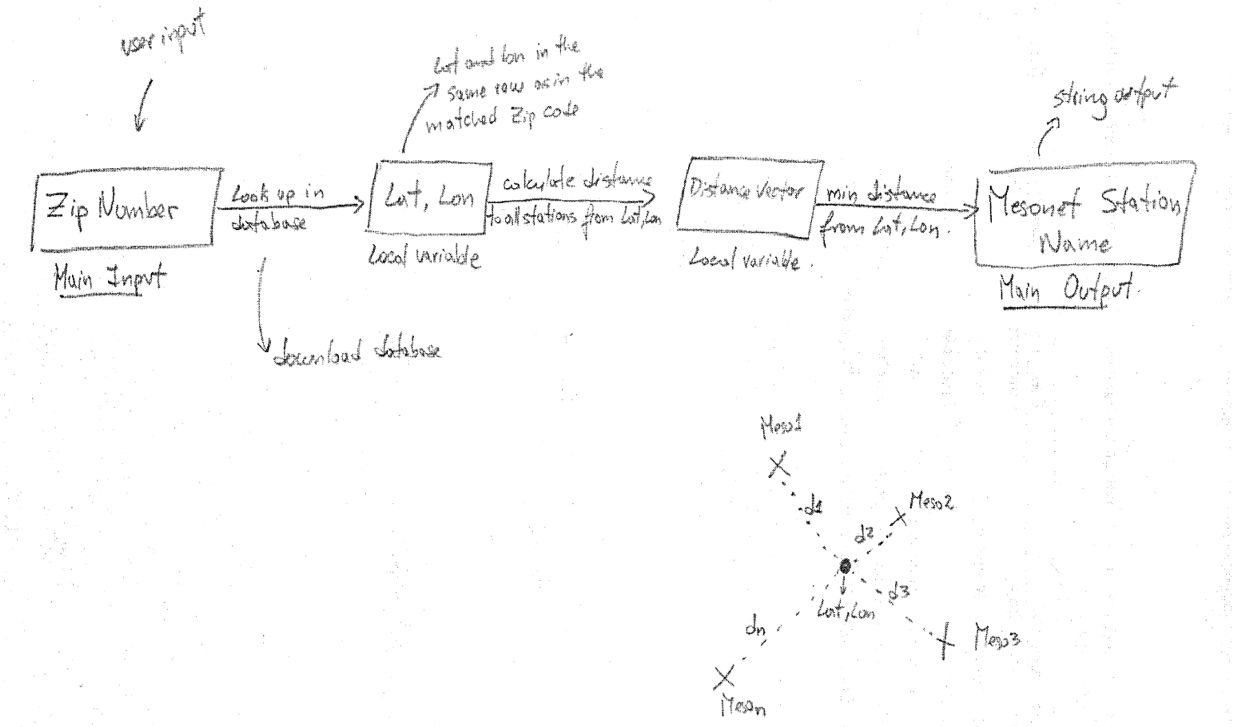
Case Study of the Coding Framework

Andres Patrignani and Tyson Ochsner
Soil Physics, Plant and Soil Sciences, Oklahoma State University

1. **Define objective of the code.**
   Objective: To create a function that determines the nearest Mesonet weather station for a given set of ZIP codes in the state of Oklahoma.

2. **Sketch the process (Prototyping and Breakdown).**



3. **Write a Pseudo-code**
   The user will type a ZIP code, which at this point is the only input. Then, the code should look up in a ZIP codes database (download a text file from the federal government website) for the entered ZIP code by the user. Save the row number of the ZIP code and retrieve the latitude (lat) and longitude (lon) which are in the same row but in different columns. Store lat and lon to be used in next step. Calculate the distance from the point with specified lat and lon to all Mesonet stations. Choose the minimum distance. Look up for the name of the station. Print station name on screen.

4. **Choose Programming Language.**

Since this function is intended only for internal use by the soil physics team, the use of Matlab programming language is appropriate.

5. Compile a list of variables and functions that may be necessary (Brainstorming).

Tentative Variables: ZIP code, ZIPLat, ZIPLon, MesonetStationLat, MesonetStationLon, ZIPcodeIndex, MesonetStationDistances

Tentative Functions:

