Preliminaries: Import the binary MATLAB file `bayAreaData.mat` into your workspace as `load BayAreaData.mat`; this file contains several arrays pertaining to precipitation and terrain elevation in the Bay Area.

1. Array `baydem` stores smoothed terrain elevation values (in m) derived from a Digital Elevation Model (DEM), and arranged in a single column (GeoEAS format). The raster data structure implicit in that single column has specifications: \([nx \ x_{\text{min}} \ x_{\text{size}}; \ ny \ y_{\text{min}} \ y_{\text{size}}]\), where \(nx\), \(xmin\) and \(xsize\) are, respectively, the # of grid nodes, the \(x\)-coordinate of the lower left grid node, and the grid node spacing, along the \(x\)-direction. Array `GridSpecs = [300 -123.5 0.0083; 360 36 0.0083]` contains the grid specification necessary to interpret and display the elevation values in the column vector `baydem`. Missing values are flagged as NaNs, and denote nodes whose elevation values are masked according to the entries of the array `baymask`.

Raster-based attribute values arranged in a GeoEAS-formatted column vector, such as `baydem`, can be displayed in a Cartesian coordinate system using the custom function `rastermap` as: `rastermap(GridSpecs,baydem,1,'cont')`, where `‘cont’` denotes a continuous variable for color scale purposes, and 1 denotes the column number of array `baydem` that is being displayed. Experiment with different options in `rastermap`, including the use of color limits, gray scale and color bar.

2. Array `bayprecip` stores the time-average (from Nov 01 1981 to Jan 31 1982) of daily precipitation in the Bay Area recorded at 77 rain gauges. More precisely, `bayprecip` consists of 4 columns with: #1 rain gauge number, #2 and #3 rain gauge longitude/latitude coordinates (in degrees), and #4 precipitation (in mm/day). Array `baycoast` contains 2 columns with longitude/latitude coordinates of points comprising a digitized coastline for the study region.

Display the sample precipitation data as: `scattermap(bayprecip,[2 3],30,4,'cont'); hold on; plot(baycoast(:,1),baycoast(:,2),'k-')`, where `[2 3]` and `4` are, respectively, the column numbers in array `bayprecip` for the coordinates and the variable, and `30` controls the size of the bullets. Expand the domain of the display to conform with that of the DEM as: `ax = [-123.5 -121 36 39]; axis(ax);` Compute the mean and variance of the sample precipitation data as: `meanPrecip = mean(bayprecip(:,4))` and `varPrecip = var(bayprecip(:,4))`; and store them for future reference.

3. Overlay on DEM elevation map the rain gauge sample locations as: `rastermap(GridSpecs,baydem,1,'cont'); hold on; plot(bayprecip(:,2),bayprecip(:,3),'w*'); hold off;` Comment qualitatively on the representativeness of the rain gauge data with respect to coverage of elevation regimes.
Nearest Neighbor & TIN Spatial Interpolation

We will be using the rain gauge precipitation data only (no covariates) to predict precipitation values at the grid nodes of the raster specified in array GridSpecs; this is the realm of spatial interpolation. Define: \( x = \text{bayprecip}(:,2), y = \text{bayprecip}(:,3) \), for future use.

**Nearest-neighbor spatial interpolation**

1. Describe briefly the concept of nearest neighbor spatial interpolation.

2. Predict (interpolate) precipitation values at the nodes of the regular grid specified in array GridSpecs as: \( \text{precipNN} = \text{nninterp}('\text{regular}', \text{GridSpecs}, \text{bayprecip}, [2 \ 3], 4) \); Display the interpolated precipitation values in a Cartesian coordinate system as: \( \text{rastermap}(	ext{GridSpecs}, \text{precipNN}, 1, '\text{cont}', [0 \ 15]) \), where [0 15] are color limits used for display. Compute the Voronoi polygons of the rain gauge locations, and overlay them on the map of interpolated values, as: \( \text{hold on, voronoi}(x,y,'w-') \), \( \text{hold off} \). Comment on the appearance of the precipitation surface, and on the role of the rain gauge configuration.

