. While the focus of this lab will be on climate and climate change, the tools and procedures you will learn can be applied to any dataset.

Data and Materials:
We'll be using climate data from an online source made available by the FAO,
http://geonetwork3.fao.org/climpag/agroclimdb_en.php

Objectives:
In this lab, you will be applying all of the skills you've learned in previous labs to:
(A) Condense a very large dataset into meaningful, interactive visualizations.
(B) Put those visualizations online.
(C) Use these visualizations to argue that the area of the world you selection has (or has not) experienced imapcts from climate change to date.
(D) Finally, you'll be using an online program called visionmaker to leverage spatial modes of climate resilience in an online environment.

Deliverables:
By Friday, November 13th you will turn in the deliverables outlined on the final page of this lab.

Grading:
This lab will be graded by rubric, following the same grading strategy we've used for all of the labs to date. The rubric can be viewed on blackboard.

Part 1: Getting The Data and Cleaning it Up

The first steps of this lab will have you downloading data and cleaning it so that it's useful. For all of these examples I will use the United States, but you can download climate data from any country you are interested in. Keep in mind, though, that some countries will have very few weather stations and thus are not strong candidates for this lab.

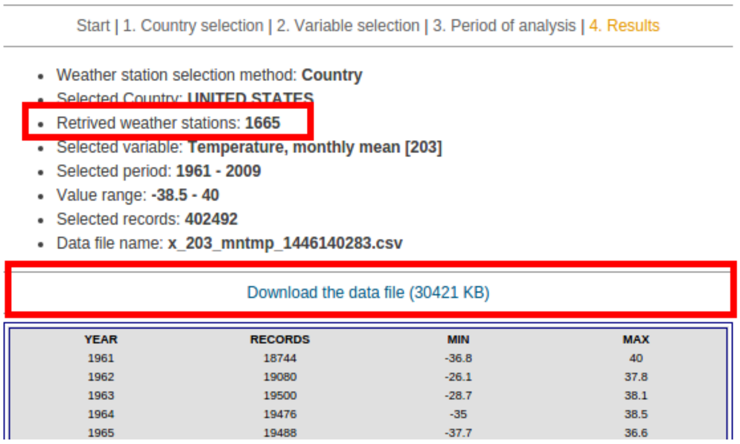
1. In your browser, go to **http://geonetwork3.fao.org/climpag/agroclimdb_en.php** , then click on "by country" towards the bottom of the screen (we want to pick out climate data for each country).

2. Choose the country you're interested in, make sure you have "time series" selected, then click continue.

3. Choose Temperature, Monthly mean, then click continue.

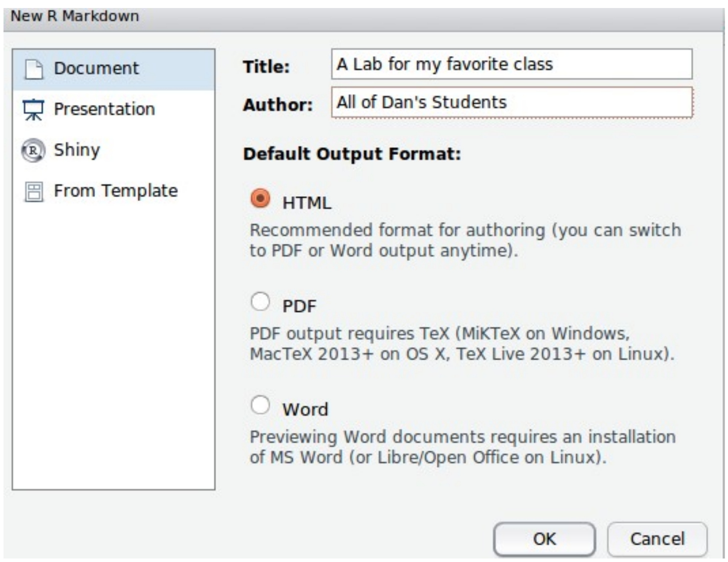
4. Check where it says "Available Weather Stations", and make sure you have at least 100 weather stations available to sample from. If you don't, you need to pick another country. If you do, you can continue - in the from and to fields, enter 1961 to 2009. We choose this date range as data is very sparse before 1961 from weather stations.

