

MATHEMATICS E-151, Fall 2020
Classic Mathematics with a Modern User Interface
Hints for Problem 2 on the Class 2 Homework

It took me about an hour to get this all working.

1. Adding the new city

- The necessary controls fit neatly under the Add Route button.
- I added a function that takes a data frame, a city name, a latitude, and a longitude and returns a data frame with an extra row at the bottom. The trick is to append the new row as a list, since it contains a character string and two numbers. See `DataFrameGuide`. You can test this function in `spherical.R`.
- After you add the new city, you need to update the city selection inputs and redraw the map!

2. Coloring the Western Hemisphere cities red

- When you are making the matrix of coordinates to plot, also make a vector of color names and use `col = colors` in the function that plots the city names. You need to do this for both the 2D and 3D plots.

3. Plotting routes that are straight lines on the Mercator map.

- The function `sph.merc` converts a latitude in degrees to the y coordinate that is plotted on the Mercator map. Find a formula for the inverse function and test it in `spherical.R`.
- On the Mercator map the route is easy. Just draw a straight line between the cities. My `sph.plotRouteMerc` function is three lines long.
- On the 3D plot the route is harder to create.
First create a vector of $n = 50$ equally spaced x coordinates and n equally spaced y coordinates along the line on the Mercator map. Then, in a for loop, convert each pair of coordinates first to a latitude and longitude, then to a unit vector, which becomes a column of a matrix. Finally, call `plot3d` as before. Be careful about which values are in radians and which are in degrees.

4. You can try my solution at <https://bamberg.shinyapps.io/AirMileTest/>

5. I'll answer your questions in my 7:30 section tonight.