Scikit MIT Haystack Data Analysis Pipeline Toolkit

Generated by Doxygen 1.8.13

Contents

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 skdiscovery Namespace Reference

Namespaces

- framework
- generic
- series
- table
- utilities
- visualization

5.2 skdiscovery.framework Namespace Reference

Namespaces

- base
- discoverypipeline
- stagecontainers

5.3 skdiscovery.framework.base Namespace Reference

Classes

- class PipelineItem

  The general class used to create pipeline items.

5.4 skdiscovery.framework.discoverypipeline Namespace Reference

Classes

- class DiscoveryPipeline

  Pipeline for running the analysis.

5.5 skdiscovery.framework.stagecontainers Namespace Reference

Classes

- class StageContainer

  Container to hold a stage for the DiscoveryPipeline.
- class StageContainerAlternative

  Stage Container that holds a list of stage containers and randomly chooses one to use.
- class StageContainerIncrementalAdd

  In each perturb call, it incrementally adds one of the filters specified in the constructor.

5.6 skdiscovery.generic Namespace Reference

Namespaces

- accumulators

5.7 skdiscovery.generic.accumulators Namespace Reference

Namespaces

- data
- gpshplotter
- hcluster

5.8 skdiscovery.generic.accumulators.data Namespace Reference

Classes

- class DataAccumulator

  Stores a copy of the data in its current state in the pipeline.

5.9 skdiscovery.generic.accumulators.gpshplotter Namespace Reference

Classes

- class GPSHPlotter

  Plots results from General_Component_Analysis, for the GPS horizontal or vertical components.

5.10 skdiscovery.generic.accumulators.hcluster Namespace Reference

Classes

- class HCluster

  Hierarchical Clustering function that produces a cluster map of the distance matrix.

5.11 skdiscovery.series Namespace Reference

Namespaces

- accumulators
- analysis
- filters

5.12 skdiscovery.series.accumulators Namespace Reference

Namespaces

- plotter

5.13 skdiscovery.series.accumulators.plotter Namespace Reference

Classes

- class Plotter

  Make a plot of series data.

5.14 skdiscovery.series.analysis Namespace Reference

Namespaces

- correlate
- gca
- mogi

5.15 skdiscovery.series.analysis.correlate Namespace Reference

Classes

- class Correlate

  Computes the correlation for series data.

5.16 skdiscovery.series.analysis.gca Namespace Reference

Classes

- class General_Component_Analysis

  Performs either ICA or PCA analysis on series data.

5.17 skdiscovery.series.analysis.mogi Namespace Reference

Classes

- class Mogi_Inversion

  Perform a Mogi source inversion on a set of gps series data.

Functions

- def MogiVectors (mogi_res, station_lat_list, station_lon_list, flag3D=False)

  Creates a set of Mogi vectors for plotting.

5.17.1 Function Documentation

5.17.1.1 MogiVectors()

```
def skdiscovery.series.analysis.MogiVectors (
              mogi_res,
              station_lat_list,
              station_lon_list,
              flag3D = False )
```

Creates a set of Mogi vectors for plotting.

Parameters

mogi_res	Magma source inversion results
station_lat_list	List of station latitudes
station_lon_list	List of station longitudes
flag3D	Flag for generating 3 dimensional vectors instead of only horizontal

Returns

x and y Mogi vectors scaled by pca amplitude change

5.18 skdiscovery.series.filters Namespace Reference

Namespaces

- dataremover
- hyperbolictan
- interpolate
- kalman
- lowpass
- median
- offset_detrend
- trend

5.19 skdiscovery.series.filters.dataremover Namespace Reference

Classes

- class DataRemover

  Sets specified series data to NaN.

5.20 skdiscovery.series.filters.hyperbolictan Namespace Reference

Classes

- class HTanFilter

  Filter to subtract arctan fit from data.

5.21 skdiscovery.series.filters.interpolate Namespace Reference

Classes

- class InterpolateFilter

  Interpolate missing values on series data.

5.22 skdiscovery.series.filters.kalman Namespace Reference

Classes

- class KalmanFilter

  Runs a Kalman Smoother on series data.

5.23 skdiscovery.series.filters.lowpass Namespace Reference

Classes

- class LowPassFilter

  A FIR Remez (Parks-McLellan) designed low pass filter for series data.

5.24 skdiscovery.series.filters.median Namespace Reference

Classes

- class MedianFilter

  A Median filter for series data.

5.25 skdiscovery.series.filters.offset_detrend Namespace Reference

Classes

- class OffsetDetrend

  Trend filter that fits a stepwise function to linearly detrended series data.

5.26 skdiscovery.series.filters.trend Namespace Reference

Classes

- class TrendFilter

  Trend Filter that removes linear and sinusoidal (annual, semi-annual) trends on series data.

5.27 skdiscovery.table Namespace Reference

Namespaces

- accumulators
- analysis
- filters
- fusion
- generators

5.28 skdiscovery.table.accumulators Namespace Reference

Namespaces

- plotter

5.29 skdiscovery.table.accumulators.plotter Namespace Reference

Classes

- class Plotter

  Make a plot of table data.

5.30 skdiscovery.table.analysis Namespace Reference

Namespaces

- correlate
- dbscan
- gca
- midas
- mogi
- outlier
- skew

5.31 skdiscovery.table.analysis.correlate Namespace Reference

Classes

- class Correlate

  Computes the correlation for table data and stores the result as a matrix.

5.32 skdiscovery.table.analysis.dbscan Namespace Reference

Classes

- class DBScan
  Runs DBScan on table data.

5.33 skdiscovery.table.analysis.gca Namespace Reference

Classes

- class General_Component_Analysis
  Performs a general component analysis on table data.

5.34 skdiscovery.table.analysis.midas Namespace Reference

Classes

- class MIDAS
  In A basic MIDAS trend estimator.

5.35 skdiscovery.table.analysis.mogi Namespace Reference

Classes

- class Mogi_Inversion
  Perform a mogi source inversion on a set of gps table data.

Functions

- def MogiVectors (mogi_res, station_lat_list, station_lon_list, flag3D=False)
  Creates a set of mogi vectors for plotting.

5.35.1 Function Documentation

5.35.1.1 MogiVectors()

```
def skdiscovery.table.analysis.MogiVectors (
              mogi_res,
              station_lat_list,
              station_lon_list,
              flag3D = False )
```

Creates a set of mogi vectors for plotting.

Parameters

mogi_res	Magma source
station_lat_list	List of station latitudes
station_lon_list	List of station longitudes
flag3D	Make vectors 3 dimensional, defaults to False (2D)

5.36 skdiscovery.table.analysis.outlier Namespace Reference

Classes

- class Outlier

  Computes (data / mad(data)) for outlier detection.

5.37 skdiscovery.table.analysis.skew Namespace Reference

Classes

- class Skew

  Calculates the skew of table data.

5.38 skdiscovery.table.filters Namespace Reference

Namespaces

- antenna_offset
- calibrate_grace
- combine_columns
- dataremover
- geolocation
- hyperbolictan
- interpolate
- kalman
- lowpass
- median
- offset_detrend
- propagate_nans
- snow_remover
- stabilization
- table_filter
- trend
- weighted_average

5.39 skdiscovery.table.filters.antenna_offset Namespace Reference

Classes

- class AntennaOffset

  Applies corrections to fix offsets in PBO GPS data induced by antenna changes.

5.40 skdiscovery.table.filters.calibrate_grace Namespace Reference

Classes

- class CalibrateGRACE

  Calibrate Grace Data.

5.41 skdiscovery.table.filters.combine_columns Namespace Reference

Classes

- class CombineColumns

  Create a new column by selecting data from a column.

5.42 skdiscovery.table.filters.dataremover Namespace Reference

Classes

- class DataRemover

  Sets specified table data to NaN.

5.43 skdiscovery.table.filters.geolocation Namespace Reference

Classes

- class GeoLocationFilter

  Removes objects not located in a specified region.

5.44 skdiscovery.table.filters.hyperbolictan Namespace Reference

Classes

- class HTanFilter

  Filter to subtract an arctan fit from data.

5.45 skdiscovery.table.filters.interpolate Namespace Reference

Classes

- class InterpolateFilter

  Interpolate missing values on table data.

5.46 skdiscovery.table.filters.kalman Namespace Reference

Classes

- class KalmanFilter

  Runs a Kalman Smoother on table data.

5.47 skdiscovery.table.filters.lowpass Namespace Reference

Classes

- class LowPassFilter

  A remez low pass filter for table data.

5.48 skdiscovery.table.filters.median Namespace Reference

Classes

- class MedianFilter

  A Median filter for table data.

5.49 skdiscovery.table.filters.offset_detrend Namespace Reference

Classes

- class OffsetDetrend

  Trend filter that fits a stepwise function to linearly detrended table data.

5.50 skdiscovery.table.filters.propagate_nans Namespace Reference

Classes

- class PropagateNaNs

  Propagates NaN's from one column to other columns.

5.51 skdiscovery.table.filters.snow_remover Namespace Reference

Classes

- class SnowRemover

  Removes data with snow errors.

5.52 skdiscovery.table.filters.stabilization Namespace Reference

Classes

- class StabilizationFilter

  This filter transforms GPS stations in a region to a local reference frame.

5.53 skdiscovery.table.filters.table_filter Namespace Reference

Classes

- class TableFilter

  This class removes tables based on their label.

5.54 skdiscovery.table.filters.trend Namespace Reference

Classes

- class TrendFilter

  Trend Filter that removes linear and sinusoidal (annual, semi-annual) trends on series data.

5.55 skdiscovery.table.filters.weighted_average Namespace Reference

Classes

- class WeightedAverage

  This filter performs a rolling weighted average using standard deviations as weight.

5.56 skdiscovery.table.fusion Namespace Reference

Namespaces

- grace
- snow

5.57 skdiscovery.table.fusion.grace Namespace Reference

Classes

- class GraceFusion

  Fuses GRACE equivelent water depth time series.

5.58 skdiscovery.table.fusion.snow Namespace Reference

Classes

- class SnowFusion

  Adds snow time series data to table based on geographic coordinates.

5.59 skdiscovery.table.generators Namespace Reference

Namespaces

- catalog_generator
- data_generator

5.60 skdiscovery.table.generators.catalog_generator Namespace Reference

Classes

- class CatalogGenerator

  In Generates galaxy catalogs for use in DiscoveryPipeline.

5.61 skdiscovery.table.generators.data_generator Namespace Reference

Classes

- class DataGenerator

  In Class for generating random data.

5.62 skdiscovery.utilities Namespace Reference

Namespaces

- amazon_control
- amazon_gui
- astro_tools
- config
- kalman_smoother
- pbo_tools
- random_walks
- spherical_voronoi
- ssh_reverse
- trendTools

5.63 skdiscovery.utilities.amazon_control Namespace Reference

Functions

- def init (in_aws_access_key, in_aws_secret, in_aws_region, in_aws_security_group, in_aws_key_name, in_↩ pem_file)

  The underlying functionality for the Amazon GUI, the user should not need to directly interface with this function.
- def closeDispyScheduler ()

  Close the Dispy Scheduler.
- def startDispyScheduler ()

  Start the Dispy Scheduler.
- def generateInfo (instance)

  Read metadata from an Amazon instance.
- def updateStatus ()

  Update status information in amazon_list.
- def setNumInstances (new_total_instances, instance_type, image_id)

  Change the number of running instances.
- def createTunnels ()

  Create reverse ssh tunnels to all instances.
- def startDispyNode ()

  Start dispy on each Amazon instance.
- def resetInstances ()

  Reboot Amazon instances.
- def reset ()

  Close and clear Amazon List.
- def close ()

  Shutdown all instances, close dispy scheduler and clear Amazon list.
- def clearAmazonList ()

  Shutdown connection tunnels to Amazon instances and clear amazon list.

Variables

- aws_access_key = None
- aws_secret = None
- aws_region = None
- aws_security_group = None
- aws_key_name = None
- pem_file = None
- ec2_res = None
- ec2_client = None
- list amazon_list = [ ]
- scheduler = None
- popen = None

5.63.1 Function Documentation

5.63.1.1 clearAmazonList()

```
def skdiscovery.utilities.amazon_control.clearAmazonList ( )
```

Shutdown connection tunnels to Amazon instances and clear amazon list.

5.63.1.2 close()

```
def skdiscovery.utilities.amazon_control.close ( )
```

Shutdown all instances, close dispy scheduler and clear Amazon list.

5.63.1.3 closeDispyScheduler()

```
def skdiscovery.utilities.amazon_control.closeDispyScheduler ( )
```

Close the Dispy Scheduler.

5.63.1.4 createTunnels()

```
def skdiscovery.utilities.amazon_control.createTunnels ( )
```

Create reverse ssh tunnels to all instances.

5.63.1.5 generateInfo()

```
def skdiscovery.utilities.amazon_control.generateInfo (
              instance )
```

Read metadata from an Amazon instance.

Returns

metadata for Amazon instance

5.63.1.6 init()

```
def skdiscovery.utilities.amazon_control.init (
              in_aws_access_key,
              in_aws_secret,
              in_aws_region,
              in_aws_security_group,
              in_aws_key_name,
              in_pem_file )
```

The underlying functionality for the Amazon GUI, the user should not need to directly interface with this function.

Parameters

in_aws_access_key	AWS access key
in_aws_secret	AWS Secret Access Key
in_aws_region	AWS region (e.g. us-west-2)
in_aws_security_group	Security Group Name
in_aws_key_name	Name of Key Pair
in_pem_file	Filename of ssh key

5.63.1.7 reset()

```
def skdiscovery.utilities.amazon_control.reset ( )
```

Close and clear Amazon List.

5.63.1.8 resetInstances()

```
def skdiscovery.utilities.amazon_control.resetInstances ( )
```

Reboot Amazon instances.

5.63.1.9 setNumInstances()

```
def skdiscovery.utilities.amazon_control.setNumInstances (
              new_total_instances,
              instance_type,
              image_id )
```

Change the number of running instances.

Parameters

new_total_instances	New number of instances
instance_type	Instance type for new instances
image_id	ID of image (ami-xxxxxxxx)

5.63.1.10 startDispyNode()

```
def skdiscovery.utilities.amazon_control.startDispyNode ( )
```

Start dispy on each Amazon instance.

5.63.1.11 startDispyScheduler()

```
def skdiscovery.utilities.amazon_control.startDispyScheduler ( )
```

Start the Dispy Scheduler.

5.63.1.12 updateStatus()

```
def skdiscovery.utilities.amazon_control.updateStatus ( )
```

Update status information in amazon_list.

