Model Comparison Script notes:

**time\_match function:**

Description: Matches temporal extent and resolution of two datasets.  
Input Data: Observational data and climate model data with the same temporal resolution

(hourly, daily, monthly, etc.) that have already been compressed down to one file using the Script4DataExtraction notebook or python script.

Output Data: Observational data and climate model data with matching time dimensions (same

start and end dates as well as same length).  
Process: (steps are numbered with in-line comments in code itself)

1. Matches the start times of the data by determining which dataset has the earlier first date and selecting only the data from that dataset beginning on the first date of the dataset with the later first date.
2. Matches the end times of the data by determining which dataset has the later last date and selecting only the data from that dataset from the beginning until the last date of the dataset with the earlier last date. *At this point both datasets should have the same start and end date but may still be different lengths if one of them includes leap days while the other does not.*
3. Address leapday differences checking if the length of the time dimension for both datasets is the same. If it is, the two datasets have been successfully time corrected and are renamed to reflect this new status. Otherwise, the length of the time dimension of the two datasets is not the same so the days that do not appear in the shorter of the two datasets are removed from the longer of the two datasets, presumably these are leap days. After this has been done the datasets are then successfully time-corrected and are renamed to reflect this new status.
4. The time-corrected datasets are returned by the time-matching function

**space\_match function:**

Description: Matches spatial extent and resolution of two datasets that already have the same

temporal extent and resolution.  
Input Data: Observational data and climate model data that has already been time-corrected after

being passed through the time-match function.  
Output Data: Observational data and climate data that have the same time extent, spatial extent,

and resolution, and are ready for analysis.  
Process:

1. Sort data by ascending latitude in preparation for matching latitudes.
2. Matches starting latitude by:
   * Determining which latitude in each dataset is closest to the lowest latitude in the other dataset, call this the starting latitude.
   * If the starting latitude of a given dataset is not the same as the lowest latitude in its own dataset, select only the data from the starting latitude upward in that dataset.
3. Matches ending latitude by:
   * Determining which latitude in each dataset is closest to the highest latitude in the other dataset, call this the ending latitude.
   * If the ending latitude of a given dataset is not the same as the last latitude in that dataset, select only the data from the first latitude in that dataset (as adjusted by step 1) until the ending latitude.
4. Matches starting longitude by:
   * Determining which longitude in each dataset is closest to the lowest lowest longitude in the other dataset, call this the starting longitude.
   * If the starting longitude of a given dataset is not the same as the lowest longitude in its own dataset, select only the data from the starting longitude upward in that dataset.
5. Matches ending longitude by:
   * Determining which longitude in each dataset is closest to the highest longitude in the other dataset, call this the ending longitude.
   * If the ending longitude of a given dataset is not the same as the last longitude in that dataset, select only the data from the first longitude in that dataset (as adjusted by step 5) until the ending longitude.
6. Addresses differences in spatial resolution by:
   * Checking if the length of the latitude variable is the same for each dataset.
   * If not, determine which dataset has lower resolution and select only the data from the higher resolution dataset that are at the grid points of the points for the lower resolution dataset. In places where there is not an exact grid point match, choose the nearest grid point.

**mask\_fun function:**

Description: Creates 4 new data files corresponding to the observational data with the ocean masked out (obs\_land), the observational data with the land masked out (obs\_ocean), the climate model data with the ocean masked out (clim\_land), and the climate model data with the land masked out (clim\_ocean).  
Input Data: Observational and climate model data that has already been corrected for differences in temporal and spatial extent and resolution using the time\_match() and space\_match() functions respectively.

