NR 592
Assignment 3

INSERT YOUR NAME HERE

INSERT THE DATE HERE

Last Updated: 2019-04-09

Instructions

Show your work. The response to each question should include all of the commands necessary to reproduce your results, and should be commented to give a naive reader an idea of the intent behind each section of code. Answers to questions should be clearly shown in the code output or spelled out in text or comments.

Before submitting your assignment, ensure that your code for each question works correctly when the lines are run in sequence, starting with an empty environment. You may submit either this R Markdown document with your code entered into the appropriate code blocks, or a regular R script with the question numbers as comments and the answers following. If you choose to Knit the R Markdown document to HTML or PDF, include those files with your submission (but it is not required that it Knit successfully).

Question 1

1.1

Go to the page for "Cherry blossom phenology and temperature reconstructions at Kyoto": http://atmenv.envi.osakafu-u.ac.jp/aono/kyophenotemp4/

Scroll down to the section labeled "Contents on Excel format (Newest Data)" and download the Excel files from the links labeled "Full-flowering data in Excel varsion" [sic] and "March mean temperature reconstructions (smoothed)." Place both files in your `/data` directory.

Load the flowering data into R as `cherryFlower`, designating empty strings (`""`) and hyphens (`"-"`) as missing values.

Load the temperature data as `cherryTemp`, designating `-50` as a missing value (Note: you may need to use `-50.00`).

If you have difficulty loading the Excel sheets into R directly, you may open them in Excel and save them as CSVs (without changing the contents), and read in the CSV files.

(Hint: Using an argument in your read function, you will need to skip the first several lines of each file that describe the data.)

```
# Load data
```

1.2

Examine the head, dimensions, and structure of each data frame.

1.3

For the `cherryFlower` data frame, look at the description of each variable in the original data set and check that any variables with categorical codes are represented as factors in your data frame. If they aren't, coerce them to factors.

If there appear to be any problems with the data (e.g. missing or extra rows or columns, incorrect variable names or values, improperly designated missing values, etc.), fix them before proceeding.

Question 2

2.1

Load the `dplyr` and `tidyr` packages (if you haven't loaded them already).

Combine `cherryFlower` and `cherryTemp` by the `AD` variable, so that you have one data frame named `cherry` with all of the variables and values from both data sets.

(Hint: Check the Data Wrangling Cheat Sheet.)

2.2

Examine the head, dimensions, and structure of `cherry`.

Print the data from the years (`AD`) 890 to 900 in all three data frames (`cherryFlower`, `cherryTemp`, and `cherry`) to check that the columns matched up successfully.

Question 3

3.1

Load the `ggplot2` package.

Using `ggplot()` and adding a `geom` layer, create a line plot of the estimated temperature by year.

Note: Pay attention to which variable should be assigned as the predictor (X) and which should be assigned as the response (Y).

3.2

Add the upper and lower limits of the 95% confidence interval for estimated temperature as a ribbon to your previous plot. Ensure the ribbon is plotted after (on top of) the line so that both are visible.

(Hint: If you're having trouble finding the limits, check the descriptions of the variables in the original data sets.)

3.3

Add two more lines to your plot to represent the temperature corrected for urban warming bias using Hikone weather station and temperature corrected using Kameoka weather station, each by year. Plot both of these before (below) your estimated temperature line.

3.4

Using themes and/or individual arguments to your functions and their aesthetic mappings, adjust any or all of the:

- line colors, types, or sizes,
- ribbon fill or transparency ("alpha"),
- panel color or fill,
- background color or fill,
- or, any additional changes not listed

... so that the lines and ribbon are all easily visible and distinguishable, your plot clearly communicates the information, and your plot is aesthetically pleasing overall. You do not need to add or change titles, labels, or legends; focus on the depiction of the data.

Briefly describe one change that you made as a result of the advice given by the required readings for this week.

Note: There are no specific criteria here, but you should make a few improvements to avoid the examples of what makes a figure *bad*. It doesn't need to be perfect.

`Delete this and enter your response here.`

Question 4

4.1

Using `ggplot()` and adding a `geom` layer, create a scatterplot of the flowering date (day of year) by estimated temperature. Color the points according to the data type code variable.

Note: Pay attention to which variable should be assigned as the predictor (X) and which should be assigned as the response (Y).

4.2

Use a different `geom` or an argument in your current `geom` to add a small amount of vertical jitter to the points. Set horizontal jitter to zero.

Increase the size of the points so that they are more easily visible but don't overlap too much.

4.3

Change the color scale of your points to one that is color-blind friendly, using any method you choose.

4.4

Use `geom_smooth()` to add a regression line and 95% confidence interval to your plot before (below) your points. Choose any method other than `"auto"` (the default). Define a color for the `geom` (such as `"black"` or `"#FFFFFF"`) so that it plots one line for all of the points rather than one line per level of `color`.

4.5

As in #3.4, make any changes you feel are necessary to improve the depiction of the data as if you were submitting this figure for publication. You must make at least one change, but do not need to explain any of them (beyond the usual requirement of commenting your code).

4.6 (Extra Credit: 1 point)

Assign sensible labels to your X axis, Y axis, and legend.

Add a plot title and cite the source of the data in a caption (using a function argument or another `geom`).

Change the values of each level of the data type code variable displayed in the legend to something short and descriptive.

(pdf / Rmd)