MOVING WINDOW REGRESSIONS USING MATLAB

Spatial data has a lot of potential applications and can be a highly relevant tool for economists. While working with spatial data in the past, I have found moving window regressions a helpful tool to analyze data. MATLAB is has great functionality when working with spatial data and that's what I have used here.

So first and foremost, what is a moving window regression? It is a regression run across a subsample of the data. It is like disaggregating the estimates from an OLS regression, across space. Alternatively, we can think of it as running many local regressions, across space. It is useful when we think our regression coefficients vary across space. For example, the effect of rainfall on agricultural output may be different across different regions of a country. Using a moving window regression, we are able to identify these differences at a very nuanced level.

I am going to walk you through the steps of running a moving window regression using Australia's rainfall data . Moving window regressions require very granular spatial data and hence, is commonly used to analyze weather data (you get sufficient variation in 0.25 degree by 0.25 degree grid). For example, if you had city/ state level data, such a regression may not be ideal. This data set provides information on the monthly rainfall in mm during 1970-2007. The dimensions of the matrix "rainfall" are (lat lon year month). We will see the effect of the mean annual rainfall in June regressed on the mean annual rainfall in January.

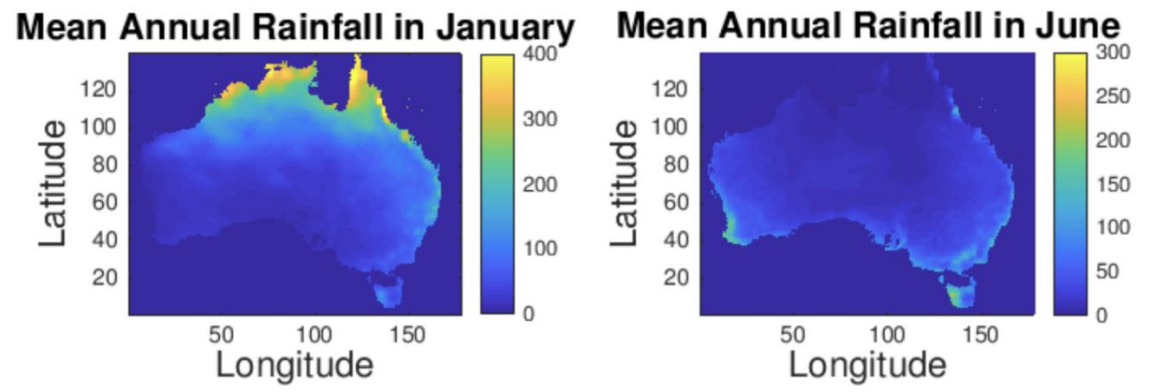
First, let us set the working directory and load the data set.

```
clear
cd('/MATLAB Drive/Published')
load lab6_data_AustraliaRainfall.mat
```

Second, we plot the mean annual rainfall in both January and June. We compute the mean annual rainfall by taking the means across the 3rd dimension (in this case, "year") of the rainfall array, since we want the mean across years.

```
%Plotting average rainfall for January and June
figure
subplot(1,2,1);
imagesc(mean(AUSrain.rainfall(:,:,:,1), 3, 'omitnan'));
axis xy; % flip the figure to north-south
axis equal; % tells matlab 1 degree lat equals 1 degree lon
axis tight;
title('Average Rainfall in January','fontsize',16);
xlabel('Longitude', 'FontSize', 16);
ylabel('Latitude', 'FontSize', 16);
caxis ([0 400]);
colorbar;
subplot(1,2,2);
imagesc(mean(AUSrain.rainfall(:,:,:,6), 3, 'omitnan'));
axis xy; % flip the figure to north-south
axis equal; % tells matlab 1 degree lat equals 1 degree lon
axis tight;
title('Average Rainfall in June','fontsize',16);
xlabel('Longitude', 'FontSize', 16);
ylabel('Latitude', 'FontSize', 16);
caxis ([0 300]);
colorbar;
box off;
print('-djpeg100','Jan_Jun_Avg_Rainfall');
```

[1] The dataset and methodology is from a Spatial Data class I took at the Goldman School of Public Policy at U.C. Berkeley that was taught by Professor Solomon Hsiang.



We can see from the figures above that the North gets on average more rainfall in January as compared to June. The west/center gets low rainfall in both January and June.

Now, the fun stuff - we run the moving window regression! The equation we want to estimate is:

$$Mean\ rainfall\ in\ June_{\ lon-s:lon+s,\ lat-s:lat+s} = \alpha_{\ lon,lat} + \beta_{\ lon,lat} * Mean\ rainfall\ in\ January_{\ lon-s:lon+s,\ lat-s:lat+s} + e_{\ lon-s:lon+s,\ lat-s:lat+s}$$

We estimate α and β for every point (lon, lat) on the raster. We assume a window of s = 5.