Output Data:

1. obs\_land -- observational data with the ocean masked out (only contains land points, everything else is NaN)
2. obs\_ocean -- observational data with the land masked out (only contains ocean points, everything else is NaN)
3. clim\_land -- climate model data with the ocean masked out (only contains land points, everything else is NaN)
4. clim\_ocean -- climate model data with the land masked out (only contains ocean points, everything else is NaN)

Process:

1. Converts mask longitude coordinates to long1 to match observational and climate data for analysis.
2. Matches mask coordinates to data coordinates by:
   * creating empty mask array with same lat, lon dimensions as observational data (which should also be the same as for the climate model data by virtue of using the space\_match() function).
   * for each lat, lon point in the observational data (which should be the same as the lat, lon points in the climate model data), find the indices of the nearest latitude and longitude in the mask data and select the mask data point corresponding to those lat, lon indices.
3. Convert new matched mask back to xarray DataArray
4. Create separate masks for the ocean and the land by:
   * finding the coordinates where the coordinate-matched mask is zero (this corresponds to the ocean) and where it is one (this corresponds to the land).
   * creating new mask files for both masking out the ocean (ocean\_mask) and masking out the land (land\_mask) with the same dimensions as the coordinate-matched mask.
   * for the ocean mask, in every place where the coordinate-matched mask is 0, replace the corresponding point in the ocean mask with a NaN, in every place where the coordinate-matched mask is 1, replace the corresponding point in the ocean mask with 1.
   * for the land mask, in every place where the coordinate-matched mask is 0, replace the corresponding point in the ocean mask with 1, in every place where the coordinate-matched mask is 1, replace the corresponding point in the land mask with a NaN.
5. Create masked datasets for the observational and climate model data by:
   * For both the climate model data and the observational data, create 2 new arrays of zeros with the same dimensions as the original data (in total should have 4 new arrays of zeros).
   * for each time in the observational and climate model data, multiply the observational data by each mask separately (land\_mask and ocean\_mask), and store in the new arrays.
6. Return the masked datasets.

**plot\_compare function:**

Input Data: Observational data and climate model data that has already been corrected for differences in temporal and spatial extent and resolution using the time\_match() and space\_match() functions respectively.

Outputs:

* "AvgDifMap" -- Map plot of difference between the climate model and the observational data for the variable of interest averaged over the full temporal extent of the data.
* "TwoHist" -- Plot of histograms of data values for both the model and observational data.
* "DifHist" -- Histogram of the values of the difference between the climate model and observational data.
* Summary Statistics -- mean, standard deviation, min, max full variance, spatial variance, and temporal variance for each dataset as well as the mean, standard deviation, min, and max for the difference between the two datasets, as well as mean squared error, and root mean squared error.
* T-test -- test-statistic and p-value for one-sample t-test to determine if mean of difference is statistically different than 0.
* F-test -- test statistic and p-value for F-test to determine if variances of observational data and climate data are statistically different from each other for full data, spatial variance only, and temporal variance only.

Process:

1. Finds difference between model data and observations.
2. Plots difference averaged over all time by:
   * Checks if longitude coordinates are continuous.
   * If coordinates are continuous, plot on a regular projection.
   * If coordinates are discontinuous (i.e. crosses 180 lon without covering the whole globe), use a plot with 180 as the central longitude instead of 0.
3. Plots histogram of each distribution (climate model data and observational data).
4. Plots histogram of the differences between climate model data and observational data with a vertical line at zero.
5. Calculates summary statistics (mean , standard deviation, minimum, and maximum) for each of the observational data, climate model data, and their differences.
6. Performs a T-test to determine if the mean of the differences is statistically different than 0, indicating that the two populations (climate model data and observational data) have statistically different means.
7. Performs three F-tests to determine if the internal variability of each of the climate model and observational datasets are statistically different for:
   * the datasets as a whole
   * spatial variability only
   * temporal variability only
8. Calculates the mean squared error and root mean squared error.

**mod\_compare function:**

Description: Combines the time\_match(), space\_match(), mask\_fun(), and plot\_compare() functions to do all of the temporal and spatial extent and resolution matching as well as the masking and model comparison in one step.

Input Data: Observational data and climate model data with the same temporal resolution (hourly, daily, monthly, etc.) that have already been compressed down to one file using the Script4DataExtraction notebook or python script.

Outputs: Ascii file containing all of the statistical analyses for the model comparison.

Process:

1. Runs the data through each of the time\_match(), space\_match(), mask\_fun(), and plot\_compare() functions in order.
2. Compiles all of the outputs from plot\_compare() into a table with columns of the name of the statistic/statistical test, and the values of each for the full comparison, comparison over land only, and comparison over ocean only.
3. Writes the statistical table to an ascii file.