5.63.2 Variable Documentation

5.63.2.1 amazon_list

```
list skdiscovery.utilities.amazon_control.amazon_list = []
```

5.63.2.2 aws_access_key

```
skdiscovery.utilities.amazon_control.aws_access_key = None
```

5.63.2.3 aws_key_name

```
skdiscovery.utilities.amazon_control.aws_key_name = None
```

5.63.2.4 aws_region

```
skdiscovery.utilities.amazon_control.aws_region = None
```

5.63.2.5 aws_secret

```
skdiscovery.utilities.amazon_control.aws_secret = None
```

5.63.2.6 aws_security_group

```
skdiscovery.utilities.amazon_control.aws_security_group = None
```

5.63.2.7 ec2_client

```
skdiscovery.utilities.amazon_control.ec2_client = None
```

5.63.2.8 ec2_res

```
skdiscovery.utilities.amazon_control.ec2_res = None
```

5.63.2.9 pem_file

```
skdiscovery.utilities.amazon_control.pem_file = None
```

5.63.2.10 popen

```
skdiscovery.utilities.amazon_control.popen = None
```

5.63.2.11 scheduler

```
skdiscovery.utilities.amazon_control.scheduler = None
```

5.64 skdiscovery.utilities.amazon_gui Namespace Reference

Functions

- def init ()
  Initialize GUI for controlling Amazon instances.
- def drawGUI ()
  Draw the GUI on the screen.
- def changeButtonState (enabled=True)
  Enable or disable the buttons and slider in the GUI.
- def checkValidValues ()
  Check if Amazon information is valid.

Variables

- widget_dict = OrderedDict()
- list disable_list
- list key_value_list

5.64.1 Function Documentation

5.64.1.1 changeButtonState()

```
def skdiscovery.utilities.amazon_gui.changeButtonState (
              enabled = True )
```

Enable or disable the buttons and slider in the GUI.

Parameters

enabled	State to change the buttons to.

5.64.1.2 checkValidValues()

```
def skdiscovery.utilities.amazon_gui.checkValidValues ( )
```

Check if Amazon information is valid.

Returns

True if all AWS text fields have data in them, false otherwise

5.64.1.3 drawGUI()

```
def skdiscovery.utilities.amazon_gui.drawGUI ( )
```

Draw the GUI on the screen.

5.64.1.4 init()

```
def skdiscovery.utilities.amazon_gui.init ( )
```

Initialize GUI for controlling Amazon instances.

5.64.2 Variable Documentation

5.64.2.1 disable_list

```
list skdiscovery.utilities.amazon_gui.disable_list
```

Initial value:

```
1 =  ['execute_instances_button', 'initialize_button', 'cache_button', 'restore_button',
2                'new_num_instances_widget']
```

5.64.2.2 key_value_list

```
list skdiscovery.utilities.amazon_gui.key_value_list
```

Initial value:

```
1 =  ['aws_id_widget', 'aws_secret_widget', 'aws_region_widget','aws_security_widget',
2                  'aws_keyname_widget','aws_pem_widget','aws_image_id', 'instance_type_widget']
```

5.64.2.3 widget_dict

```
skdiscovery.utilities.amazon_gui.widget_dict = OrderedDict()
```

5.65 skdiscovery.utilities.astro_tools Namespace Reference

Functions

- def z_to_v (z)

  Convert redshift to km/s assuming shift is due to velocity using special relativity.
- def v_to_z (v)

  Convert km/s to redshift assuming all are using special relativity.
- def angular_separation (ra1, dec1, ra2, dec2)

  Angular seperation between two objects via the haversine formula.
- def move_point (ra, dec, ang_dist, bearing)

  Move a point along a great circle at a particular bearing.
- def abs_mag (app_mag, z)

  Get the absolute magnitude from apparent magnitude.
- def app_mag (abs_mag, z)

  Get the apparent magnitude from absolute magnitude.
- def nfw (R, norm_constant, Rs, Rcore)

  2D Navarro-Frenk-White surface radial profile probability density
- def lf (x, A, mstar, alpha)

  Schechter function.
- def dlf (x, A, m1, a1, m2, a2)

  double Schechter function.
- def cdf_dlf (x, A, m1, a1, m2, a2, start=-26)

  Cumulative Schechter function.
- def inv_cdf_dlf (p, A, m1, a1, m2, a2, start=-26, end=-15)

  Inverse Cumulative Schechter function.

5.65.1 Function Documentation

5.65.1.1 abs_mag()

```
def skdiscovery.utilities.astro_tools.abs_mag (
              app_mag,
              z )
```

Get the absolute magnitude from apparent magnitude.

Assumes concordance cosmology. No kcorrection is applied.

Parameters

app_mag	Apparent magnitude
z	Redshift

Returns

absolute magnitude of object at z

5.65.1.2 angular_separation()

```
def skdiscovery.utilities.astro_tools.angular_separation (
              ra1,
              dec1,
              ra2,
              dec2 )
```

Angular seperation between two objects via the haversine formula.

All inputs are in degrees.

Taken from http://www.movable-type.co.uk/scripts/gis-faq-5.1.html

Formula is presented in R.W. Sinnott, "Virtues of the Haversine", Sky and Telescope, vol. 68, no. 2, 1984, p. 159

Parameters

ra1	Right Ascention of first object (degrees)
dec1	Declination of first object (degrees)
ra2	Right Ascention of second object (degrees)
dec2	Declination of second object (degrees)

Returns

angular seperation between two objects

5.65.1.3 app_mag()

```
def skdiscovery.utilities.astro_tools.app_mag (
              abs_mag,
              z )
```

Get the apparent magnitude from absolute magnitude.

Assumes concordance cosmology. No kcorrection is assumed.

Parameters

abs_mag	Absolute magnitude
z	Redshift

Returns

apparent magnitude of object at z

5.65.1.4 cdf_dlf()

```
def skdiscovery.utilities.astro_tools.cdf_dlf (
              x,
              A,
              m1,
              a1,
              m2,
              a2,
              start = -26 )
```

Cumulative Schechter function.

Second LF is set to be 2∗A of first LF.

Parameters

x	magnitude
A	Scale factor
m1	Knee of distribution 1
a1	Faint-end turnover of first lf
m2	Knee of distribution 2
a2	Faint-end turnover of second lf
start	Brightest magnitude

Returns

Probability that galaxy has a magnitude greater than x

5.65.1.5 dlf()

```
def skdiscovery.utilities.astro_tools.dlf (
              x,
```

```
                A,
                m1,
                a1,
                m2,
                a2 )
```

double Schechter function.

Second LF is set to be 2∗A of first LF.

Parameters

x	magnitude
A	Scale factor
m1	Knee of distribution 1
a1	Faint-end turnover of first lf
m2	Knee of distribution 2
a2	Faint-end turnover of second lf

Returns

float: Double Schecter function at magnitude x

5.65.1.6 inv_cdf_dlf()

```
def skdiscovery.utilities.astro_tools.inv_cdf_dlf (
                p,
                A,
                m1,
                a1,
                m2,
                a2,
                start = -26,
                end = -15 )
```

Inverse Cumulative Schechter function.

Second LF is set to be 2∗A of first LF.

Parameters

p	probability
A	Scale factor
m1	Knee of distribution 1
a1	Faint-end turnover of first lf
m2	Knee of distribution 2
a2	Faint-end turnover of second lf
start	Brightest magnitude
end	Faintest possible magnitude

Returns

Magnitude associated with cdf probability p

5.65.1.7 lf()

```
def skdiscovery.utilities.astro_tools.lf (
              x,
              A,
              mstar,
              alpha )
```

Schechter function.

Parameters

x	magnitude
A	Scale factor
mstar	Knee of distribution
alpha	Faint-end turnover

Returns

float: Schecter function at magnitude x

5.65.1.8 move_point()

```
def skdiscovery.utilities.astro_tools.move_point (
              ra,
              dec,
              ang_dist,
              bearing )
```

Move a point along a great circle at a particular bearing.

All inputs are in degrees The formula was obtained from http://www.movable-type.co.uk/scripts/latlong.↩
html

Parameters

ra	Starting right ascension
dec	Starting declination
ang_dist	Angular distance to travel
bearing	Direction to travel (0 is north, 90 is positive RA)

Returns

tuple containing updated ra and dec

5.65.1.9 nfw()

```
def skdiscovery.utilities.astro_tools.nfw (
              R,
              norm_constant,
              Rs,
              Rcore )
```

2D Navarro-Frenk-White surface radial profile probability density

See

Navarro, J. F., Frenk, C. S., & White, S. D. M. 1996, ApJ, 462, 563 Bartelmann, M., A&A, 1996, 313, 697 Rykoff, E.S., et al., ApJ, 746, 178

Parameters

R	Radius
norm_constant	Normalization constant
Rs	Scale radius
Rcore	Since NFW profile diverges at R=0, the value at the center is held fixed starting at Rcore

Returns

probability density of profile at R

5.65.1.10 v_to_z()

```
def skdiscovery.utilities.astro_tools.v_to_z (
              v )
```

Convert km/s to redshift assuming all are using special relativity.

Parameters

v	velocity in km/s

Returns

Redshift of object with speed in km/s

5.65.1.11 z_to_v()

```
def skdiscovery.utilities.astro_tools.z_to_v (
              z )
```

Convert redshift to km/s assuming shift is due to velocity using special relativity.

Parameters

z	Redshift

Returns

speed in km/s assuming shift is due to motion using special relativity

5.66 skdiscovery.utilities.config Namespace Reference

Functions

- def getConfig ()
  Retrieve skdiscovery configuaration.
- def writeConfigValue (section, key, value)
  Write config to disk.
- def getDispyPassword ()
  Get dispy password.
- def getHostName ()
  Get Host name for displaying link to dispy status.

5.66.1 Function Documentation

5.66.1.1 getConfig()

```
def skdiscovery.utilities.config.getConfig ( )
```

Retrieve skdiscovery configuaration.

Returns

skdiscovery configparser

5.66.1.2 getDispyPassword()

```
def skdiscovery.utilities.config.getDispyPassword ( )
```

Get dispy password.

Returns

dispy password

5.66.1.3 getHostName()

```
def skdiscovery.utilities.config.getHostName ( )
```

Get Host name for displaying link to dispy status.

Returns

Hostname

5.66.1.4 writeConfigValue()

```
def skdiscovery.utilities.config.writeConfigValue (
              section,
              key,
              value )
```

Write config to disk.

Parameters

section	Name of section
key	Name of key
value	Value to write

5.67 skdiscovery.utilities.kalman_smoother Namespace Reference

Functions

- def KalmanFilter (in_data, t, sigma_sq, R, Pinit, x0=0, invert=False, clipping=5)

Runs the kalman filter on data.

- def FitFOGMParameters (data, Pinit=100, R=1, method='brute', x0=0, clipping=5)

  Find best FOGM parameters for a given data set.

- def IterativeGridSearch (f, args, intervals, max_iter=50, tol=0.1, bounds=None, prev_minimum=None, verbose=False)

  Find the minimum of f using an iterative grid search with 3 points per dimension.

- def KalmanSmoother (in_data, Pinit=1e6, Restimate=1, clipping=5, method='simple', t=None, sigma_sq=None, R=1, verbose=False, max_clip_iter=10)

  Smoother based on a forward and a backward kalman filter.

- def FOGM (size, t, sigma_sq, R)

  Generates data from a First Order Gaussian-Markov process.

5.67.1 Function Documentation

5.67.1.1 FitFOGMParameters()

```
def skdiscovery.utilities.kalman_smoother.FitFOGMParameters (
              data,
              Pinit = 100,
              R = 1,
              method = 'brute',
              x0 = 0,
              clipping = 5 )
```

Find best FOGM parameters for a given data set.

Parameters

data	input data
Pinit	Initial updated covariance
R	Noise Variance
method	Method used to fit FOGM parameters. Use "simple", "brute", or "igrid".
x0	Initial value of x0 to use in the kalman filter
clipping	Clipping factor used when computing cost functions

Returns

best fit correlation time
FOGM variance
Noise variance
correlation time from L
FOGM variance from Chat

5.67.1.2 FOGM()

```
def skdiscovery.utilities.kalman_smoother.FOGM (
              size,
              t,
              sigma_sq,
              R )
```

Generates data from a First Order Gaussian-Markov process.

Parameters

size	Number of data points
t	Correlation time
sigma_sq	FOGM variance
R	Measurement variance

Returns

Data generated from a FOGM

5.67.1.3 IterativeGridSearch()

```
def skdiscovery.utilities.kalman_smoother.IterativeGridSearch (
              f,
              args,
              intervals,
              max_iter = 50,
              tol = 0.1,
              bounds = None,
              prev_minimum = None,
              verbose = False )
```

Find the minimum of f using an iterative grid search with 3 points per dimension.

Parameters

f	Function to be minimized. The function must accept a tuple with coordinates for the first input.
args	additional arguments to pass on to the function.
intervals	Space that contains the minimum. Must be a list of tuples, even if only 1 dimension.
max_iter	Maximum number of iterations before stopping search.
tol	Error tolerance on result.
bounds	Additonal set of bounds for ending search.
prev_minimum	Previous minimum of function. If the current minimum is close to the previous minimum the serach will stop
verbose	Output debugging information.

Returns

A tuple containing a numpy array with the location of the minimum; and the minimum value of the function.

5.67.1.4 KalmanFilter()

```
def skdiscovery.utilities.kalman_smoother.KalmanFilter (
              in_data,
              t,
              sigma_sq,
              R,
              Pinit,
              x0 = 0,
              invert = False,
              clipping = 5 )
```

Runs the kalman filter on data.

Parameters

in_data	Input data
t	Correlation time
sigma_sq	FOGM variance
R	Noise variance
Pinit	Initial variance
x0	Intial updated state (default: 0)
invert	Run the filter backwards (boolean flag)
clipping	Clipping factor to use when computing cost functions

Returns

the predicted state
the predicted covariance
the updated state
the updated covariance
C_hat, the sample innovation variance
L, a different log variance cost function

5.67.1.5 KalmanSmoother()

```
def skdiscovery.utilities.kalman_smoother.KalmanSmoother (
              in_data,
              Pinit = 1e6,
```

```
Restimate = 1,
clipping = 5,
method = 'simple',
t = None,
sigma_sq = None,
R = 1,
verbose = False,
max_clip_iter = 10 )
```

Smoother based on a forward and a backward kalman filter.

Parameters

in_data	Data to be smoothed (must be in a Pandas DataFrame)
Pinit	Initial updated covariance
Restimate	Initial estimate for noise variance
clipping	Iteratively remove points beyond clipping $*$ MSE.
method	Method used to fit FOGM parameters. Use either "simple", "brute", or "igrid".
t	Fixed correlation time to use. Both sigma_sq and R must also be specified.
sigma_sq	Fixed sigma squared to use. Both t and R must also be specified.
R	Fixed measurement error to use Both t and sigma_sq must also be specified.
verbose	Output additional information.
max_clip_iter	Maximum number of clip iterations.