5. After a few minutes your final dataset will be ready for download – double check the retrieved weathers stations count is greater than 100, then click the download link and save the file into your H: drive (i.e., your R working directory). If you have less than 100 records, you need to select a different country.



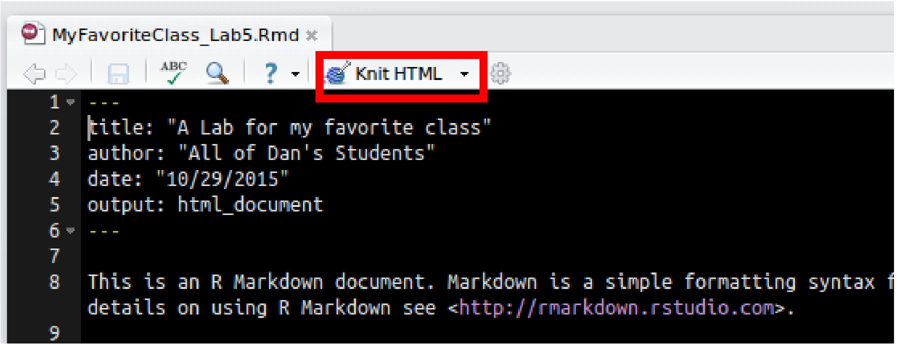
6. Open up R (I knew all along you were anxiously waiting for this step!!).

7. We're going to do something a bit different than before – instead of a normal R document, we're going to make something called an Rmarkdown document. These are very similar to normal R documents, but they let you put your content online. To do this, click on File → New File → Rmarkdown Document. In the wizard that pops up, you'll put a title and your name, but leave everything else at the defaults as shown below, then click OK.



8. Your default R document is now going to look a little crazy. It's OK! Click on File → Save, and save your field with the file extension "Rmd" in your H: working directory. For example, "MyFavoriteClass_Lab5.Rmd".

9. The advantage of Rmarkdown documents is they let you automatically build websites using your R code. To try this out, click on the "Knit HTML" button at the top of your screen and view the output. Later on, it will be very easy to publish these types of outputs to the web. On the window that pops up, you can click on "View in Browser" to see what it will look like online.



10. There are a couple of new things in Rmarkdown you need to know. First, is the ``` set of characters. This character (above tab to the far left of your keyboard, just under escape) is what Rmarkdown uses to define where R code lives in your document. So, for example, if you type in:

MyClimateData ← read.csv(x_203_mntmp_1446136697.csv)

R won't know what to do, and will just print out the text. Instead, if you type in:

```

*MyClimateData ← read.csv(x_203_mntmp_1446136697.csv)*

```

With the three "`" before and after your R code, R will be able to interpret the code correctly.

NOTE: If you run your R code by highlighting it all, you'll get a lot of error messges – you can safely ignore the ones that say "`Error: attempt to use zero-length variable name`", as those are just referring to R trying to interpret the "```" in your code. They are only interpreted correctly when you click the "Knit" button (which you'll learn more about below).

11. Alright, time to get started for real. Delete everything in your document and just put in the following lines:

```{r, echo=TRUE}

print("Hello, Web!")