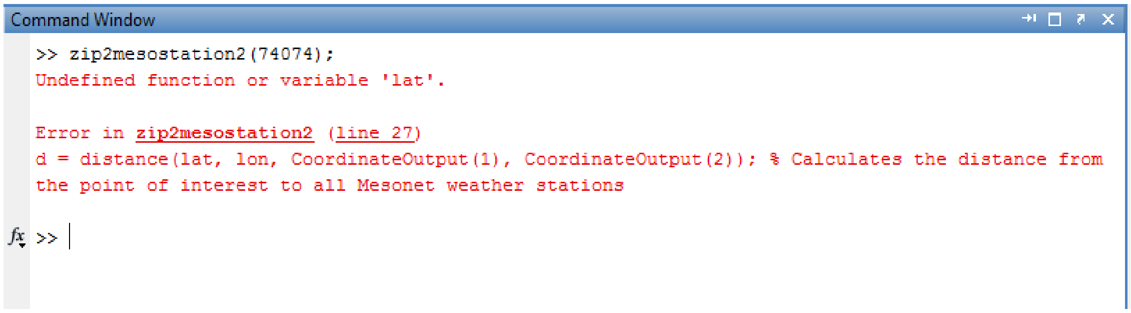
- textscan to read .dat or .txt file with zip codes, and their lat and lon coordinates.
- Likely to use find or logical indexing to find the row of the matching ZIP code, and so lookup the ZIPLat and ZIPLon.
- Shaperead to read a shapefile with all Mesonet stations and their lat and lon coordinates.
- min to detect the minimum distance from the ZIPLat and ZIPLon to all Mesonet stations.

6. **Draft the code.**

```
function [MesonetStation]=zip(ZIP_Code)
%ZIP Finds nearest Mesonet Station
%
% Function to retrieve latitude, longitude, and nearest Mesonet station
% from a given zip code in the state of Oklahoma.

% Author: Andres Patrignani
% Date: 5/10/2013 17:00 PM

%% Read the list of Zip codes
fid = fopen('zip_OK.dat');
C = textscan(fid, '%f32 %s %s %f32 %f32'); % Five columns in file
fclose('all');
ZIPList=cell2mat(C(1,1));

%% Get Lat and Lon for the specified ZIP Code
Coordinate=cell2mat(C(1,[4 5]));
CoordinateOutput=Coordinate(ZIPList==ZIP_Code,[1 2]);
ZIP_Lat=CoordinateOutput(1);
ZIP_Lon=CoordinateOutput(2);

%% Search for nearest Mesonet Station using Lat and Lon for the inserted ZIP
code.
stations = shaperead('ok_mesonet_sites_active_20120315.shp', 'UseGeoCoords',
true); % Read Mesonet active stations shapefile.

% Extract the point latitudes and longitudes with extractfield.
lats = extractfield(stations,'Lat');
lons = extractfield(stations,'Lon');

d = distance(lat, lon, CoordinateOutput(1), CoordinateOutput(2)); %
Calculates the distance from the point of interest to all Mesonet weather
stations
k = d == min(d(:)); % Obtains the nearest Mesonet station.

MesonetStation= stations(k,1).NAME;
```

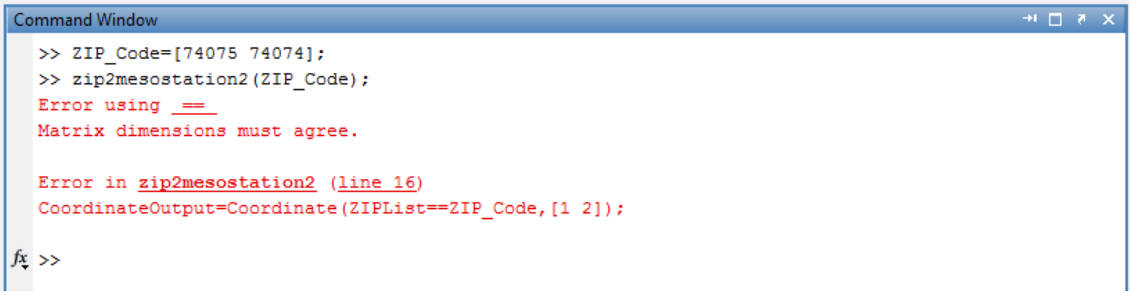
7. Error debugging

- In page 3 line 30 “`d = distance(lat, lon,…`” should say “`d = distance(lats, lons,…`”



```
>> zip2mesostation2(74074);
Undefined function or variable 'lat'.

Error in zip2mesostation2 (line 27)
d = distance(lat, lon, CoordinateOutput(1), CoordinateOutput(2)); % Calculates the distance from
the point of interest to all Mesonet weather stations

>>
```

- The function does not accept ZIP_Code as a vector. Therefore, a large data set of ZIP codes cannot be analyzed. This is critical since the function was intended for that purpose.



```
>> ZIP_Code=[74075 74074];
>> zip2mesostation2(ZIP_Code);
Error using  == 
Matrix dimensions must agree.

Error in zip2mesostation2 (line 16)
CoordinateOutput=Coordinate(ZIPList==ZIP_Code,[1 2]);

>>
```