3. Use the Boolean land mask array baymask to set to NaN any interpolated value not in land as: \( \text{precipNN2} = \text{precipNN} \); \( \text{precipNN2(baymask==0)} = \text{NaN} \). Display the result using function \( \text{rastermap} \), and overlay on the interpolated precipitation map the coastline and the rain gauge measurements (for this last overlay, use function \( \text{scattermap} \) without any colorbar). What is implicitly assumed when masking the precipitation surface?

4. Use functions \( \text{nanmean} \) and \( \text{nanvar} \) from the Statistics Toolbox to compute the mean and variance of the interpolated precipitation values in array \( \text{precipNN2} \). Compare these statistics to the mean and variance of the 77 rain gauge data used for interpolation. Comment on the smoothing effect of estimation, and on any possible bias (if any).

**TIN-based spatial interpolation**

5. Describe briefly the concept of TIN-based spatial interpolation.

6. Use function \( \text{tininterp} \) to predict precipitation values at the nodes of the same grid as above as: \( \text{precipTIN} = \text{tininterp}('\text{regular}', \text{GridSpecs}, \text{bayprecip}, [2 \ 3], 4) \); Display the interpolated values using function \( \text{rastermap} \), using the same color limits \([0 \ 15]\) as before. Compute the Delaunay triangulation of the rain gauge locations as: \( \text{TRI} = \text{delaunay}(x,y) \); and overlay it on the map of interpolated values as: \( \text{hold on; triplot}(\text{TRI},x,y) \), \( \text{hold off} \). Comment qualitatively on the appearance of the resulting precipitation surface, and on the influence of the sample data configuration. Why are there empty spaces in the interpolated surface, i.e., grid nodes with no predictions (NaNs), without you specifying any land mask?

7. Compute the mean and variance of the interpolated precipitation values, and compare them to the mean and variance of the 77 rain gauge data that were used for interpolation. Comment on the smoothing effect of estimation, and on any possible bias (if any).
Inverse Distance Interpolation & Cross-Validation

1. Describe briefly the concept of inverse distance weighted spatial interpolation.

2. Consider the same regular grid specified by GridSpecs, and use function `idwinterp` (with global search) to interpolate precipitation values at the nodes of that grid. Consider the following 3 exponents: pow=1, pow=2 and pow=10, to produce 3 alternative sets of interpolated values (precipitation surfaces). For pow=1, for example, use: `precipIDW1 = idwinterp('regular', GridSpecs, bayprecip, [2 3], 4, 1)`. Note: You need to run function `idwinterp` 3 times, each time with a different exponent. Display the resulting 3 surfaces over land using function `rastermap`, and comment qualitatively on the appearance of the resulting 3 surfaces, the influence of the sample data configuration, and the influence of the particular exponent value used.

3. Use function `idwinterp` with exponent pow=2, but now with a local circular search neighborhood of radius 0.5 with no less than 4 and no more than 12 closest neighbors, specified as `srchSpecs = [0 0.5 0.5]` and `srchData = [4 12]`, to predict precipitation at the same grid nodes as before. Display the resulting interpolated surface using function `rastermap`, and comment on the appearance of the plot, the influence of the local search, and the influence of the sample data configuration.

Cross-Validation for evaluating interpolation performance

4. Consider the following automatic interpolation methods: (i) nearest neighbor interpolation, and (ii) inverse distance squared interpolation with a global search neighborhood (all sample data used for interpolation). Use functions `nninterp` and `idwinterp` to perform leave-one-out cross-validation with the N = 77 sample rain gauge data. NOTE: You need to toggle on the cross-validation option in those functions; for example: `precipNNcross = nninterp('crossval', [], bayprecip, [2 3], 4);` What are the prediction locations in this case? What do the different columns in the output of these two functions contain? (See the function help to answer this).

5. Display the cross-validation errors at the N = 77 rain gauges as, say: `scattermap(precipNNcross, [1 2], 30, 5, ’cont’), colorbar`. Comment qualitatively on the appearance of the resulting displays, particularly on the magnitude and spatial arrangement of these cross-validation errors. Based on the cross-validation error statistics, which of the two methods would you trust more for this particular case study, and why?

6. Optional: Describe in writing (no computations) a cross-validation-based means for choosing the “optimal” exponent parameter pow in function `idwinterp`. 