```

Now, click the "Knit" button – you'll see both the R command you put in, as well as the return come out on the website. Now, change the echo=TRUE to echo=FALSE and knit it again. **Question 1-11:** what changed when you changed echo from TRUE to FALSE? Why?
```

12. We won't want to show all of our code on the website, as not all of it will be relevant for our argument. Let's start with a block of code that we don't show (i.e., echo=FALSE) and inside that block of code read the CSV you downloaded into R, i.e.:

````
```{r, echo=FALSE}
setwd("/home/dan/Desktop/GitRepo/COLL_100/labs/lab5_cga/Data")
ClimateData <- read.csv("x_203_mntmp_1446136697.csv")

```
````

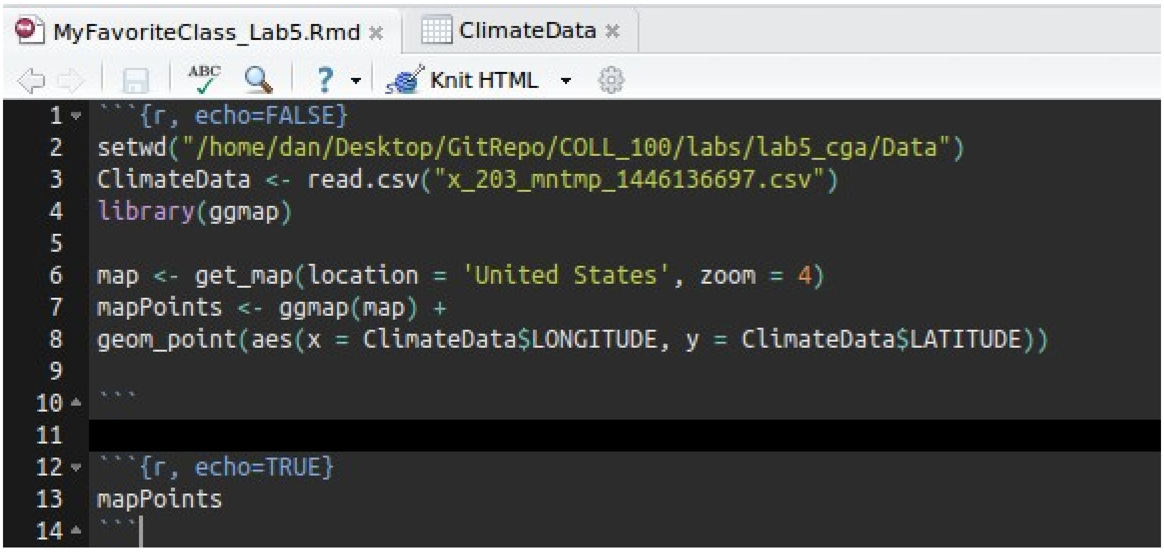
Now, click run on just your R lines – note your CSV will be different from mine! If you knit your HTML at this point, you shouldn't see anything (as echo = FALSE).

13. Let's make a map to visualize our data. We'll use a new library called “ggmap” to give our points some context – install it (install.packages()) and load it (library()). We'll want to do all of the code to make the map in our “echo=FALSE” part of the code.

14. Just like we did way back in lab 1, we're going to make a map with our geographic location and all of the weather station points on top of it. In the US example, this works:

```
map <- get_map(location = 'United States', zoom = 4)
mapPoints <- ggmap(map) +
geom_point(aes(x = ClimateData$LONGITUDE, y = ClimateData$LATITUDE))
```

To test if it worked for you, type in “mapPoints” into the console and hit enter – after a minute your map should come up. If it worked, add a new section of your code in which echo is equal to TRUE, and then just type in mapPoints. The code should look something like this:



15. Click the knit HTML button, and see if your map comes out on the web page. If it does, great! You're ready to move on to the next section. Take a screenshot or export your map and use it to answer question **1-15**.

Part 2: Building Interactive Figures

16. First, let's build a traditional climate graph – taking a look at what has happened to temperatures over time. First, we need to build what's called a “time series object” out of our data. To do that, we first need to formally declare our “Date” field as a “Date” - i.e., the computer doesn't know that a column of data represents time unless we tell it so:

```
ClimateData$DATE <- as.Date(as.character(ClimateData$DATE))
```

17. Now that we have that, we will build the time series itself – we're primarily interested in temperature for now, so that's all we'll pick out. First, install and load the library “xts”. Then:

