```
Help on module spin:

NAME
    spin

FILE
    /home/ishanu/Dropbox/ZED/Research/spin_/bin/spin.py

DESCRIPTION
    Spatio temporal analysis for inferrence of statistical causality
    @author zed.uchicago.edu

CLASSES
    spatioTemporal
    uNetworkModels

    class spatioTemporal
     |  Utilities for spatio temporal analysis
     |  @author zed.uchicago.edu
     |
     |  Attributes:
     |      log_store (Pickle): Pickle storage of class data & dataframes
     |      log_file (string): path to CSV of legacy dataframe
     |      ts_store (string): path to CSV containing most recent ts export
     |      DATE (string):
     |      EVENT (string): column label for category filter
     |      coord1 (string): first coordinate level type; is column name
     |      coord2 (string): second coordinate level type; is column name
     |      coord3 (string): third coordinate level type;
     |                       (z coordinate)
     |      end_date (datetime.date): upper bound of daterange
     |      freq (string): timeseries increments; e.g. D for date
     |      columns (list): list of column names to use;
     |          required at least 2 coordinates and event type
     |      types (list of strings): event type list of filters
     |      value_limits (tuple): boundaries (magnitude of event;
     |                            above threshold)
     |      grid (dict): dict with coord and eps (see example)
     |      threshold (float): significance threshold
     |
     |  Methods defined here:
     |
     |  __init__(self, log_store='log.p', log_file=None, ts_store=None, DATE='Da
te', year=None, month=None, day=None, EVENT='Primary Type', coord1='Latitude', c
oord2='Longitude', coord3=None, init_date=None, end_date=None, freq=None, column
s=None, types=None, value_limits=None, grid=None, threshold=None)
     |
     |  fit(self, grid=None, INIT=None, END=None, THRESHOLD=None, csvPREF='TS')
     |      Utilities for spatio temporal analysis
     |      @author zed.uchicago.edu
     |
     |      Fit dataproc with specified grid parameters and
     |      create timeseries for
     |      date boundaries specified by INIT, THRESHOLD,
     |      and END which do not have
     |      to match the arguments first input
     |      to the dataproc
     |
     |      Inputs:
     |          grid (pd.DataFrame): dataframe of location
```

```
|          timeseries data
|          INIT (datetime.date): starting timeseries date
|          END (datetime.date): ending timeseries date
|          THRESHOLD (float): significance threshold
|
|      Outputs:
|          (None)
|
|  getTS(self, _types=None, tile=None)
|      Utilities for spatio temporal analysis
|      @author zed.uchicago.edu
|
|      Utilities for spatio temporal analysis
|      @author zed.uchicago.edu
|
|      Given location tile boundaries and type category filter, creates the
|      corresponding timeseries as a pandas DataFrame
|      (Note: can reassign type filter, does not have to be the same one
|      as the one initialized to the dataproc)
|
|      Inputs:
|          _types (list of strings): list of category filters
|          tile (list of floats): location boundaries for tile
|
|      Outputs:
|          pd.Dataframe of timeseries data to corresponding grid tile
|          pd.DF index is stringified LAT/LON boundaries
|          with the type filter  included
|
|  pull(self, domain='data.cityofchicago.org', dataset_id='crimes', token='
ZIgqoPrBu0rsvhRr7WfjyPOzW', store=True, out_fname='pull_df.p', pull_all=False)
|      Utilities for spatio temporal analysis
|      @author zed.uchicago.edu
|
|      Pulls new entries from datasource
|      NOTE: should make flexible but for now use city of Chicago data
|
|      Input -
|          domain (string): Socrata database domain hosting data
|          dataset_id (string): dataset ID to pull
|          token (string): Socrata token for increased pull capacity
|          store (boolean): whether or not to write out new dataset
|          pull_all (boolean): pull complete dataset
|          instead of just updating
|
|      Output -
|          None (writes out files if store is True and modifies inplace)
|
|  timeseries(self, LAT, LON, EPS, _types, CSVfile='TS.csv', THRESHOLD=None
)
|      Utilities for spatio temporal analysis
|      @author zed.uchicago.edu
|
|      Creates DataFrame of location tiles and their
|      respective timeseries from
|      input datasource with
|      significance threshold THRESHOLD
|      latitude, longitude coordinate boundaries given by LAT, LON
|      calls on getTS for individual tile then concats them together
|
```

```
 |      Input:
 |          LAT (float):
 |          LON (float):
 |          EPS (float): coordinate increment ESP
 |          _types (list): event type filter; accepted event type list
 |          CSVfile (string): path to output file
 |
 |      Output:
 |          (None): grid pd.Dataframe written out as CSV file
 |                  to path specified

class uNetworkModels
 |  Utilities for storing and manipulating XPFSA models
 |  inferred by XGenESeSS
 |  @author zed.uchicago.edu
 |
 |  Attributes:
 |      jsonFile (string): path to json file containing models
 |
 |  Methods defined here:
 |
 |  __init__(self, jsonFILE)
 |
 |  augmentDistance(self)
 |      Utilities for storing and manipulating XPFSA models
 |      inferred by XGenESeSS
 |      @author zed.uchicago.edu
 |
 |      Calculates the distance between all models and stores
 |      them under the
 |      distance key of each model;
 |
 |      No I/O
 |
 |  select(self, var='gamma', n=None, reverse=False, store=None)
 |      Utilities for storing and manipulating XPFSA models
 |      inferred by XGenESeSS
 |      @author zed.uchicago.edu
 |
 |      Selects the N top models as ranked by var specified value
 |      (in reverse order if reverse is True)
 |
 |      Inputs -
 |          var (string): model parameter to rank by
 |          n (int): number of models to return
 |          reverse (boolean): return in ascending order (True)
 |              or descending (False) order
 |          store (string): name of file to store selection json
 |
 |      Returns -
 |          (dictionary): top n models as ranked by var
 |                        in ascending/descending order
 |
 |  to_json(outFile)
 |      Utilities for storing and manipulating XPFSA models
 |      inferred by XGenESeSS
 |      @author zed.uchicago.edu
 |
 |      Writes out updated models json to file
 |
```

```
 |      Input -
 |          outFile (string): name of outfile to write json to
 |
 |      Returns -
 |          Nonexs
 |
 |  ----------------------------------------------------------------------
 |  Data descriptors defined here:
 |
 |  models

FUNCTIONS
    draw_screen_poly(lats, lons, m, ax, val, cmap, ALPHA=0.6)
        utility function to draw polygons on basemap

    getalpha(arr, index, F=0.9)
        utility function to normalize transparency of quiver

    readTS(TSfile, csvNAME='TS1', BEG=None, END=None)
        Utilities for spatio temporal analysis
        @author zed.uchicago.edu

        Reads in output TS logfile into pd.DF
            and then outputs necessary
            CSV files in XgenESeSS-friendly format

        Input -
            TSfile (string): filename input TS to read
            csvNAME (string)
            BEG (string): start datetime
            END (string): end datetime

        Returns -
            dfts (pandas.DataFrame)

    showGlobalPlot(coords, ts=None, fsize=[14, 14], cmap='jet', m=None, figname=
'fig', F=2)
        plot global distribution of events
        within time period specified

        Inputs -
            coords (string): filename with coord list as lat1#lat2#lon1#lon2
            ts (string): time series filename with data in rows, space separated
            fsize (list):
            cmap (string):
            m (mpl.mpl_toolkits.Basemap): mpl instance for plotting
            figname (string): Name of the Plot

        Returns -
            m (mpl.mpl_toolkits.Basemap): mpl instance of heat map of
                crimes from fitted data

    splitTS(TSfile, csvNAME='TS1', dirname='./', prefix='@', BEG=None, END=None)
        Utilities for spatio temporal analysis
        @author zed.uchicago.edu

        Writes out each row of the pd.DataFrame as a separate CSVfile
        For XgenESeSS binary

        No I/O
```

```
    stringify(List)
        Utility function
        @author zed.uchicago.edu

        Converts list into string separated by dashes
                 or empty string if input list
                 is not list or is empty

        Input:
            List (list): input list to be converted

        Output:
            (string)

    to_json(pydict, outFile)
        Writes dictionary json to file
        @author zed.uchicago.edu

        Input -
            pydict (dict): ditionary to store
            outFile (string): name of outfile to write json to

        Returns -
            Nonexs

    viz(unet, jsonfile=False, colormap='autumn', res='c', drawpoly=False, fignam
e='fig')
          utility function to visualize spatio temporal
          interaction networks
          @author zed.uchicago.edu


        Inputs -
            unet (string): json filename
            unet (python dict):
            jsonfile (bool): True if unet is string  specifying json filename
            colormap (string): colormap
            res (string): 'c' or 'f'
            drawpoly (bool): if True draws transparent patch showing srcs
            figname  (string): prefix of pdf image file
        Returns -
            m (Basemap handle)
            fig (figure handle)
            ax (axis handle)
            cax (colorbar handle)

DATA
    __DEBUG__ = False
    __version__ = '0.31415'

VERSION
    0.31415
```