Returns

values smoothed by the kalman smoother
associated variance of smoothed result
t, same as input, might have been altered by fitting parameters
sigma_sq, same as input, might have been altered by fitting parameters
R, same as input, might have been altered by fitting parameters

5.68 skdiscovery.utilities.pbo_tools Namespace Reference

Functions

- def mogi (xdata, lat, lon, source_depth, amplitude)

  Compute the surface deformation due to changes in a mogi source.
- def finite_sphere (xdata, lat, lon, source_depth, amplitude, alpha_rad)

  Compute the surface deformation due to changes in a finite sphere source.
- def closed_pipe (xdata, lat, lon, source_depth, amplitude, pipe_delta)

  Compute the surface deformation due to changes in a closed pipe source.
- def constant_open_pipe (xdata, lat, lon, source_depth, amplitude, pipe_delta)

  Compute the surface deformation due to changes in a constant width open pipe source.
- def rising_open_pipe (xdata, lat, lon, source_depth, amplitude, pipe_delta, open_pipe_top)

  Compute the surface deformation due to changes in a rising width amplitude open pipe source.
- def sill (xdata, lat, lon, source_depth, amplitude)

Compute the surface deformation due to changes in a sill-like source.

- def dirEigenvectors (coord_list, pca_comps, pdir='H')

  Takes eigenvectors (north and east) and forces them to point "outward".

- def datetimeToNumber (in_time)

  Converts input pandas Timestamp or pandas DatetimeIndex to unix time.

5.68.1 Function Documentation

5.68.1.1 closed_pipe()

```
def skdiscovery.utilities.pbo_tools.closed_pipe (
              xdata,
              lat,
              lon,
              source_depth,
              amplitude,
              pipe_delta )
```

Compute the surface deformation due to changes in a closed pipe source.

For reference, see "Volcano Deformation", Dzurisin 2006, pg 292 (http://link.springer.com/book/10.↩1007/978-3-540-49302-0)

Parameters

xdata	List of the position data with each array element containing [direction (x, y, or z), lat, lon]
lat	Latitude of source
lon	Longitude of source
source_depth	Depth of source
amplitude	Ampltiude of source
pipe_delta	Pipe delta from source depth to top/bottom

Returns

list of resulting deformation for each point in xdata

5.68.1.2 constant_open_pipe()

```
def skdiscovery.utilities.pbo_tools.constant_open_pipe (
              xdata,
```

```
            lat,
            lon,
            source_depth,
            amplitude,
            pipe_delta )
```

Compute the surface deformation due to changes in a constant width open pipe source.

For reference, see "Volcano Deformation", Dzurisin 2006, pg 295 (http://link.springer.com/book/10.↩
1007/978-3-540-49302-0)

Parameters

xdata	List of the position data with each array element containing [direction (x, y, or z), lat, lon]
lat	Latitude of source
lon	Longitude of source
source_depth	Depth of source
amplitude	Ampltiude of source
pipe_delta	Pipe delta from source depth to top/bottom

Returns

list of resulting deformation for each point in xdata

5.68.1.3 datetimeToNumber()

```
def skdiscovery.utilities.pbo_tools.datetimeToNumber (
            in_time )
```

Converts input pandas Timestamp or pandas DatetimeIndex to unix time.

Parameters

in_time	Input pandas timestamp or pandas DatetimeIndex

Returns

unix time

5.68.1.4 dirEigenvectors()

```
def skdiscovery.utilities.pbo_tools.dirEigenvectors (
            coord_list,
```

```
                pca_comps,
                pdir = 'H' )
```

Takes eigenvectors (north and east) and forces them to point "outward".

Flips the sign of the projection if needed so that eigenvectors point outward. Needed because the "positive" direction for PCA is arbitrary

Parameters

coord_list	Location of stations for projecting the eigenvectors
pca_comps	PCA components
pdir	PCA direction, vertical or horizontal

Returns

station_lat_list: the station latitude coordinates
station_lon_list: the station longitude coordinates
ev_lat_list: the properly origented corresponding eigenvector latitude component
ev_lon_list: the properly origented corresponding eigenvector longitude component
direction scale factor (1 for no flip, or -1 for flip)

5.68.1.5 finite_sphere()

```
def skdiscovery.utilities.pbo_tools.finite_sphere (
                xdata,
                lat,
                lon,
                source_depth,
                amplitude,
                alpha_rad )
```

Compute the surface deformation due to changes in a finite sphere source.

For reference, see "Volcano Deformation", Dzurisin 2006, pg 290 (http://link.springer.com/book/10.↩1007/978-3-540-49302-0)

Parameters

xdata	List of the position data with each array element containing [direction (x, y, or z), lat, lon]
lat	Latitude of source
lon	Longitude of source
source_depth	Depth of source
amplitude	Ampltiude of source
alpha_rad	Alpha radius of the source

Returns

list of resulting deformation for each point in xdata

5.68.1.6 mogi()

```
def skdiscovery.utilities.pbo_tools.mogi (
              xdata,
              lat,
              lon,
              source_depth,
              amplitude )
```

Compute the surface deformation due to changes in a mogi source.

Parameters

xdata	List of the position data with each array element containing [direction (x, y, or z), lat, lon]
lat	Latitude of source
lon	Longitude of source
source_depth	Depth of source
amplitude	Amplitude of mogi source

Returns

list of resulting deformation for each point in xdata

5.68.1.7 rising_open_pipe()

```
def skdiscovery.utilities.pbo_tools.rising_open_pipe (
              xdata,
              lat,
              lon,
              source_depth,
              amplitude,
              pipe_delta,
              open_pipe_top )
```

Compute the surface deformation due to changes in a rising width amplitude open pipe source.

For reference, see "Volcano Deformation", Dzurisin 2006, pg 295 (http://link.springer.com/book/10.↩1007/978-3-540-49302-0)

Parameters

xdata	List of the position data with each array element containing [direction (x, y, or z), lat, lon]
lat	Latitude of source
lon	Longitude of source
source_depth	Depth of source
amplitude	Ampltiude of source
pipe_delta	Pipe delta from source depth to top/bottom
open_pipe_top	Depth of the top of the open pipe

Returns

list of resulting deformation for each point in xdata

5.68.1.8 sill()

```
def skdiscovery.utilities.pbo_tools.sill (
              xdata,
              lat,
              lon,
              source_depth,
              amplitude )
```

Compute the surface deformation due to changes in a sill-like source.

For reference, see "Volcano Deformation", Dzurisin 2006, pg 297 (http://link.springer.com/book/10.↩1007/978-3-540-49302-0)

Parameters

xdata	List of the position data with each array element containing [direction (x, y, or z), lat, lon]
lat	Latitude of source
lon	Longitude of source
source_depth	Depth of source
amplitude	Ampltiude of source

Returns

list of resulting deformation for each point in xdata

5.69 skdiscovery.utilities.random_walks Namespace Reference

Functions

- def uniform_walk (pos, grid, step_size=None)

  A uniform random walk function.
- def gaussian_walk (pos, grid, step_size=None)

  A gaussian random walk function.
- def keep_in_bound (pos, grid)

  Function for truncating and bounding the random walk to within the defined grid.

5.69.1 Function Documentation

5.69.1.1 gaussian_walk()

```
def skdiscovery.utilities.random_walks.gaussian_walk (
              pos,
              grid,
              step_size = None )
```

A gaussian random walk function.

Parameters

pos	tuple of input point
grid	bounds for walk
step_size	maximal step size

Returns

position tuple

5.69.1.2 keep_in_bound()

```
def skdiscovery.utilities.random_walks.keep_in_bound (
              pos,
              grid )
```

Function for truncating and bounding the random walk to within the defined grid.

Parameters

pos	tuple of the point to be checked
grid	the bounds for limiting the walk

Returns

position tuple after bounding the point

5.69.1.3 uniform_walk()

```
def skdiscovery.utilities.random_walks.uniform_walk (
              pos,
              grid,
              step_size = None )
```

A uniform random walk function.

Parameters

pos	tuple of input point
grid	bounds for walk
step_size	maximal step size

Returns

position tuple

5.70 skdiscovery.utilities.spherical_voronoi Namespace Reference

Functions

- def sphericalToXYZ (lat, lon, radius=1)

  Convert spherical coordinates to x,y,z.
- def xyzToSpherical (x, y, z)

  Convert x,y,z to spherical coordinates.
- def find_match (region_index, region_list)

  Find neighboring regions.
- def getVoronoiCollection (data, lat_name, lon_name, bmap=None, v_name=None, full_sphere=False, max_v=.3, min_v=-0.3, cmap=matplotlib.cm.get_cmap('jet'))

  Perform a Spherical Voronoi Tessellation on the input data.

5.70.1 Function Documentation

5.70.1.1 find_match()

```
def skdiscovery.utilities.spherical_voronoi.find_match (
              region_index,
              region_list )
```

Find neighboring regions.

Parameters

region_index	Numeric index of region to find matches for (number between 0 and len(vertices))
region_list	list of lists of vertices that define regions

Returns

Numeric indices of regions that border the region specified by region_index

5.70.1.2 getVoronoiCollection()

```
def skdiscovery.utilities.spherical_voronoi.getVoronoiCollection (
              data,
              lat_name,
              lon_name,
              bmap = None,
              v_name = None,
              full_sphere = False,
              max_v = .3,
              min_v = -0.3,
              cmap = matplotlib.cm.get_cmap('jet') )
```

Perform a Spherical Voronoi Tessellation on the input data.

In the case where the data is restricted to one part of the globe, a polygon will not be returned for all objects, as matplotlib polygons won't be able to stretch over half the globe.

Parameters

data	Input pandas data frame
lat_name	Name of latitude column
lon_name	Name of longitude column
bmap	Basemap instance used to convert from lat, lon coordinates to projection coordinates
v_name	Name of value column. Use this to color each cell according to a value.
full_sphere	Set to true if the data spans the entire globe. If false, a fictional point is created during tessellation and removed later to work around issues when polygons are suppose to span the over half the globe.
max_v	Specify a maximum value to use when assigning values to the tessellation
min_v	Specify a minimum value to use when assigning values to the tessellation
cmap	Matplotlib color map to use

Returns

Matplotlib patch collection of tessellation, scipy.spatial.SphericalVoronoi object, integer index of objects in patch collection.

5.70.1.3 sphericalToXYZ()

```
def skdiscovery.utilities.spherical_voronoi.sphericalToXYZ (
              lat,
              lon,
              radius = 1 )
```

Convert spherical coordinates to x,y,z.

Parameters

lat	Latitude, scalar or array
lon	Longitude, scalar or array
radius	Sphere's radius

Returns

Numpy array of x,y,z coordinates

5.70.1.4 xyzToSpherical()

```
def skdiscovery.utilities.spherical_voronoi.xyzToSpherical (
              x,
              y,
              z )
```

Convert x,y,z to spherical coordinates.

Parameters

x	Cartesian coordinate x
y	Cartesian coordinate y
z	Cartesian coordinate z

Returns

numpy array of latitude,longitude, and radius

5.71 skdiscovery.utilities.ssh_reverse Namespace Reference

Classes

- class ReverseTunnel

  Create a reverse ssh tunnel.

Functions

- def print_verbose (s, verbose=False)

  Print statement if verbose is True.
- def handler (chan, host, port, verbose=False)

  Handler is responsible for sending and receiving data through ssh tunnel.
- def reverse_forward_tunnel (server_port, remote_host, remote_port, transport, check=30, verbose=False)

  Creates a reverse ssh tunnel.

5.71.1 Function Documentation

5.71.1.1 handler()

```
def skdiscovery.utilities.ssh_reverse.handler (
              chan,
              host,
              port,
              verbose = False )
```

Handler is responsible for sending and receiving data through ssh tunnel.

Parameters

chan	SSH Channel for transferring data
host	Address of remote host
port	Port to forward
verbose	Print status information

5.71.1.2 print_verbose()

```
def skdiscovery.utilities.ssh_reverse.print_verbose (
              s,
              verbose = False )
```

Print statement if verbose is True.

Parameters

s	Statement to print
verbose	Print only if verbose is True

5.71.1.3 reverse_forward_tunnel()

```
def skdiscovery.utilities.ssh_reverse.reverse_forward_tunnel (
              server_port,
              remote_host,
              remote_port,
              transport,
              check = 30,
              verbose = False )
```

Creates a reverse ssh tunnel.

Parameters

server_port	Port on local host
remote_host	Address of remote host
remote_port	Port of remote host
transport	SSH Transport
check	Amount of time to wait in seconds when opening up a channel
verbose	Print status information

Returns

Thread running reverse ssh tunnel, event used to close ssh tunnel, list of child threads started by main thread

5.72 skdiscovery.utilities.trendTools Namespace Reference

Functions

- def getTrend (xdata)

  The getTrend function applies the signal.detrend function.
- def sinuFits (xdata, fitN=2, rmve=1)

  The sinuFits function fits annual and semi-annual sinusoid trends.
- def interpNaN (data)

  A simple wrapper for the linear interpolation function from Numpy to fill in NaN's.
- def medianFilter (data, window, interpolate=True)

  A median filter.

5.72.1 Function Documentation

5.72.1.1 getTrend()

```
def skdiscovery.utilities.trendTools.getTrend (
              xdata )
```

The getTrend function applies the signal.detrend function.

Returns the trend, given a time index input.

Parameters

xdata	1D time-series data in a pandas series format

Returns

the detrended data in pandas series format
the linear trend assuming a 1 day per sample time fit
the parameters for the linear trend

5.72.1.2 interpNaN()

```
def skdiscovery.utilities.trendTools.interpNaN (
              data )
```

A simple wrapper for the linear interpolation function from Numpy to fill in NaN's.

A copy of the data is made in place with endpoint NaN's extrapolated from closest non-NaN value Modified slightly from sample code at ref: http://stackoverflow.com/questions/6518811/interpolate-nan-values-in-a-numpy-a

Parameters

data	1d numpy or pandas array with possible NaN's

Returns

data after interpolation

5.72.1.3 medianFilter()

```
def skdiscovery.utilities.trendTools.medianFilter (
              data,
```

```
                window,
                interpolate = True )
```

A median filter.

If interpolate is True, data will be interpolated before smoothering. Otherwise, all available data within the window will be used

Parameters

data	Input data
window	Size of filter window
interpolate	Interpolate data before smoothing

Returns

Smoothed data

5.72.1.4 sinuFits()

```
def skdiscovery.utilities.trendTools.sinuFits (
                xdata,
                fitN = 2,
                rmve = 1 )
```

The sinuFits function fits annual and semi-annual sinusoid trends.