```
#Aggregate to an average for every day in your time series
ClimateDailyAg <- aggregate(VALUE ~ DATE,
        data = ClimateData,
        FUN = mean)

#Create a time series formally (xts) by telling the computer the date and value
ClimateTimeSeries <- xts(ClimateDailyAg$VALUE, order.by = ClimateDailyAg$DATE)

#Aggregate to the yearly timesteps
ClimateTimeStep <- apply.yearly(ClimateTimeSeries,mean)
```

18. Let's take a quick look at what we just did – **question 2-18: copy and paste the figure you produce showing temperature over time:**

```
plot(ClimateTimeStep)
```

19. That's pretty handy, but we really want to make something even more dynamic. Install the package “dygraphs”, and then add the following code to a echo=TRUE section:

````
```{r, echo=TRUE}
library(dygraphs)
dygraph(ClimateTimeStep, main = "United States Temperatures") %>%
  dyRangeSelector(dateWindow = c("1960-01-01", "1990-01-01"))
```
````

20. Now, click “Knit HTML” - if you have a new, dynamic graph that shows up, you did it right and can move to the next step! You should now have a graph and a slider that shows you temperature fluctuation over time.

Part 3: Adding more Interactivity

21. Let's add a few more things – for this particular step, I'm just going to tell you the package and give you general instructions, rather than explicitly giving you the code. It will be up to you to design the implementation using the tools at your disposal (R – remember the “?” command for help!- or Google). The first thing we'll want is a data table (library DT) of the daily climate averages – make sure to put this in an ECHO=TRUE section of your code.

22. Now, we're going to try a few different types of visualizations using a new dataset – we've looked a lot at climate change, now we're going to consider some of the drivers of climate change. One key driver is car emissions – lets load up a database that will provide us with some insights into emissions. Type in:

CarData ← mtcars

In the above, *mtcars* is a default dataframe provided with R – we're simply renaming it to CarData for future use.

23. Let's visualize the data in a traditional way – type in. “wt” means weight, in 1000s of pounds. Let's see if there's a relationship between car weight and gas mileage.

```
plot(CarData$mpg, CarData$wt)
```

Q3-23 – Take a screenshot of this plot and include it in your final deliverable.

24. Now, let's make that interactive to add it to our website (put this in an echo=true section of your code) – we'll use the package metricsgraphics. After you install that package and load it, you'll type in something like this:

```
mjs_plot(CarData, x=wt, y=mpg) %>%
  mjs_point() %>%
  mjs_labs(x="Weight of Car", y="Miles per Gallon")
```

Q3-24: Change mjs_point in the above code to include “mjs_point(color_accessor=hp, size_accessor=hp)” . Create a new scatterplot, and then take a screenshot of that scatterplot to turn in. Write 1-4 sentences explaining why the scatterplot is different from the one you originally created (given that “hp” means “horsepower”).

25. Let's add some 3D interactivity to our site. Install the package “threejs”, then type in:

```
scatterplot3js(CarData$mpg,CarData$wt,CarData$hp)
```

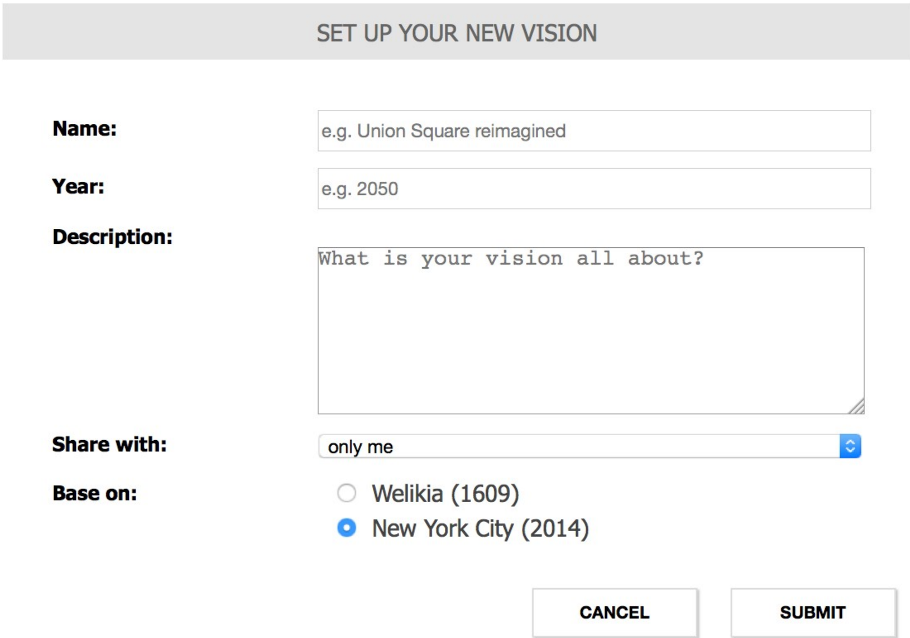
Add that to a echo=TRUE section, and then type in ?scatterplot3js. Using that help documentation, figure out how to remove the grid in the figure. Note you can click on the chart to move it around.

26. Finally, let's publish our website. At the top, click on “Knit HTML”, then on the preview that pops up click on “Publish”. **Sign up for a free account, and then provide your URL to answer question 3-26 below.** An example of what your output could look like can be found here: http://rpubs.com/geogdan/example_lab5 . Including additional figures, graphs, or analyses on this is a good way to “be creative” (...hint, hint, hint...).

Part 4: Modeling with Maps
In this final section of the lab, you'll be using an online tool called visionmaker to create new future visions of what land cover might look like to mitigate things like climate change. This is the same tool that our guest lecturer – Eric Sanderson – created to help illustrate the challenges associated with balancing tradeoffs between climate mitigation, biodiversity, and economic goals. You can learn more at https://welikia.org.

27. Goto to the New York City Vision maker website, create an account and login.
The website may have compatibility issues with some web browsers; use Firefox or Google Chrome if you experience problems. https://visionmaker.us/nyc/

28. Once you have created your account and logged into the Visionmaker portal, choose Create New Vision and set up your new vision by giving it a name. Under the Base on selection button, choose New York City (2014) as the basis for your new vision.



29. After creating your new vision, you will want to start by choosing a (small! You want to aim for no more than 3-4 city blocks) neighborhood somewhere in New York City. If you are familiar with New York City, then this will likely be simple for you. If you are not familiar with New York City, then try to think of some famous locations in NYC that come to your mind and consider those places. Following are
some suggested locations.

- Times Square
- The Empire State Building
- The Brooklyn Bridge
- Central Park
- Broadway
- Wall Street
- Yankee Stadium
- Harlem
- Rockefeller Center

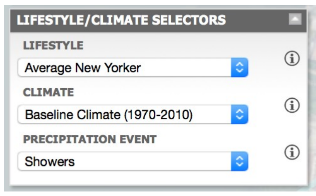
• Madison Square Garden
• JFK Airport

30. Once you have chosen your neighborhood, find it in the Visionmaker portal and zoom in towards that location. If you have problems finding it using the Visionmaker portal then use google to search your New York City neighborhood and then google maps to find the exact location; then use this as a guide to find your place of interest in the Visionmaker portal. On the right hand side of the Visionmaker portal is the zoom tool, use it to bring your neighbourhood into focus at zoom level 17.

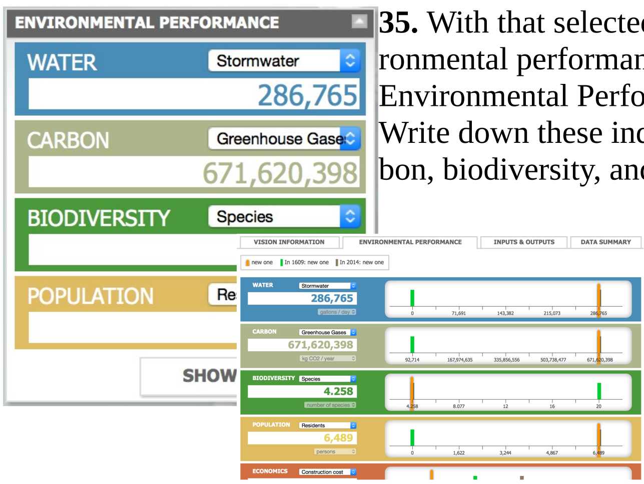
31. Once you have zoomed into your neighborhood, click on the Specify Vision Extent tool from the toolbar on the right side of the portal interface. Once the tool has been activated, blocks of land will highlight as the cursor traverses the aerial photograph of your New York City neighborhood. Use the Specify Vision Extent to select a few (no more than ~5) continuous blocks of parcels. Once you have selected the blocks of land that will comprise your neighborhood, click on the Save button on the top middle of the portal.