We choose a point (a,b) in the grid where rasters are located. We can think of a raster as a 2 dimensional grid. Using these rasters, we select the window around our point (a,b). The size of the window depends on how many pixels around the point (a,b) we want to consider in each local regression. We then reshape the data to be able to use it in an OLS regression. We then estimate a linear regression with a constant term using a subset of our data. For each subset, we extract a slope coefficient "beta". We save the beta corresponding to the point (a,b). Then, we repeat the above steps for every point in the raster. By varying the size of the window we can vary the area around (a,b) that we consider for each local regression. We use a window of size 5, which means we have ((5*2+1)^2) = 121 points in each local regression.

```
% Running the moving window regressions
%Create variables for the mean annual rainfall
Average_rainfall_Jan = mean(AUSrain.rainfall(:,:,:,1), 3, 'omitnan');
Average_rainfall_June = mean(AUSrain.rainfall(:,:,:,6), 3, 'omitnan');
R1= Average_rainfall_Jan;
R2 = Average_rainfall_June ;
% Size of window
s = 5 ;
obs = ((s*2)+1) * ((s*2)+1) ; % needed to reshape the data below
% Coefficients and correlation
beta = NaN(length(AUSrain.lat),length(AUSrain.lon)); % we will fill in the
betas below
% Regressions
for j = (s+1):(length(AUSrain.lon)-s) %account for edge effects here
for i = (s+1):(length(AUSrain.lat)-s)%and here
X = reshape(R1(i-s:i+s, j-s:j+s), obs, 1);
Y = reshape(R2(i-s:i+s, j-s:j+s), obs, 1);
coef = regress(Y, [ones(obs,1), X]);
beta(i,j) = coef(2,1); %saving the betas
end
end
%create a new dataset with a "lat", "lon" and corresponding "beta"
betafield = struct ('beta', beta, 'lat', AUSrain.lat, 'lon', AUSrain.lon);
```

Note, when you run the above code you may notice that you get an error that says "Warning: X is rank deficient to within machine precision". This means that, for some of the local regressions there isn't enough data to estimate the equation. Don't panic, this is simply because we include points in the ocean In the regression above.

In our dataset, there is no rainfall data for the oceans (ocean pixels have values of zero across months and years). So, the oceans have rainfall = 0 but we have calculated beta coefficients for ocean pixels too. So since we are regressing across zeroes, it gives the above error for all ocean points. Hence, we need to mask out the ocean so we don't falsely show a relationship between January and June rainfall for a pixel that is really just a zero. We do this below.

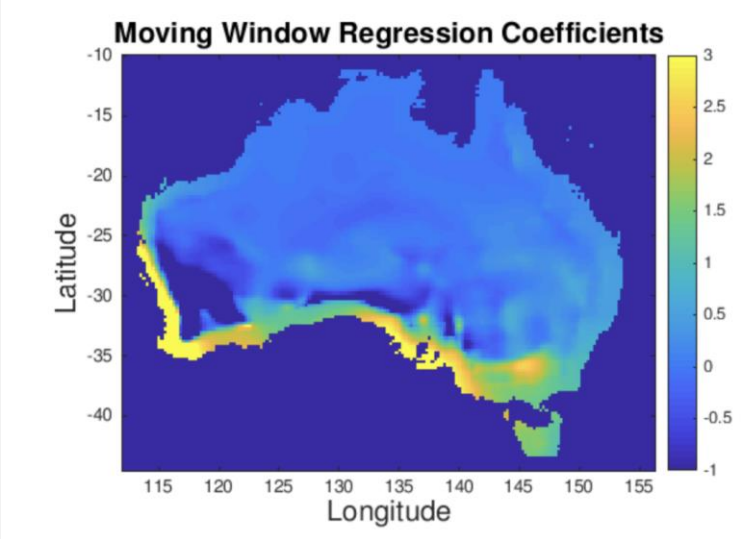
```
%create a new dataset with a "lat", "lon" and corresponding "beta"
field = struct ('beta', beta, 'lat', AUSrain.lat, 'lon', AUSrain.lon);

%mask out the oceans
%Create a variable summing the rainfall across months for any year
AUSrain.annual = sum(AUSrain.rainfall(:,:,1,:), 4);
for i = 1:length(AUSrain.lat)
   for j = 1:length(AUSrain.lon)
       if AUSrain.annual (i,j,:) ==0;
           land(i,j) =NaN;
       else
           land(i,j) =1;
       end
   end
end

field.betafinal = field.beta .* land;
```

Lastly, we plot the beta coefficients across space. Note, we only plot the beta coefficients for "land" where we have data. The points on the ratser which are ocean are coded as "NaN" and plotted as dark blue in the map below.

```
%plot beta coeff
figure;
imagesc(field.lon, field.lat, field.betafinal)
axis xy; axis equal; axis tight
title('Moving Window Regression Coefficients','fontsize',16);
xlabel('Longitude', 'FontSize', 16);
ylabel('Latitude', 'FontSize', 16);
caxis ([-1 3]);
colorbar ;
print('-djpeg100','Moving Window Regression Coefficients');
```



The final map shows us correlation between rainfall in January and June across years. We can interpret the regions in the south (high beta coefficients) as regions of low volatility in rainfall. The north in contrast has very little correlation between January and June rainfall and hence, has higher volatility in rainfall.

Sanghamitra W Mukherjee, DPhil Candidate in Economics, Somerville College, Oxford
20 May 2019