Other options allow for a monthly and seasonal sinusoid fit. The data is expected to be in pandas format

Parameters

xdata	1D time-series data in a pandas series format
fitN	the number of sinusoids to fit. 1-annual, 2-semi-annual, 3-seasonal, 4-monthly
rmve	a flag to return sinusoid removed data, or the sinusoids

Returns

retrDat: the returned data, either sinusoid removed or the sum of the sinusoids

5.73 skdiscovery.visualization Namespace Reference

Namespaces

- multi_ca_plot
- multi_dist

5.74 skdiscovery.visualization.multi_ca_plot Namespace Reference

Functions

- def multiCaPlot (pipeline, mogiFlag=False, offset=.15, direction='H', pca_comp=0, scaleFactor=2.5, map_res='i')

  The multiCaPlot function generates a geographic eigenvector plot of several pipeline runs.

5.74.1 Function Documentation

5.74.1.1 multiCaPlot()

```
def skdiscovery.visualization.multiCaPlot (
              pipeline,
              mogiFlag = False,
              offset = .15,
              direction = 'H',
              pca_comp = 0,
              scaleFactor = 2.5,
              map_res = 'i' )
```

The multiCaPlot function generates a geographic eigenvector plot of several pipeline runs.

This function plots multiple pipeline runs over perturbed pipeline parameters. The various perturbations are plotted more transparently (alpha=.5), while the median eigen_vector and Mogi inversion are plotted in solid blue and red

Parameters

pipeline	The pipeline object with multiple runs
mogiFlag	Flag to indicate plotting the Mogi source as well as the PCA
offset	Offset for padding the corners of the generated map
direction	Indicates the eigenvectors to plot. Only Horizontal component is currently supported ('H')
pca_comp	Choose the PCA component to use (integer)
scaleFactor	Size of the arrow scaling factor : Map data resolution for Basemap ('c', 'i', 'h', 'f', or None)

5.75 skdiscovery.visualization.multi_dist Namespace Reference

Functions

- def calc_distance_map (pipeline, ap_name, ca_name, ca_type, plotFlag=True, histIdx=False, fontsize=10)

  Calculates distances/similarities between pipeline runs.

Variables

- font

5.75.1 Function Documentation

5.75.1.1 calc_distance_map()

```
def skdiscovery.visualization.calc_distance_map (
              pipeline,
              ap_name,
              ca_name,
              ca_type,
              plotFlag = True,
              histIdx = False,
              fontsize = 10 )
```

Calculates distances/similarities between pipeline runs.

Optionally visualizes the result as a seaborn clustermap for PBO pipelines (requires multiple stations)

Calculates the square root of the summed squared differences between eigenvectors. Only works, because of internal assumptions, on pipelines with multiple stations Returns the distances as a pandas dataframe

Parameters

pipeline	Pipeline to analyze.
ap_name	Name of the pipeline item that is being perturbed
ca_name	Name of the pipeline item used as the comparison metric for calculating the distance
ca_type	Type of comparison metric [PCA for PCA, MogiSource of Mogi Source, MogiVector for Mogi vectors]
plotFlag	Boolean flag for plotting the clustermap of distances
histIdx	Flag for returning the perturbed pipeline item parameters
fontsize	Fontsize adjustments

Returns

cg: The generated clustermap of the calculated distances/similarities
dist_mat: A matrix of the calculated distances/similarities
history: The record of the perturbed pipeline item parameters

5.75.2 Variable Documentation

5.75.2.1 font

skdiscovery.visualization.font

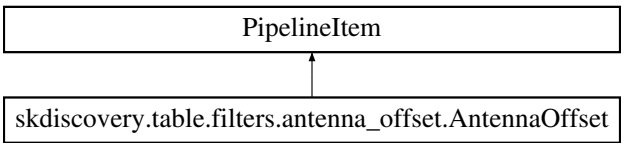
Chapter 6

Class Documentation

6.1 skdiscovery.table.filters.AntennaOffset Class Reference

Applies corrections to fix offsets in PBO GPS data induced by antenna changes.

Inheritance diagram for skdiscovery.table.filters.AntennaOffset:



Public Member Functions

- def __init__ (self, str_description, antenna_data, min_diff=0.0, column_list=None)

  Initialize AntennaOffset function.
- def process (self, obj_data)

  Applies the function to the data, updating in place.

Public Attributes

- antenna_data
- column_list
- min_diff

6.1.1 Detailed Description

Applies corrections to fix offsets in PBO GPS data induced by antenna changes.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 __init__()

```
def skdiscovery.table.filters.AntennaOffset.__init__ (
              self,
              str_description,
              antenna_data,
              min_diff = 0.0,
              column_list = None )
```

Initialize AntennaOffset function.

Parameters

str_description	String describing the filter
antenna_data	Data containing the log of antenna changes
min_diff	Difference in position needed to be considered an offset
column_list	Names of the columns to apply the function to

6.1.3 Member Function Documentation

6.1.3.1 process()

```
def skdiscovery.table.filters.AntennaOffset.process (
              self,
              obj_data )
```

Applies the function to the data, updating in place.

Parameters

obj_data	Table data wrapper

6.1.4 Member Data Documentation

6.1.4.1 antenna_data

skdiscovery.table.filters.AntennaOffset.antenna_data

6.1.4.2 column_list

skdiscovery.table.filters.AntennaOffset.column_list

6.1.4.3 min_diff

skdiscovery.table.filters.AntennaOffset.min_diff

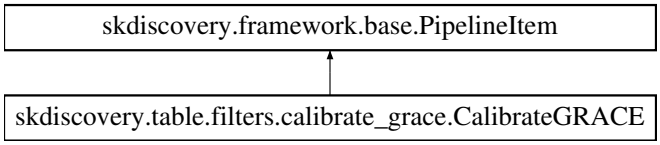
The documentation for this class was generated from the following file:

- table/filters/antenna_offset.py

6.2 skdiscovery.table.filters.CalibrateGRACE Class Reference

Calibrate Grace Data.

Inheritance diagram for skdiscovery.table.filters.CalibrateGRACE:



Public Member Functions

- def __init__ (self, str_description, ewd_column_name='EWD', round_dates=True)
  Initialize GRACE calibration filter.
- def process (self, obj_data)
  Calibrates GRACE, updating in place.
- def perturbParams (self)
  choose other random value for all parameters
- def resetParams (self)
  set all parameters to initial value
- def __str__ (self)
  String represntation of object.
- def getMetadata (self)
  Retrieve metadata about filter.

Public Attributes

- ewd_column_name
- round_dates
- str_description
- ap_paramList
- ap_paramNames

6.2.1 Detailed Description

Calibrate Grace Data.

Averages the three solutions and applies a scale factor

6.2.2 Constructor & Destructor Documentation

6.2.2.1 __init__()

```
def skdiscovery.table.filters.CalibrateGRACE.__init__ (
              self,
              str_description,
              ewd_column_name = 'EWD',
              round_dates = True )
```

Initialize GRACE calibration filter.

Parameters

str_description	String describing filter
ewd_column_name	Name of new column for the calibrated GRACE data
round_dates	Option for rounding to dates to the nearest day

6.2.3 Member Function Documentation

6.2.3.1 __str__()

```
def skdiscovery.framework.PipelineItem.__str__ (
              self )  [inherited]
```

String represntation of object.

Returns

String listing all currenter parameters

6.2.3.2 getMetadata()

```
def skdiscovery.framework.PipelineItem.getMetadata (
              self )  [inherited]
```

Retrieve metadata about filter.

Returns

String containing the item description and current parameters for filter.

6.2.3.3 perturbParams()

```
def skdiscovery.framework.PipelineItem.perturbParams (
              self )  [inherited]
```

choose other random value for all parameters

6.2.3.4 process()

```
def skdiscovery.table.filters.CalibrateGRACE.process (
              self,
              obj_data )
```

Calibrates GRACE, updating in place.

Parameters

obj_data	Table data wrapper

6.2.3.5 resetParams()

```
def skdiscovery.framework.PipelineItem.resetParams (
              self )  [inherited]
```

set all parameters to initial value

6.2.4 Member Data Documentation

6.2.4.1 ap_paramList

```
skdiscovery.framework.PipelineItem.ap_paramList  [inherited]
```

6.2.4.2 ap_paramNames

```
skdiscovery.framework.PipelineItem.ap_paramNames  [inherited]
```

6.2.4.3 ewd_column_name

```
skdiscovery.table.filters.CalibrateGRACE.ewd_column_name
```

6.2.4.4 round_dates

```
skdiscovery.table.filters.CalibrateGRACE.round_dates
```

6.2.4.5 str_description

```
skdiscovery.framework.PipelineItem.str_description  [inherited]
```

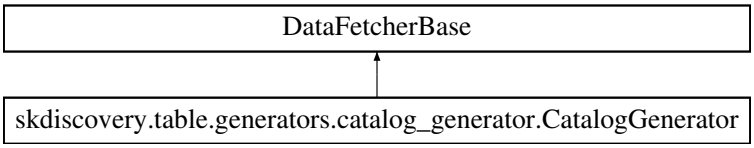
The documentation for this class was generated from the following file:

- table/filters/calibrate_py

6.3 skdiscovery.table.generators.CatalogGenerator Class Reference

In Generates galaxy catalogs for use in DiscoveryPipeline.

Inheritance diagram for skdiscovery.table.generators.CatalogGenerator:



Public Member Functions

- def __init__ (self, ap_paramList, ra1, dec1, ra2, dec2, background_density, z)
- def output (self)

  Generates galaxy catalog.
- def nfw_cumulative (self, R)

  Cumulative radial NFW distribution.
- def inverse_nfw_cumulative (self, p)

  inverse of radial nfw cumulative distribution

Public Attributes

- ra1
- dec1
- ra2
- dec2
- background_density
- z

6.3.1 Detailed Description

In Generates galaxy catalogs for use in DiscoveryPipeline.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 __init__()

```
def skdiscovery.table.generators.CatalogGenerator.__init__ (
              self,
              ap_paramList,
              ra1,
              dec1,
              ra2,
              dec2,
              background_density,
              z )
```

Parameters

ap_paramList[seed]	Seed for random number generator
ra1	Left right ascension
dec1	Bottom declination
ra2	Right right ascension
dec2	Top declination
background_density	galaxy background density in galaxies/square degree
z	Redshift of galaxy cluster

6.3.3 Member Function Documentation

6.3.3.1 inverse_nfw_cumulative()

```
def skdiscovery.table.generators.CatalogGenerator.inverse_nfw_cumulative (
              self,
              p )
```

inverse of radial nfw cumulative distribution

Parameters

p	Probability

Returns

float: Radius corresponding to probability p

6.3.3.2 nfw_cumulative()

```
def skdiscovery.table.generators.CatalogGenerator.nfw_cumulative (
              self,
              R )
```

Cumulative radial NFW distribution.

Parameters

R	Radius

Returns

float: Probability of being within R

6.3.3.3 output()

```
def skdiscovery.table.generators.CatalogGenerator.output (
              self )
```

Generates galaxy catalog.

Returns

DataWrapper: Table data wrapper of galaxy catalog

6.3.4 Member Data Documentation

6.3.4.1 background_density

```
skdiscovery.table.generators.CatalogGenerator.background_density
```

6.3.4.2 dec1

```
skdiscovery.table.generators.CatalogGenerator.dec1
```

6.3.4.3 dec2

```
skdiscovery.table.generators.CatalogGenerator.dec2
```

6.3.4.4 ra1

```
skdiscovery.table.generators.CatalogGenerator.ra1
```

6.3.4.5 ra2

skdiscovery.table.generators.CatalogGenerator.ra2

6.3.4.6 z

skdiscovery.table.generators.CatalogGenerator.z

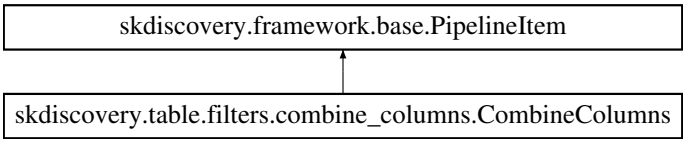
The documentation for this class was generated from the following file:

- table/generators/catalog_generator.py

6.4 skdiscovery.table.filters.CombineColumns Class Reference

Create a new column by selecting data from a column.

Inheritance diagram for skdiscovery.table.filters.CombineColumns:



Public Member Functions

- def __init__ (self, str_description, column_1, column_2, new_column_name)

  Initialize a CombineColumns object.
- def process (self, obj_data)

  Apply combine column filter to data set, operating on the data_obj.
- def perturbParams (self)

  choose other random value for all parameters
- def resetParams (self)

  set all parameters to initial value
- def __str__ (self)

  String represntation of object.
- def getMetadata (self)

  Retrieve metadata about filter.

Public Attributes

- column_1
- column_2
- new_column_name
- str_description
- ap_paramList
- ap_paramNames

6.4.1 Detailed Description

Create a new column by selecting data from a column.

Fills in any missing values using a second column

6.4.2 Constructor & Destructor Documentation

6.4.2.1 __init__()

```
def skdiscovery.table.filters.CombineColumns.__init__ (
              self,
              str_description,
              column_1,
              column_2,
              new_column_name )
```

Initialize a CombineColumns object.

Parameters

str_description	String describing filter
column_1	Name of primary column
column_2	Name of secondary column to be used when data from the primary column is not avaiable
new_column_name	Name of resulting column

6.4.3 Member Function Documentation

6.4.3.1 __str__()

```
def skdiscovery.framework.PipelineItem.__str__ (
              self )  [inherited]
```

String represntation of object.

Returns

String listing all currenter parameters

6.4.3.2 getMetadata()

```
def skdiscovery.framework.PipelineItem.getMetadata (
              self )  [inherited]
```

Retrieve metadata about filter.

Returns

String containing the item description and current parameters for filter.

6.4.3.3 perturbParams()

```
def skdiscovery.framework.PipelineItem.perturbParams (
              self )  [inherited]
```

choose other random value for all parameters

6.4.3.4 process()

```
def skdiscovery.table.filters.CombineColumns.process (
              self,
              obj_data )
```

Apply combine column filter to data set, operating on the data_obj.

Parameters

obj_data	Table data wrapper.

6.4.3.5 resetParams()

```
def skdiscovery.framework.PipelineItem.resetParams (
              self )  [inherited]
```

set all parameters to initial value

6.4.4 Member Data Documentation

6.4.4.1 ap_paramList

```
skdiscovery.framework.PipelineItem.ap_paramList  [inherited]
```

6.4.4.2 ap_paramNames

```
skdiscovery.framework.PipelineItem.ap_paramNames  [inherited]
```

6.4.4.3 column_1

```
skdiscovery.table.filters.CombineColumns.column_1
```

6.4.4.4 column_2

```
skdiscovery.table.filters.CombineColumns.column_2
```

6.4.4.5 new_column_name

```
skdiscovery.table.filters.CombineColumns.new_column_name
```

6.4.4.6 str_description

```
skdiscovery.framework.PipelineItem.str_description  [inherited]
```

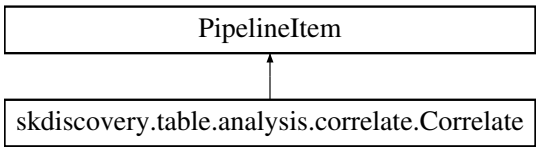
The documentation for this class was generated from the following file:

- table/filters/combine_columns.py

6.5 skdiscovery.table.analysis.Correlate Class Reference

Computes the correlation for table data and stores the result as a matrix.