32. After you have saved the Vision Extent, your neighborhood will be transformed into individual pixels that represent the land use for that lo-cation. See the Ecosystem Tool Key at the end of this document to review the colors representing various Built and Natural Environment
land uses.

33. Select the Grid Inspector Tooland choose several of the grid cells within your neighborhood. Note the existing land use designations as well as the natural habitat for several grid cells.

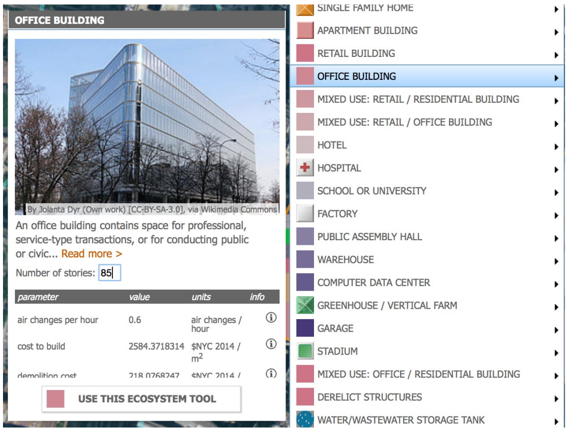


34. Now, find the Lifestyle/Climate Selectors toolbar and choose the lifestyle of the average person who will inhabit your neighborhood.



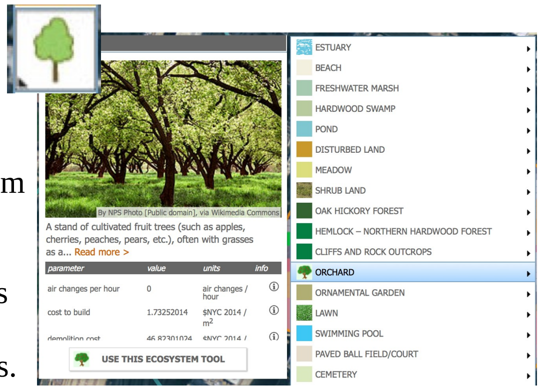
35. With that selected, we can now calculated your neighborhood's environmental performance in terms of water, carbon, biodiversity. Find the Environmental Performance toolbar, and choose the Calculate button. Write down these indicators. **Q 4-34:** What are your initial water, carbon, biodiversity, and population performance scores?

36. For your first exercise, make like you're the up-and-coming republican nominee, Mr. Trump himself. Find the Buildings button on the right hand side of the Visionmaker portal graphical interface. Choose one of the land uses under the buildings tab and start using that ecosystem tool to convert the existing uses to the new land use of your choice. Experiment with using the paint brush slider at the top of the page of the ecosystem tool to change the size. Your goal in this step is to convert each grid cell in your neighbourhood into its most profitable economic use. For example, transform low density residential uses to high density residential or commercial uses. Increase the number of stories of mixed use retail and office buildings and introduce hotels. Convert controlled access highways and freeways to parking decks, local streets and pedestrian pathways. Add powerplants and water treatment facilities if you feel those are necessary or simply leave them external to your neighborhood vision. Try to be realistic while also profit seeking. *Burn that environment to the ground!*



37. Once you are finished, recalculate the environmental performance of your neighborhood and report your new scored to answer question **4-37**. Write ~one paragraph detailing how you changed your neighborhood and what the impact was relative to your initial baseline. Include a screenshot of your revised vision.

38. Next imagine you are Michael Moore and transform your neighbourhood by maximizing environmental values. Go back to Step 4-36 in the Donald Trump scenario, but instead of pretending to convert land uses for maximum profit, let's build some trees. *Burn that economy to the ground!* Convert high density uses to natural land uses or introduce environmentally conscientious land uses such as solar energy electricity generation. Introduce forests, orchards, public parks or community recreational facilities. Introduce universities or public meeting spaces and reduce the functional classification of transportation facilities, introduce pedestrian pathways and other open or common spaces. Try to reduce carbon emissions, maximise species and habitat biodiversity ***while also maintaining population or permitting it to increase.***