8. Polishing the code

- Function name is not intuitive.
- Check for user input errors. Use the *isnumeric* function.
- Missing Input and Output description.
- Add improvements since the first version.
- Add examples.
- Add copyright statement.
- Time the function to check performance. Use “tic-toc” or “Run and Time” to check every step and detect slow parts of the code.

```
function[MesonetStation,ZIP_Lat,ZIP_Lon,Runtime]=zip2mesostation(ZIP_Code)
%ZIP2MESOSTATION Finds nearest Mesonet Station for a given ZIP code in OK.
%
% Function to retrieve latitude, longitude, and nearest Mesonet station
% from a given input zip code in the state of Oklahoma. The Mesonet is an
% environmental network of weather stations across Oklahoma. A total of
% 117 stations are currently active (July, 2013).
% Zip codes and their matching primary cities with their respective
% latitude and longitude were download from
% http://federalgovernmentzipcodes.us/
%
%% Improvements since version 1.0
% * Added user input error checkpoints.
% * |ZIP_Code| can be a vector.
% * Added |ZIP_Lat|, |ZIP_Lon|, |Runtime| as outputs.
% * Added example for Stillwater, OK (74074).
% * Show location in Google Earth.
%
%% Input
% |ZIP_Code| = US ZIP code format. The function only works for the state of
Oklahoma.
% The variable can be a scalar, vector
%
% "The values of Lat and Lon are provided to two decimal places or more.
% Since a degree of latitude is about 62 miles, one one-hundredth is .62
% miles, Similarlly, for the populated places in the US, a degree of
% latitudes is less than 65 miles. So the numeric values are a single point
% falls within a quite small area of about a mile for rounding errors." -
% Federal Government.
%
%% Outputs
% |MesonetStation| = Name of the nearest mesonet station.
%
% |ZIP_Lat| = Latitude in decimal degrees of the specified ZIP code.
%
% |ZIP_Lon| = Longitude in decimal degrees of the specified ZIP code.
%
% |Runtime| = Total time for function to perform a given search.
%
% Example
% Example for the city of Stillwater, OK with a zip code of 74074
%
```

```
% [MesonetStation, ZIP_Lat, ZIP_Lon]=zip2mesostation(74074);
%
% Output is:
%
%           Nearest Mesonet station for ZIP code 74074 is Stillwater
%
%           Runtime_zip2mesostation = 0.250290022431941

% Author: Andres Patrignani
% Date Created: 5/10/2013 17:00 PM
% Last Updated: 7/5/2013  13:00 PM
% Copyright (c) 2013, Andres Patrignani

%% Check for possible user errors
tic; % Start timing the function
if  isnumeric(ZIP_Code)==0 % Check for numeric input
    error('ZIP_Codes must be numeric') % If input is not numeric, then show
an error message.
end

%% Read the list of Zip codes.
fid = fopen('zip_OK.dat'); % Open file with list of zip codes.
C = textscan(fid, '%f32 %s %s %f32 %f32'); % Read Five columns in file (num,
string, string, num, num).
fclose('all'); % Close file.
ZIPList=cell2mat(C(1,1)); % Extract data into ZIPList for convenience.

%% Get Lat and Lon for the specified ZIP Code
Coordinate=cell2mat(C(1,[4 5])); % C is a cell array
ZIPIndex=arrayfun(@(x) find(ZIPList == x,1,'first'), ZIP_Code ); % arrayfun
applies a function to each element of array
CoordinateOutput=Coordinate(ZIPIndex,[1 2]);
ZIP_Lat=CoordinateOutput(:,1);
ZIP_Lon=CoordinateOutput(:,2);

%% Search for nearest Mesonet Station.
stations = shaperead('ok_mesonet_sites_active_20120315.shp', 'UseGeoCoords',
true); % Read Mesonet active stations shapefile.

%% Extract the Mesonet stations lat and lon.
lats = extractfield(stations,'Lat');
lons = extractfield(stations,'Lon');
MesonetStation = cell(numel(CoordinateOutput(:,1)),1); % Pre-allocating
memory
for n=1:numel(CoordinateOutput(:,1));
    d = distance(lats, lons, CoordinateOutput(n,1), CoordinateOutput(n,2)); %
Calculates the distance from the point of interest to all Mesonet weather
stations
    k = d == min(d(:)); % Obtains the nearest Mesonet station.

    % geoshow(stations(k).Lat, stations(k).Lon, ...
    %           'DisplayType', 'point', ...
    %           'Marker', 'o', ...
    %           'MarkerEdgeColor', 'k', ...
    %           'MarkerFaceColor', 'y', ...
    %           'MarkerSize', 3)
```

```
MesonetStation{n,1}= stations(k,1).NAME; % Mesonet Station name. Final Output.
      d=[];k=[]; %#ok<NASGU>
  end

  %% Show nearest city using Google Earth.
  if numel(ZIP_Code)==1   % Only execute code if |ZIP_Code| is a scalar.
      K = kml('ZIP_Code'); %create a kmltoolbox object
      f = K.createFolder('ZIP.point');
      f.point(ZIP_Lon,ZIP_Lat,0,'iconURL','wht-stars','iconScale',3); % Set
  height of the icon in the map, icon type, and icon size.
      K.run;
  end

  Runtime=toc; % Finish function timing.
```