Inheritance diagram for skdiscovery.table.analysis.Correlate:



Public Member Functions

- def __init__ (self, str_description, column_names=None, local_match=False, correlation_type='pearson')
  Initialize Correlate analysis item for use on tables.
- def process (self, obj_data)
  *Computes the correlation between columns and stores the results in obj_*

Public Attributes

- column_names
- local_match
- corr_type

6.5.1 Detailed Description

Computes the correlation for table data and stores the result as a matrix.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 __init__()

```
def skdiscovery.table.analysis.Correlate.__init__ (
              self,
              str_description,
              column_names = None,
              local_match = False,
              correlation_type = 'pearson' )
```

Initialize Correlate analysis item for use on tables.

Parameters

str_description	String describing analysis item
column_names	List of column names to correlate
local_match	Only correlate data on the same frames
correlation_type	Type of correlation to be passed to pandas ('pearson', 'kendall', 'spearman')

6.5.3 Member Function Documentation

6.5.3.1 process()

```
def skdiscovery.table.analysis.Correlate.process (
              self,
              obj_data )
```

Computes the correlation between columns and stores the results in obj_

Parameters

obj_data	Data wrapper

6.5.4 Member Data Documentation

6.5.4.1 column_names

```
skdiscovery.table.analysis.Correlate.column_names
```

6.5.4.2 corr_type

```
skdiscovery.table.analysis.Correlate.corr_type
```

6.5.4.3 local_match

`skdiscovery.table.analysis.Correlate.local_match`

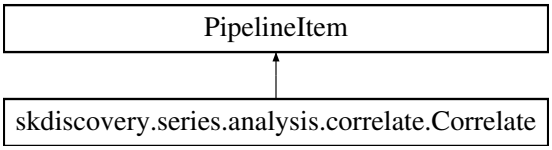
The documentation for this class was generated from the following file:

- table/analysis/correlate.py

6.6 skdiscovery.series.analysis.Correlate Class Reference

Computes the correlation for series data.

Inheritance diagram for skdiscovery.series.analysis.Correlate:



Public Member Functions

- def __init__ (self, str_description, labels=None, column_names=None)

  Initialize Correlate analysis item.
- def process (self, obj_data)

  Computes the correlation between all the time series.

Public Attributes

- labels
- column_names

6.6.1 Detailed Description

Computes the correlation for series data.

Stores the result as a matrix

6.6.2 Constructor & Destructor Documentation

6.6.2.1 __init__()

```
def skdiscovery.series.analysis.Correlate.__init__ (
              self,
              str_description,
              labels = None,
              column_names = None )
```

Initialize Correlate analysis item.

Parameters

str_description	String describing analysis item
labels	List of labels used to select data
column_names	List of column names used to select data

6.6.3 Member Function Documentation

6.6.3.1 process()

```
def skdiscovery.series.analysis.Correlate.process (
              self,
              obj_data )
```

Computes the correlation between all the time series.

The results are stored in obj_data

Parameters

obj_data	Data wrapper for correlating

6.6.4 Member Data Documentation

6.6.4.1 column_names

```
skdiscovery.series.analysis.Correlate.column_names
```

6.6.4.2 labels

```
skdiscovery.series.analysis.Correlate.labels
```

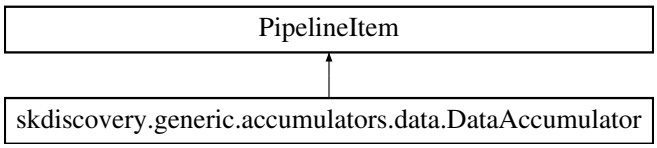
The documentation for this class was generated from the following file:

- series/analysis/correlate.py

6.7 skdiscovery.generic.accumulators.DataAccumulator Class Reference

Stores a copy of the data in its current state in the pipeline.

Inheritance diagram for skdiscovery.generic.accumulators.DataAccumulator:



Public Member Functions

- def process (self, obj_data)

  Store a copy of the data in the object wrapper results.

6.7.1 Detailed Description

Stores a copy of the data in its current state in the pipeline.

6.7.2 Member Function Documentation

6.7.2.1 process()

```
def skdiscovery.generic.accumulators.DataAccumulator.process (
              self,
              obj_data )
```

Store a copy of the data in the object wrapper results.

Parameters

obj_data	Data Wrapper to be copied

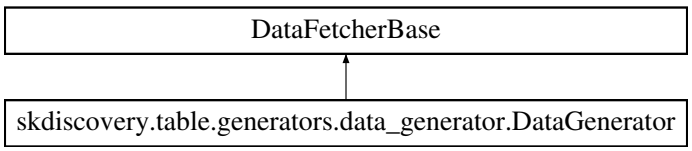
The documentation for this class was generated from the following file:

- generic/accumulators/data.py

6.8 skdiscovery.table.generators.DataGenerator Class Reference

In Class for generating random data.

Inheritance diagram for skdiscovery.table.generators.DataGenerator:



Public Member Functions

- def __init__ (self, length, args, seed=None, final_function=None)
  Initialize Random data generator.
- def output (self)

Public Attributes

- length
- seed
- args
- final_function

6.8.1 Detailed Description

In Class for generating random data.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 __init__()

```
def skdiscovery.table.generators.DataGenerator.__init__ (
              self,
              length,
              args,
              seed = None,
              final_function = None )
```

Initialize Random data generator.

Parameters

length	Number of rows to generate
∗args	Dictionaries containing entries: 'name','start', 'end', and optionally 'func'
seed	Seed to use when generating random data
final_function	Final function to call on random data

6.8.3 Member Function Documentation

6.8.3.1 output()

```
def skdiscovery.table.generators.DataGenerator.output (
              self )
```

6.8.4 Member Data Documentation

6.8.4.1 args

```
skdiscovery.table.generators.DataGenerator.args
```

6.8.4.2 final_function

```
skdiscovery.table.generators.DataGenerator.final_function
```

6.8.4.3 length

```
skdiscovery.table.generators.DataGenerator.length
```

6.8.4.4 seed

`skdiscovery.table.generators.DataGenerator.seed`

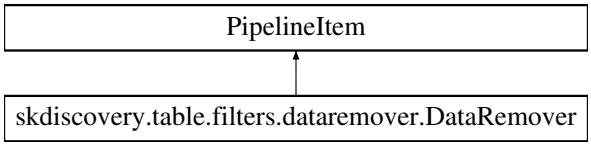
The documentation for this class was generated from the following file:

- table/generators/data_generator.py

6.9 skdiscovery.table.filters.DataRemover Class Reference

Sets specified table data to NaN.

Inheritance diagram for skdiscovery.table.filters.DataRemover:



Public Member Functions

- def __init__ (self, str_description, column_names, start=None, end=None, labels=None)
  Initialize DataRemover.
- def process (self, obj_data)
  NaN's data from DataWrapper.

Public Attributes

- labels
- column_names
- start
- end

6.9.1 Detailed Description

Sets specified table data to NaN.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 __init__()

```
def skdiscovery.table.filters.DataRemover.__init__ (
              self,
              str_description,
              column_names,
              start = None,
              end = None,
              labels = None )
```

Initialize DataRemover.

Parameters

str_description	String describing filter
column_names	List of column names to select data to be removed (using None will apply to all columns)
start	Starting index value
end	Ending index value (inclusive)
labels	List of labels used to select data to be removed (using None will apply to all labels)

6.9.3 Member Function Documentation

6.9.3.1 process()

```
def skdiscovery.table.filters.DataRemover.process (
              self,
              obj_data )
```

NaN's data from DataWrapper.

Parameters

obj_data	Input DataWrapper, will be modified in place

6.9.4 Member Data Documentation

6.9.4.1 column_names

```
skdiscovery.table.filters.DataRemover.column_names
```

6.9.4.2 end

`skdiscovery.table.filters.DataRemover.end`

6.9.4.3 labels

`skdiscovery.table.filters.DataRemover.labels`

6.9.4.4 start

`skdiscovery.table.filters.DataRemover.start`

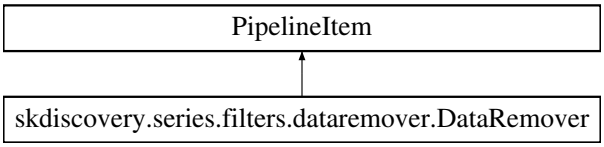
The documentation for this class was generated from the following file:

- table/filters/dataremover.py

6.10 skdiscovery.series.filters.DataRemover Class Reference

Sets specified series data to NaN.

Inheritance diagram for skdiscovery.series.filters.DataRemover:



Public Member Functions

- def __init__ (self, str_description, start=None, end=None, labels=None, column_names=None)
  Initialize DataRemover.
- def process (self, obj_data)
  NaN's data from DataWrapper.

Public Attributes

- labels
- column_names
- start
- end

6.10.1 Detailed Description

Sets specified series data to NaN.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 __init__()

```
def skdiscovery.series.filters.DataRemover.__init__ (
              self,
              str_description,
              start = None,
              end = None,
              labels = None,
              column_names = None )
```

Initialize DataRemover.

Parameters

str_description	String describing filter
start	Starting index value
end	Ending index value (inclusive)
labels	List of labels used to select data to be removed (None will operate on all labels)
column_names	List of column names to select data to be removed (None will operate on all columns)

6.10.3 Member Function Documentation

6.10.3.1 process()

```
def skdiscovery.series.filters.DataRemover.process (
              self,
              obj_data )
```

NaN's data from DataWrapper.

Parameters

obj_data	Input DataWrapper, which will be modified in place

6.10.4 Member Data Documentation

6.10.4.1 column_names

`skdiscovery.series.filters.DataRemover.column_names`

6.10.4.2 end

`skdiscovery.series.filters.DataRemover.end`

6.10.4.3 labels

`skdiscovery.series.filters.DataRemover.labels`

6.10.4.4 start

`skdiscovery.series.filters.DataRemover.start`

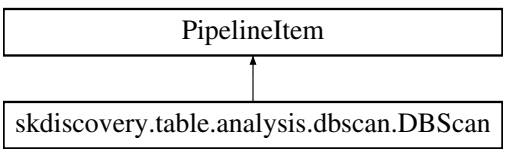
The documentation for this class was generated from the following file:

- series/filters/dataremover.py

6.11 skdiscovery.table.analysis.DBScan Class Reference

Runs DBScan on table data.

Inheritance diagram for skdiscovery.table.analysis.DBScan:



Public Member Functions

- def __init__ (self, str_description, ap_paramList, column_names)

  Initialize DBScan pipelne item.
- def process (self, obj_data)

  Run DBScan on data.

Public Attributes

- column_names

6.11.1 Detailed Description

Runs DBScan on table data.

Adds cluster information column to data

6.11.2 Constructor & Destructor Documentation

6.11.2.1 __init__()

```
def skdiscovery.table.analysis.DBScan.__init__ (
              self,
              str_description,
              ap_paramList,
              column_names )
```

Initialize DBScan pipelne item.

Parameters

str_description	Description of item
ap_paramList[epsilon]	Distance between two nodes for them to be considered connected
ap_paramList[min_points]	Minimum number of points for a cluster
column_names	List of column names to use

6.11.3 Member Function Documentation

6.11.3.1 process()

```
def skdiscovery.table.analysis.DBScan.process (
              self,
              obj_data )
```

Run DBScan on data.

Stores result in data wrapper

```
   @param obj_data: Data wrapper to be processed
```

6.11.4 Member Data Documentation

6.11.4.1 column_names

```
skdiscovery.table.analysis.DBScan.column_names
```

The documentation for this class was generated from the following file:

- table/analysis/dbscan.py

6.12 skdiscovery.DiscoveryPipeline Class Reference

Pipeline for running the analysis.

Public Member Functions

- def __init__ (self, data_fetcher, list_StageContainers)
  Initialize a new pipeline.
- def run (self, num_runs=1, perturb_data=False, num_cores=1, amazon=False, verbose=False)
  Run the pipeline.
- def perturb (self)
  Perturb the paramters in the stage containers.
- def reset (self)
  Reset the stage containers to their default values and clear previous runs.
- def getMetadata (self)
  Retrieve Metadata from stage containers.
- def getMetadataHistory (self)
  Get the metadata for each run in the pipeline.
- def perturbData (self)

Perturb the input data.

- def getResults (self, index=None)

  Return results from previous runs.

- def resultIter (self)

  Retrieves and iterator to the results and history of the pipeline.

- def plotPipelineInstance (self)

  Plot current instance of pipeline stages with metadata.

- def plotPipelineStructure (self)

  Plot pipeline structure.

- def getMetadataNestedTypes (self)

  Get the Metadata Nested Types.

- def getMetadataNestedGraph (self)

  Retrieve the metadata nested graph.

- def __str__ (self)

  String representation of the pipeline.

Public Attributes

- stage_containers
- data_fetcher
- stageConfigurationHistory
- RA_results

6.12.1 Detailed Description

Pipeline for running the analysis.

6.12.2 Constructor & Destructor Documentation

6.12.2.1 __init__()

```
def skdiscovery.DiscoveryPipeline.__init__ (
              self,
              data_fetcher,
              list_StageContainers )
```

Initialize a new pipeline.

Parameters

data_fetcher	Data fetcher to use as a data source (from skdaccess)
list_StageContainers	List of stage containers

6.12.3 Member Function Documentation

6.12.3.1 __str__()

```
def skdiscovery.DiscoveryPipeline.__str__ (
              self )
```

String representation of the pipeline.

Returns

String of current metadata of pipeline containers.

6.12.3.2 getMetadata()

```
def skdiscovery.DiscoveryPipeline.getMetadata (
              self )
```

Retrieve Metadata from stage containers.

Returns

list of metadata for the current run

6.12.3.3 getMetadataHistory()

```
def skdiscovery.DiscoveryPipeline.getMetadataHistory (
              self )
```

Get the metadata for each run in the pipeline.

Returns

list of metadata configurations for all runs

6.12.3.4 getMetadataNestedGraph()

```
def skdiscovery.DiscoveryPipeline.getMetadataNestedGraph (
              self )
```

Retrieve the metadata nested graph.