39. Once you are finished, recalculate the environmental performance of your neighborhood and report your new scored to answer question **4-39**. Write ~one paragraph detailing how you changed your neighborhood and what the impact was relative to your initial baseline. Include a screenshot of your revised vision.

40. Finally, imagine you are New York City's Principal Land Use Planner and you need to administer the city's Comprehensive Plan, Zoning Ordinance and Development Regulations following the Policies and Objectives set forth by the City of New York Commission and Planning Councils (yep, those are all really things and really a job). Again, go back to Step 4-36 in the Donald Trump scenario, but this time you will need to meet several municipal, district and neighbourhood planning constraints, relative to your initial baseline calculated in **step 4-34.** Note it may be very difficult to achieve all of these aims without changing the lifestyle of individuals in the area you're looking at, but do your best!

- reduce greenhouse gas emissions by 15%
- increase species biodiversity by 15% and habitat biodiversity by 25%
- maintain the existing total population
- limit construction costs to 10 million USD

41. Once you are finished, recalculate the environmental performance of your neighborhood and report your new scored to answer question **4-41**. Write ~one paragraph detailing how you changed your neighborhood and what the impact was relative to your initial baseline. Include a screenshot of your revised vision.

Lab Deliverables:
(1) Answer the following questions:
Q1-11: What changed when you changed echo from TRUE to FALSE? Why?

Q1-15: Include the map of weather stations you produced.

Q2-18: Copy and paste the figure you produce showing temperature over time.

Q3-23 – Take a screenshot of this plot and include it in your final deliverable.

Q3-24: Change mjs_point in the above code to include "mjs_point(color_accessor=hp, size_accessor=hp)" . Create a new scatterplot, and then take a screenshot of that scatterplot to turn in. Write 1-4 sentences explaining why the scatterplot is different from the one you originally created (given that "hp" means "horsepower").

Q3-26: Provide the URL for your final KnitR output.

Q4-34: What are your initial water, carbon, biodiversity, and population performance scores?

Q4-37. Write ~one paragraph detailing how you changed your neighborhood and what the impact was relative to your initial baseline. Include a screenshot of your revised vision. (Economic Scenario)

Q4-39. Write ~one paragraph detailing how you changed your neighborhood and what the impact was relative to your initial baseline. Include a screenshot of your revised vision. (Environmental Scenario)

Q4-41. Write ~one paragraph detailing how you changed your neighborhood and what the impact was relative to your initial baseline, and how you met (or failed to meet) your goals. Include a screenshot of your revised vision. (Real-world Scenario)

Stretch Goals:
1) Improve your R website by adding basic titles and formatting to your website. Google "Rmarkdown" or "KnitR" and learn how to remove all R code from the website so it looks more natural.
2) Go back to the Social Networks lab and add the output figures you created to a new website. Provide the URL.