Returns

String: Metadata nested graph

6.12.3.5 getMetadataNestedTypes()

```
def skdiscovery.DiscoveryPipeline.getMetadataNestedTypes (
              self )
```

Get the Metadata Nested Types.

Returns

String: Metadata Nested types

6.12.3.6 getResults()

```
def skdiscovery.DiscoveryPipeline.getResults (
              self,
              index = None )
```

Return results from previous runs.

Parameters

index	Index of run. If None, return all previous results

Returns

results from a run at index. If index=None, returns list of all results

6.12.3.7 perturb()

```
def skdiscovery.DiscoveryPipeline.perturb (
              self )
```

Perturb the paramters in the stage containers.

6.12.3.8 perturbData()

```
def skdiscovery.DiscoveryPipeline.perturbData (
              self )
```

Perturb the input data.

6.12.3.9 plotPipelineInstance()

```
def skdiscovery.DiscoveryPipeline.plotPipelineInstance (
              self )
```

Plot current instance of pipeline stages with metadata.

Returns

iPython display object

6.12.3.10 plotPipelineStructure()

```
def skdiscovery.DiscoveryPipeline.plotPipelineStructure (
              self )
```

Plot pipeline structure.

Returns

iPython display object

6.12.3.11 reset()

```
def skdiscovery.DiscoveryPipeline.reset (
              self )
```

Reset the stage containers to their default values and clear previous runs.

6.12.3.12 resultIter()

```
def skdiscovery.DiscoveryPipeline.resultIter (
              self )
```

Retrieves and iterator to the results and history of the pipeline.

Returns

A 2 component iterator to the results and history of previous runs

6.12.3.13 run()

```
def skdiscovery.DiscoveryPipeline.run (
              self,
              num_runs = 1,
              perturb_data = False,
              num_cores = 1,
              amazon = False,
              verbose = False )
```

Run the pipeline.

Parameters

num_runs	Number of times to run the pipeline
perturb_data	Boolean flag. If running the pipeline multiple times then perturb the data instead of the pipeline
num_cores	Number of cores on the local machine to use. Defaults to 1 core. Use 0 to select the minimum between the number of runs and cpu cores.
amazon	Offload the pipeline on amazon
verbose	Display the pipeline for each run

6.12.4 Member Data Documentation

6.12.4.1 data_fetcher

`skdiscovery.DiscoveryPipeline.data_fetcher`

6.12.4.2 RA_results

`skdiscovery.DiscoveryPipeline.RA_results`

6.12.4.3 stage_containers

`skdiscovery.DiscoveryPipeline.stage_containers`

6.12.4.4 stageConfigurationHistory

`skdiscovery.DiscoveryPipeline.stageConfigurationHistory`

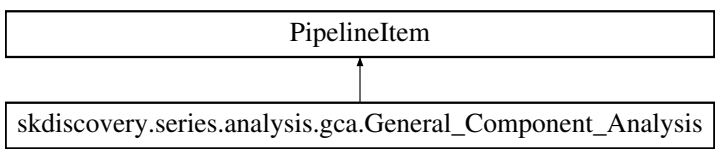
The documentation for this class was generated from the following file:

- framework/discoverypipeline.py

6.13 skdiscovery.series.analysis.General_Component_Analysis Class Reference

Performs either ICA or PCA analysis on series data.

Inheritance diagram for skdiscovery.series.analysis.General_Component_Analysis:



Public Member Functions

- def __init__ (self, str_description, ap_paramList)

  Initialize Analysis object.
- def process (self, obj_data)

  Perform component analysis on data:

Public Attributes

- str_description
- ap_paramList
- ap_paramNames
- results

6.13.1 Detailed Description

Performs either ICA or PCA analysis on series data.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 __init__()

```
def skdiscovery.series.analysis.General_Component_Analysis.__init__ (
              self,
              str_description,
              ap_paramList )
```

Initialize Analysis object.

Parameters

str_description	String description of analysis
ap_paramList[num_components]	Number of components
ap_paramList[component_type]	Type of component analysis (CA); either PCA or ICA
ap_paramList[start_time]	Starting time for CA
ap_paramList[end_time]	ending time for CA
ap_paramList[labels]	Optional list of label names

6.13.3 Member Function Documentation

6.13.3.1 process()

```
def skdiscovery.series.analysis.General_Component_Analysis.process (
              self,
              obj_data )
```

Perform component analysis on data:

Results are added to the data wrapper as a dictionary with results['CA'] = Eigenvenctors results['Projection'] = Projection on to the eigenvectors

Parameters

obj_data	Data wrapper containing the data

6.13.4 Member Data Documentation

6.13.4.1 ap_paramList

```
skdiscovery.series.analysis.General_Component_Analysis.ap_paramList
```

6.13.4.2 ap_paramNames

```
skdiscovery.series.analysis.General_Component_Analysis.ap_paramNames
```

6.13.4.3 results

```
skdiscovery.series.analysis.General_Component_Analysis.results
```

6.13.4.4 str_description

`skdiscovery.series.analysis.General_Component_Analysis.str_description`

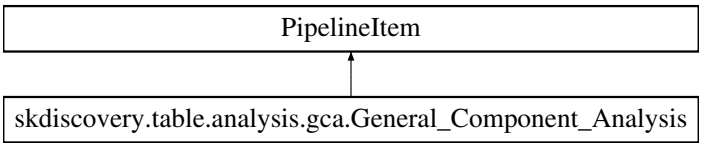
The documentation for this class was generated from the following file:

- series/analysis/gca.py

6.14 skdiscovery.table.analysis.General_Component_Analysis Class Reference

Performs a general component analysis on table data.

Inheritance diagram for skdiscovery.table.analysis.General_Component_Analysis:



Public Member Functions

- def __init__ (self, str_description, ap_paramList, n_components, column_names)

  Initialize Analysis object.
- def process (self, obj_data)

  Perform component analysis on data.

Public Attributes

- str_description
- ap_paramList
- ap_paramNames
- n_components
- column_names
- results

6.14.1 Detailed Description

Performs a general component analysis on table data.

Currently, the two built-in types of analysis are either ICA or PCA.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 __init__()

```
def skdiscovery.table.analysis.General_Component_Analysis.__init__ (
              self,
              str_description,
              ap_paramList,
              n_components,
              column_names )
```

Initialize Analysis object.

Parameters

str_description	String description of analysis
ap_paramList[component_type]	Type of CA; either PCA or ICA
ap_paramList[start_time]	Starting time for CA
ap_paramList[end_time]	ending time for CA
n_components	Number of components to compute
column_names	Columns names to use

6.14.3 Member Function Documentation

6.14.3.1 process()

```
def skdiscovery.table.analysis.General_Component_Analysis.process (
              self,
              obj_data )
```

Perform component analysis on data.

Results are added to the data wrapper as a dictionary with results['CA'] = Eigenvenctors results['Projection'] = Projection on to the eigenvectors

Parameters

obj_data	Data wrapper

6.14.4 Member Data Documentation

6.14.4.1 ap_paramList

skdiscovery.table.analysis.General_Component_Analysis.ap_paramList

6.14.4.2 ap_paramNames

skdiscovery.table.analysis.General_Component_Analysis.ap_paramNames

6.14.4.3 column_names

skdiscovery.table.analysis.General_Component_Analysis.column_names

6.14.4.4 n_components

skdiscovery.table.analysis.General_Component_Analysis.n_components

6.14.4.5 results

skdiscovery.table.analysis.General_Component_Analysis.results

6.14.4.6 str_description

skdiscovery.table.analysis.General_Component_Analysis.str_description

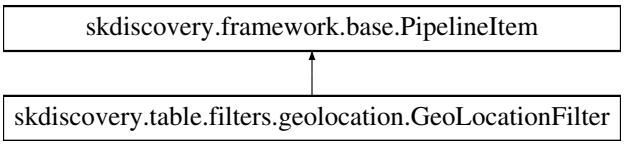
The documentation for this class was generated from the following file:

- table/analysis/gca.py

6.15 skdiscovery.table.filters.GeoLocationFilter Class Reference

Removes objects not located in a specified region.

Inheritance diagram for skdiscovery.table.filters.GeoLocationFilter:



Public Member Functions

- def __init__ (self, str_description, ap_paramList)
  Initialize GeolocationFilter.
- def process (self, obj_data)
  Apply geolocation filter to data set.
- def perturbParams (self)
  choose other random value for all parameters
- def resetParams (self)
  set all parameters to initial value
- def __str__ (self)
  String represntation of object.
- def getMetadata (self)
  Retrieve metadata about filter.

Public Attributes

- str_description
- ap_paramList
- ap_paramNames

6.15.1 Detailed Description

Removes objects not located in a specified region.

6.15.2 Constructor & Destructor Documentation

6.15.2.1 __init__()

```
def skdiscovery.table.filters.GeoLocationFilter.__init__ (
            self,
            str_description,
            ap_paramList )
```

Initialize GeolocationFilter.

Parameters

str_description	String describing filter
ap_paramList[ap_lat]	Latitude coordinate
ap_paramList[ap_lon]	Longitude coordinate
ap_paramList[ap_radius]	cut objects whose distance from lat/lon is greater than ap_radius

6.15.3 Member Function Documentation

6.15.3.1 __str__()

```
def skdiscovery.framework.PipelineItem.__str__ (
              self )  [inherited]
```

String represntation of object.

Returns

String listing all currenter parameters

6.15.3.2 getMetadata()

```
def skdiscovery.framework.PipelineItem.getMetadata (
              self )  [inherited]
```

Retrieve metadata about filter.

Returns

String containing the item description and current parameters for filter.

6.15.3.3 perturbParams()

```
def skdiscovery.framework.PipelineItem.perturbParams (
              self )  [inherited]
```

choose other random value for all parameters

6.15.3.4 process()

```
def skdiscovery.table.filters.GeoLocationFilter.process (
              self,
              obj_data )
```

Apply geolocation filter to data set.

Parameters

obj_data	Table data wrapper

6.15.3.5 resetParams()

```
def skdiscovery.framework.PipelineItem.resetParams (
              self )  [inherited]
```

set all parameters to initial value

6.15.4 Member Data Documentation

6.15.4.1 ap_paramList

```
skdiscovery.framework.PipelineItem.ap_paramList  [inherited]
```

6.15.4.2 ap_paramNames

```
skdiscovery.framework.PipelineItem.ap_paramNames  [inherited]
```

6.15.4.3 str_description

```
skdiscovery.framework.PipelineItem.str_description  [inherited]
```

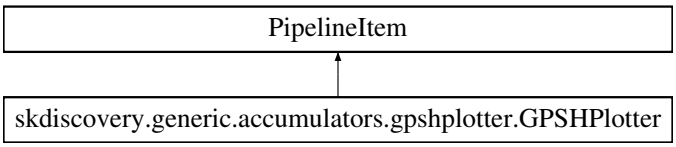
The documentation for this class was generated from the following file:

- table/filters/geolocation.py

6.16 skdiscovery.generic.accumulators.GPSHPlotter Class Reference

Plots results from General_Component_Analysis, for the GPS horizontal or vertical components.

Inheritance diagram for skdiscovery.generic.accumulators.GPSHPlotter:



Public Member Functions

- def __init__ (self, str_description, comp_name, mogi_name=None, pca_dir='H', pca_comp=0, scaleFactor=2.5, offset=.15, KF_tau=0, errorEllipses=False, map_resolution='i')

  Initialize GPHSHPlotter.
- def process (self, obj_data)

  *Plot the General Component Analysis results present stored in obj_*

Public Attributes

- dir_sign
- pca_dir
- pca_comp
- scaleFactor
- offset
- errorE
- KF_tau
- comp_name
- mogi_name

6.16.1 Detailed Description

Plots results from General_Component_Analysis, for the GPS horizontal or vertical components.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 __init__()

```
def skdiscovery.generic.accumulators.GPSHPlotter.__init__ (
              self,
              str_description,
              comp_name,
              mogi_name = None,
              pca_dir = 'H',
              pca_comp = 0,
              scaleFactor = 2.5,
              offset = .15,
              KF_tau = 0,
              errorEllipses = False,
              map_resolution = 'i' )
```

Initialize GPHSHPlotter.

Parameters

str_description	String describing accumulator
comp_name	Name of the GPCA results for accessing the GPCA output
mogi_name	Name of the Mogi results (optional)
pca_dir	PCA direction to plot, horizontal (H) or vertical (V)
pca_comp	The PCA component that will be plotted
scaleFacter	Scale factor for arrows
offset	Offset for plotting larger area on map
KF_tau	Tau used in kalman filter
errorEllpses	Boolean indicating whether or not to plot errorEllipses

6.16.3 Member Function Documentation

6.16.3.1 process()

```
def skdiscovery.generic.accumulators.GPSHPlotter.process (
              self,
              obj_data )
```

Plot the General Component Analysis results present stored in obj_

Saves the basemap in obj_data results.

Parameters

obj_data	Data Wrapper that holds component analysis HPCA

6.16.4 Member Data Documentation

6.16.4.1 comp_name

skdiscovery.generic.accumulators.GPSHPlotter.comp_name

6.16.4.2 dir_sign

skdiscovery.generic.accumulators.GPSHPlotter.dir_sign

6.16.4.3 errorE

skdiscovery.generic.accumulators.GPSHPlotter.errorE

6.16.4.4 KF_tau

skdiscovery.generic.accumulators.GPSHPlotter.KF_tau

6.16.4.5 mogi_name

skdiscovery.generic.accumulators.GPSHPlotter.mogi_name

6.16.4.6 offset

skdiscovery.generic.accumulators.GPSHPlotter.offset

6.16.4.7 pca_comp

skdiscovery.generic.accumulators.GPSHPlotter.pca_comp

6.16.4.8 pca_dir

skdiscovery.generic.accumulators.GPSHPlotter.pca_dir

6.16.4.9 scaleFactor

skdiscovery.generic.accumulators.GPSHPlotter.scaleFactor

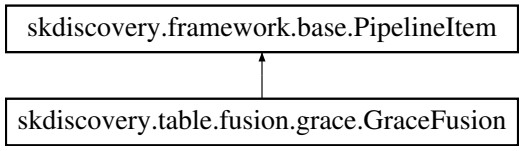
The documentation for this class was generated from the following file:

- generic/accumulators/gpshplotter.py

6.17 skdiscovery.table.fusion.GraceFusion Class Reference

Fuses GRACE equivelent water depth time series.

Inheritance diagram for skdiscovery.table.fusion.GraceFusion:



Public Member Functions

- def __init__ (self, str_description, metadata, column_data_name='Grace', column_error_name='Grace_↩ Uncertainty', gldas="Off")

  Initialize Grace Fusion item.
- def process (self, obj_data)

  Adds columns for GRACE data and uncertainties.
- def perturbParams (self)

  choose other random value for all parameters
- def resetParams (self)

  set all parameters to initial value
- def __str__ (self)

  String represntation of object.
- def getMetadata (self)

  Retrieve metadata about filter.

Public Attributes

- metadata
- column_data_name
- column_error_name
- gldas
- str_description
- ap_paramList
- ap_paramNames

6.17.1 Detailed Description

Fuses GRACE equivelent water depth time series.

Works on table data (original data from http://grace.jpl.nasa.gov/data/get-data/monthly-mass-grids-land/)

6.17.2 Constructor & Destructor Documentation

6.17.2.1 __init__()

```
def skdiscovery.table.fusion.GraceFusion.__init__ (
              self,
              str_description,
              metadata,
              column_data_name = 'Grace',
              column_error_name = 'Grace_Uncertainty',
              gldas = "Off" )
```

Initialize Grace Fusion item.

Parameters

str_description	String describing item
metadata	Metadata that contains lat,lon coordinates based on data labels
column_data_name	Name of column for GRACE data
column_error_name	Grace Uncertainty column name
gldas	Indicating use of the global land data assimilation water model

6.17.3 Member Function Documentation

6.17.3.1 __str__()

```
def skdiscovery.framework.PipelineItem.__str__ (
              self )  [inherited]
```

String represntation of object.

Returns

String listing all currenter parameters

6.17.3.2 getMetadata()

```
def skdiscovery.framework.PipelineItem.getMetadata (
              self )  [inherited]
```

Retrieve metadata about filter.

Returns

String containing the item description and current parameters for filter.

6.17.3.3 perturbParams()

```
def skdiscovery.framework.PipelineItem.perturbParams (
              self )  [inherited]
```

choose other random value for all parameters

6.17.3.4 process()

```
def skdiscovery.table.fusion.GraceFusion.process (
              self,
              obj_data )
```

Adds columns for GRACE data and uncertainties.

Parameters

obj_data	Input DataWrapper, will be modified in place

6.17.3.5 resetParams()

```
def skdiscovery.framework.PipelineItem.resetParams (
              self )  [inherited]
```

set all parameters to initial value

6.17.4 Member Data Documentation

6.17.4.1 ap_paramList

```
skdiscovery.framework.PipelineItem.ap_paramList  [inherited]
```

6.17.4.2 ap_paramNames

```
skdiscovery.framework.PipelineItem.ap_paramNames  [inherited]
```

6.17.4.3 column_data_name

```
skdiscovery.table.fusion.GraceFusion.column_data_name
```

6.17.4.4 column_error_name

```
skdiscovery.table.fusion.GraceFusion.column_error_name
```

6.17.4.5 gldas

```
skdiscovery.table.fusion.GraceFusion.gldas
```

6.17.4.6 metadata

```
skdiscovery.table.fusion.GraceFusion.metadata
```

6.17.4.7 str_description

```
skdiscovery.framework.PipelineItem.str_description  [inherited]
```

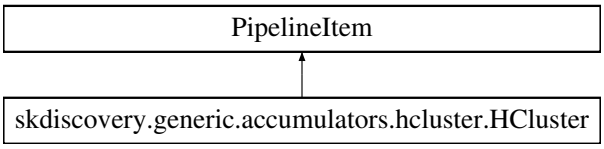
The documentation for this class was generated from the following file:

- table/fusion/grace.py

6.18 skdiscovery.generic.accumulators.HCluster Class Reference

Hierarchical Clustering function that produces a cluster map of the distance matrix.

Inheritance diagram for skdiscovery.generic.accumulators.HCluster:



Public Member Functions

- def __init__ (self, str_description, obj_name)

  Initialize HCluster.
- def process (self, obj_data)

  Produces a cluster map and stores the linkage results.

Public Attributes

- obj_name

6.18.1 Detailed Description

Hierarchical Clustering function that produces a cluster map of the distance matrix.

6.18.2 Constructor & Destructor Documentation

6.18.2.1 __init__()

```
def skdiscovery.generic.accumulators.HCluster.__init__ (
              self,
              str_description,
              obj_name )
```

Initialize HCluster.

Parameters

str_description	String describing accumulator
obj_name	Name of distance matrix parameter in the obj_data results

6.18.3 Member Function Documentation

6.18.3.1 process()

```
def skdiscovery.generic.accumulators.HCluster.process (
              self,
              obj_data )
```

Produces a cluster map and stores the linkage results.

Parameters

obj_data	Data wrapper

6.18.4 Member Data Documentation

6.18.4.1 obj_name

```
skdiscovery.generic.accumulators.HCluster.obj_name
```

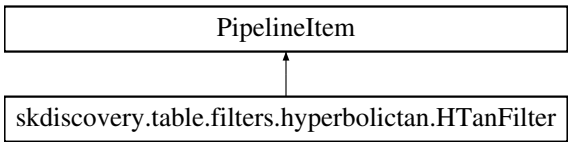
The documentation for this class was generated from the following file:

- generic/accumulators/hcluster.py

6.19 skdiscovery.table.filters.HTanFilter Class Reference

Filter to subtract an arctan fit from data.

Inheritance diagram for skdiscovery.table.filters.HTanFilter:



Public Member Functions

- def __init__ (self, str_description, t0, amplitude=5, timescale=1., offset=0, slope=0, labels=None, column_↩ names=None, start_time_limit=None, end_time_limit=None, start=None, end=None)

  Fit and remove hyperbolic tangent function from data.
- def process (self, obj_data)

  Apply Arctangent filter to data param.

Public Attributes

- a
- t0
- c
- slope
- offset
- labels
- column_names
- start_time_limit
- end_time_limit
- start
- end

6.19.1 Detailed Description

Filter to subtract an arctan fit from data.

6.19.2 Constructor & Destructor Documentation

6.19.2.1 __init__()

```
def skdiscovery.table.filters.HTanFilter.__init__ (
              self,
              str_description,
              t0,
              amplitude = 5,
              timescale = 1.,
              offset = 0,
              slope = 0,
              labels = None,
              column_names = None,
              start_time_limit = None,
              end_time_limit = None,
              start = None,
              end = None )
```

Fit and remove hyperbolic tangent function from data.

Parameters

str_description	String description of data
t0	Initial time offset of arctangent
amplitude	initial amplitude of arctangent
timescale	Timescale of fit
offset	Initial Y offset of arctangent
slope	Slope of the data
labels	Labels to apply arctangent function to
column_names	Column names to apply arctanget function to
start_time_limit	Starting time bound for fit to arctan (default: no bound)
end_time_limit	Ending time bound for fit to arctan (default: no bound)
start	Index of the first data point to fit (default: index of first data point)
end	Index of the last data point to fit (default: index of last data point)

6.19.3 Member Function Documentation

6.19.3.1 process()

```
def skdiscovery.table.filters.HTanFilter.process (
              self,
              obj_data )
```

Apply Arctangent filter to data param.

Parameters

obj_data	Input data. Changes are made in place.

6.19.4 Member Data Documentation

6.19.4.1 a

```
skdiscovery.table.filters.HTanFilter.a
```

6.19.4.2 c

```
skdiscovery.table.filters.HTanFilter.c
```

6.19.4.3 column_names

```
skdiscovery.table.filters.HTanFilter.column_names
```

6.19.4.4 end

```
skdiscovery.table.filters.HTanFilter.end
```

6.19.4.5 end_time_limit

```
skdiscovery.table.filters.HTanFilter.end_time_limit
```

6.19.4.6 labels

skdiscovery.table.filters.HTanFilter.labels

6.19.4.7 offset

skdiscovery.table.filters.HTanFilter.offset

6.19.4.8 slope

skdiscovery.table.filters.HTanFilter.slope

6.19.4.9 start

skdiscovery.table.filters.HTanFilter.start

6.19.4.10 start_time_limit

skdiscovery.table.filters.HTanFilter.start_time_limit

6.19.4.11 t0

skdiscovery.table.filters.HTanFilter.t0

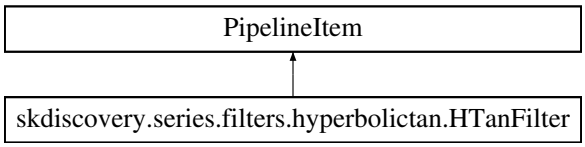
The documentation for this class was generated from the following file:

- table/filters/hyperbolictan.py

6.20 skdiscovery.series.filters.HTanFilter Class Reference

Filter to subtract arctan fit from data.

Inheritance diagram for skdiscovery.series.filters.HTanFilter:



Public Member Functions

- def __init__ (self, str_description, t0, amplitude=5, timescale=1., offset=0, slope=0, labels=None, column_↩ names=None, start_time_limit=None, end_time_limit=None, start=None, end=None)

  Fit and remove hyperbolic tangent function from data.
- def process (self, obj_data)

  Apply Arctangent filter to data param.

Public Attributes

- a
- t0
- c
- slope
- offset
- labels
- column_names
- start_time_limit
- end_time_limit
- start
- end

6.20.1 Detailed Description

Filter to subtract arctan fit from data.

[DEPRECATED] [will be removed]

6.20.2 Constructor & Destructor Documentation

6.20.2.1 __init__()

```
def skdiscovery.series.filters.HTanFilter.__init__ (
              self,
              str_description,
              t0,
              amplitude = 5,
              timescale = 1.,
              offset = 0,
              slope = 0,
              labels = None,
              column_names = None,
              start_time_limit = None,
              end_time_limit = None,
              start = None,
              end = None )
```

Fit and remove hyperbolic tangent function from data.

Parameters

str_description	String description of data
t0	Initial time offset of arctangent
amplitude	Initial amplitude of arctangent
timescale	Timescale of fit
offset	Initial Y offset of arctangent
slope	Slope of the data
labels	Labels to apply arctangent function to
column_names	Column names to apply arctanget function to
start_time_limit	Starting time bound for fit to arctan (default: no bound)
end_time_limit	Ending time bound for fit to arctan (default: no bound)
start	Index of the first data point to fit (default: index of first data point)
end	Index of the last data point to fit (default: index of last data point)

6.20.3 Member Function Documentation

6.20.3.1 process()

```
def skdiscovery.series.filters.HTanFilter.process (
              self,
              obj_data )
```

Apply Arctangent filter to data param.

Parameters

obj_data	Input data. Changes are made in place.

6.20.4 Member Data Documentation

6.20.4.1 a

```
skdiscovery.series.filters.HTanFilter.a
```

6.20.4.2 c

```
skdiscovery.series.filters.HTanFilter.c
```

6.20.4.3 column_names

```
skdiscovery.series.filters.HTanFilter.column_names
```

6.20.4.4 end

```
skdiscovery.series.filters.HTanFilter.end
```

6.20.4.5 end_time_limit

```
skdiscovery.series.filters.HTanFilter.end_time_limit
```

6.20.4.6 labels

skdiscovery.series.filters.HTanFilter.labels

6.20.4.7 offset

skdiscovery.series.filters.HTanFilter.offset

6.20.4.8 slope

skdiscovery.series.filters.HTanFilter.slope

6.20.4.9 start

skdiscovery.series.filters.HTanFilter.start

6.20.4.10 start_time_limit

skdiscovery.series.filters.HTanFilter.start_time_limit

6.20.4.11 t0

skdiscovery.series.filters.HTanFilter.t0

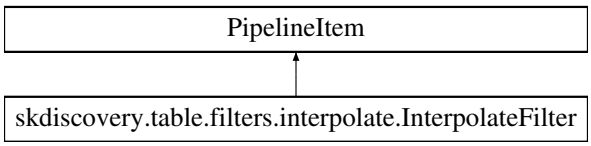
The documentation for this class was generated from the following file:

- series/filters/hyperbolictan.py

6.21 skdiscovery.table.filters.InterpolateFilter Class Reference

Interpolate missing values on table data.

Inheritance diagram for skdiscovery.table.filters.InterpolateFilter:



Public Member Functions

- def process (self, obj_data)

  Interpolate missing data in obj_data DataWrapper.

6.21.1 Detailed Description

Interpolate missing values on table data.

6.21.2 Member Function Documentation

6.21.2.1 process()

```
def skdiscovery.table.filters.InterpolateFilter.process (
              self,
              obj_data )
```

Interpolate missing data in obj_data DataWrapper.

Parameters

obj_data	Input DataWrapper, will be modified in place

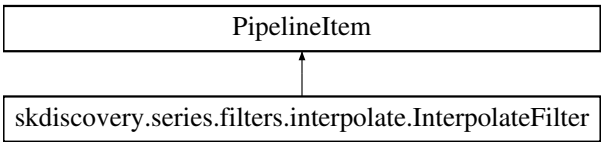
The documentation for this class was generated from the following file:

- table/filters/interpolate.py

6.22 skdiscovery.series.filters.InterpolateFilter Class Reference

Interpolate missing values on series data.

Inheritance diagram for skdiscovery.series.filters.InterpolateFilter:



Public Member Functions

- def process (self, obj_data)

  Interpolate missing data in obj_data DataWrapper.

6.22.1 Detailed Description

Interpolate missing values on series data.

6.22.2 Member Function Documentation

6.22.2.1 process()

```
def skdiscovery.series.filters.InterpolateFilter.process (
              self,
              obj_data )
```

Interpolate missing data in obj_data DataWrapper.

Parameters

obj_data	Input DataWrapper, will be modified in place

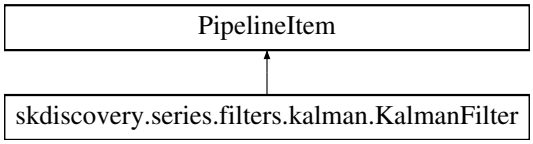
The documentation for this class was generated from the following file:

- series/filters/interpolate.py

6.23 skdiscovery.series.filters.KalmanFilter Class Reference

Runs a Kalman Smoother on series data.

Inheritance diagram for skdiscovery.series.filters.KalmanFilter:



Public Member Functions

- def __init__ (self, str_description, ap_paramList, uncertainty_clip=5)

  Initialize KalmanFilter.
- def process (self, obj_data)

  Apply kalman smoother to data set.

Public Attributes

- uncertainty_clip
- ap_paramNames

6.23.1 Detailed Description

Runs a Kalman Smoother on series data.

For more information see: Ji, K. H. 2011, PhD thesis, MIT.

6.23.2 Constructor & Destructor Documentation

6.23.2.1 __init__()

```
def skdiscovery.series.filters.KalmanFilter.__init__ (
              self,
              str_description,
              ap_paramList,
              uncertainty_clip = 5 )
```

Initialize KalmanFilter.

Parameters

str_description	String describing filter
ap_paramList[ap_tau]	the correlation time
ap_paramList[ap_sigmaSq]	the data noise
ap_paramList[ap_R]	the process noise
uncertainty_clip	Clip data with uncertainties greater than uncertainty_clip $*$ median uncertainty

6.23.3 Member Function Documentation

6.23.3.1 process()

```
def skdiscovery.series.filters.KalmanFilter.process (
              self,
              obj_data )
```

Apply kalman smoother to data set.

Parameters

obj_data	Input DataWrapper. Changes are made in place.

6.23.4 Member Data Documentation

6.23.4.1 ap_paramNames

```
skdiscovery.series.filters.KalmanFilter.ap_paramNames
```

6.23.4.2 uncertainty_clip

```
skdiscovery.series.filters.KalmanFilter.uncertainty_clip
```

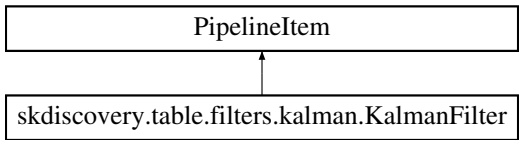
The documentation for this class was generated from the following file:

- series/filters/kalman.py

6.24 skdiscovery.table.filters.KalmanFilter Class Reference

Runs a Kalman Smoother on table data.

Inheritance diagram for skdiscovery.table.filters.KalmanFilter:



Public Member Functions

- def __init__ (self, str_description, ap_paramList, uncertainty_clip=5, column_names=None, error_column_names=None, fillna=True)

  Initialize KalmanFilter.
- def process (self, obj_data)

  Apply kalman smoother to data set.

Public Attributes

- uncertainty_clip
- ap_paramNames
- column_names
- error_column_names
- fillna

6.24.1 Detailed Description

Runs a Kalman Smoother on table data.

For more information see: Ji, K. H. 2011, PhD thesis, MIT.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 __init__()

```
def skdiscovery.table.filters.KalmanFilter.__init__ (
              self,
              str_description,
              ap_paramList,
              uncertainty_clip = 5,
              column_names = None,
              error_column_names = None,
              fillna = True )
```

Initialize KalmanFilter.

Parameters

str_description	String describing filter
ap_paramList[ap_tau]	the correlation time
ap_paramList[ap_sigmaSq]	the data noise
ap_paramList[ap_R]	the process noise
uncertainty_clip	Clip data with uncertainties greater than uncertainty_clip $*$ median uncertainty
column_names	List of column names to smooth (using None will apply to all columns)
error_column_names	List of error column names to smooth (using None will use default error columns)
fillna	Fill in missing values

6.24.3 Member Function Documentation

6.24.3.1 process()

```
def skdiscovery.table.filters.KalmanFilter.process (
              self,
              obj_data )
```

Apply kalman smoother to data set.

Parameters

obj_data	Input data. Changes are made in place.

6.24.4 Member Data Documentation

6.24.4.1 ap_paramNames

```
skdiscovery.table.filters.KalmanFilter.ap_paramNames
```

6.24.4.2 column_names

```
skdiscovery.table.filters.KalmanFilter.column_names
```

6.24.4.3 error_column_names

skdiscovery.table.filters.KalmanFilter.error_column_names

6.24.4.4 fillna

skdiscovery.table.filters.KalmanFilter.fillna

6.24.4.5 uncertainty_clip

skdiscovery.table.filters.KalmanFilter.uncertainty_clip

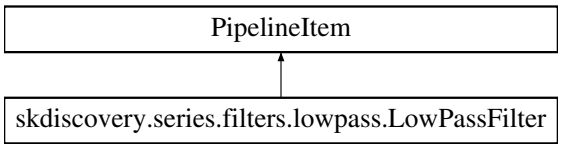
The documentation for this class was generated from the following file:

- table/filters/kalman.py

6.25 skdiscovery.series.filters.LowPassFilter Class Reference

A FIR Remez (Parks-McLellan) designed low pass filter for series data.

Inheritance diagram for skdiscovery.series.filters.LowPassFilter:



Public Member Functions

- def __init__ (self, str_description, ap_paramList)
  Initialize LowPassFilter.
- def process (self, obj_data)
  Apply lowpass filter to data set, with changes applied in place.

Public Attributes

- ap_paramNames

6.25.1 Detailed Description

A FIR Remez (Parks-McLellan) designed low pass filter for series data.

6.25.2 Constructor & Destructor Documentation

6.25.2.1 __init__()

```
def skdiscovery.series.filters.LowPassFilter.__init__ (
              self,
              str_description,
              ap_paramList )
```

Initialize LowPassFilter.

Parameters

str_description	String describing filter
ap_paramList[ntaps]	Number of filter taps
ap_paramList[fpassf_per]	Frequency passband ratio/percentage
ap_paramList[fstopf_per]	Frequency stopband ratio/percentage
ap_paramList[wghts]	Band importance weights
ap_paramList[miter]	Maximum number of iterations for generating the filter

6.25.3 Member Function Documentation

6.25.3.1 process()

```
def skdiscovery.series.filters.LowPassFilter.process (
              self,
              obj_data )
```

Apply lowpass filter to data set, with changes applied in place.

Parameters

obj_data	Input data with data

6.25.4 Member Data Documentation

6.25.4.1 ap_paramNames

```
skdiscovery.series.filters.LowPassFilter.ap_paramNames
```

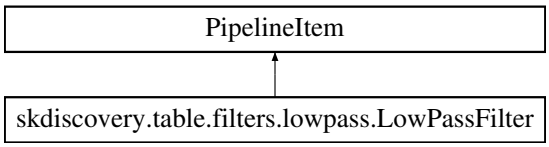
The documentation for this class was generated from the following file:

- series/filters/lowpass.py

6.26 skdiscovery.table.filters.LowPassFilter Class Reference

A remez low pass filter for table data.

Inheritance diagram for skdiscovery.table.filters.LowPassFilter:



Public Member Functions

- def __init__ (self, str_description, ap_paramList)
  Initialize LowPassFilter.
- def process (self, obj_data)
  Apply lowpass filter to data set.

Public Attributes

- ap_paramNames

6.26.1 Detailed Description

A remez low pass filter for table data.

6.26.2 Constructor & Destructor Documentation

6.26.2.1 __init__()

```
def skdiscovery.table.filters.LowPassFilter.__init__ (
              self,
              str_description,
              ap_paramList )
```

Initialize LowPassFilter.

Parameters

str_description	String describing filter
ap_paramList[ntaps]	number of filter taps
ap_paramList[fpassf_per]	frequency passband ratio/percentage
ap_paramList[fstopf_per]	frequency stopband ratio/percentage
ap_paramList[wghts]	band importance weights
ap_paramList[miter]	maximum number of iterations for generating the filter

6.26.3 Member Function Documentation

6.26.3.1 process()

```
def skdiscovery.table.filters.LowPassFilter.process (
              self,
              obj_data )
```

Apply lowpass filter to data set.

Parameters

obj_data	Input data. Changes are made in place.

6.26.4 Member Data Documentation

6.26.4.1 ap_paramNames

```
skdiscovery.table.filters.LowPassFilter.ap_paramNames
```

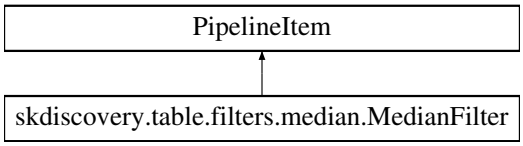
The documentation for this class was generated from the following file:

- table/filters/lowpass.py

6.27 skdiscovery.table.filters.MedianFilter Class Reference

A Median filter for table data.

Inheritance diagram for skdiscovery.table.filters.MedianFilter:



Public Member Functions

- def __init__ (self, str_description, ap_paramList, interpolate=True, subtract=False, regular_period=True, min_periods=1)

  Initialize MedianFilter.
- def process (self, obj_data)

  Apply median filter to data set.

Public Attributes

- interpolate
- subtract
- ap_paramNames
- regular_period
- min_periods

6.27.1 Detailed Description

A Median filter for table data.

6.27.2 Constructor & Destructor Documentation

6.27.2.1 __init__()

```
def skdiscovery.table.filters.MedianFilter.__init__ (
              self,
              str_description,
              ap_paramList,
              interpolate = True,
              subtract = False,
              regular_period = True,
              min_periods = 1 )
```

Initialize MedianFilter.

Parameters

str_description	String describing filter
ap_paramList[ap_window]	median filter window width
interpolate	Interpolate data points before filtering
subtract	Subtract filtered result from original
regular_period	Assume the data is regularly sampled
min_periods	Minimum required number of data points in window

6.27.3 Member Function Documentation

6.27.3.1 process()

```
def skdiscovery.table.filters.MedianFilter.process (
              self,
              obj_data )
```

Apply median filter to data set.

Parameters

obj_data	Input panda's data series. Changes are made in place.

6.27.4 Member Data Documentation

6.27.4.1 ap_paramNames

```
skdiscovery.table.filters.MedianFilter.ap_paramNames
```

6.27.4.2 interpolate

```
skdiscovery.table.filters.MedianFilter.interpolate
```

6.27.4.3 min_periods

skdiscovery.table.filters.MedianFilter.min_periods

6.27.4.4 regular_period

skdiscovery.table.filters.MedianFilter.regular_period

6.27.4.5 subtract

skdiscovery.table.filters.MedianFilter.subtract

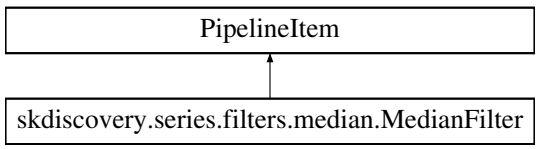
The documentation for this class was generated from the following file:

- table/filters/median.py

6.28 skdiscovery.series.filters.MedianFilter Class Reference

A Median filter for series data.

Inheritance diagram for skdiscovery.series.filters.MedianFilter:



Public Member Functions

- def __init__ (self, str_description, ap_paramList, interpolate=True, subtract=False)

  Initialize MedianFilter.
- def process (self, obj_data)

  Apply median filter to data set.

Public Attributes

- interpolate
- subtract
- ap_paramNames

6.28.1 Detailed Description

A Median filter for series data.

6.28.2 Constructor & Destructor Documentation

6.28.2.1 __init__()

```
def skdiscovery.series.filters.MedianFilter.__init__ (
              self,
              str_description,
              ap_paramList,
              interpolate = True,
              subtract = False )
```

Initialize MedianFilter.

Parameters

str_description	String describing filter
ap_paramList[ap_window]	median filter window width
interpolate	Flag to interpolate data points before filtering
subtract	Flag to subtract filtered result from original

6.28.3 Member Function Documentation

6.28.3.1 process()

```
def skdiscovery.series.filters.MedianFilter.process (
              self,
              obj_data )
```

Apply median filter to data set.

Parameters

obj_data	Input DataWrapper. Changes are made in place.

6.28.4 Member Data Documentation

6.28.4.1 ap_paramNames

`skdiscovery.series.filters.MedianFilter.ap_paramNames`

6.28.4.2 interpolate

`skdiscovery.series.filters.MedianFilter.interpolate`

6.28.4.3 subtract

`skdiscovery.series.filters.MedianFilter.subtract`

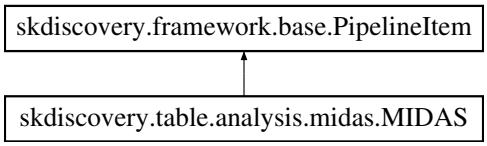
The documentation for this class was generated from the following file:

- series/filters/median.py

6.29 skdiscovery.table.analysis.MIDAS Class Reference

In A basic MIDAS trend estimator.

Inheritance diagram for skdiscovery.table.analysis.MIDAS:



Public Member Functions

- def __init__ (self, str_description, column_names=None)

  Initiatlize the MIDAS filtering item.
- def process (self, obj_data)

  Apply the MIDAS estimator to generate velocity estimates.
- def perturbParams (self)

  choose other random value for all parameters
- def resetParams (self)

  set all parameters to initial value
- def __str__ (self)

  String represntation of object.
- def getMetadata (self)

  Retrieve metadata about filter.

Public Attributes

- column_names
- str_description
- ap_paramList
- ap_paramNames

6.29.1 Detailed Description

In A basic MIDAS trend estimator.

See http://onlinelibrary.wiley.com/doi/10.1002/2015JB012552/full

6.29.2 Constructor & Destructor Documentation

6.29.2.1 __init__()

```
def skdiscovery.table.analysis.MIDAS.__init__ (
              self,
              str_description,
              column_names = None )
```

Initiatlize the MIDAS filtering item.

Parameters

str_description	String description of filter
column_names	List of column names to analyze

6.29.3 Member Function Documentation

6.29.3.1 __str__()

```
def skdiscovery.framework.PipelineItem.__str__ (
              self )  [inherited]
```

String represntation of object.

Returns

String listing all currenter parameters

6.29.3.2 getMetadata()

```
def skdiscovery.framework.PipelineItem.getMetadata (
              self )  [inherited]
```

Retrieve metadata about filter.

Returns

String containing the item description and current parameters for filter.

6.29.3.3 perturbParams()

```
def skdiscovery.framework.PipelineItem.perturbParams (
              self )  [inherited]
```

choose other random value for all parameters

6.29.3.4 process()

```
def skdiscovery.table.analysis.MIDAS.process (
              self,
              obj_data )
```

Apply the MIDAS estimator to generate velocity estimates.

Adds the result to the data wrapper

Parameters

obj_data	Data wrapper

6.29.3.5 resetParams()

```
def skdiscovery.framework.PipelineItem.resetParams (
              self )  [inherited]
```

set all parameters to initial value

6.29.4 Member Data Documentation

6.29.4.1 ap_paramList

```
skdiscovery.framework.PipelineItem.ap_paramList  [inherited]
```

6.29.4.2 ap_paramNames

```
skdiscovery.framework.PipelineItem.ap_paramNames  [inherited]
```

6.29.4.3 column_names

```
skdiscovery.table.analysis.MIDAS.column_names
```

6.29.4.4 str_description

```
skdiscovery.framework.PipelineItem.str_description  [inherited]
```

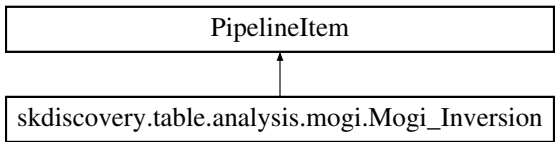
The documentation for this class was generated from the following file:

- table/analysis/midas.py

6.30 skdiscovery.table.analysis.Mogi_Inversion Class Reference

Perform a mogi source inversion on a set of gps table data.

Inheritance diagram for skdiscovery.table.analysis.Mogi_Inversion:



Public Member Functions

- def __init__ (self, str_description, ap_paramList, pca_name, column_names=['dN', dE, dU)

  Initialize Mogi analysis item.
- def FitPCA (self, hPCA_Proj)

  Determine the timing of the inflation event from the first component of the pca projection.
- def FitTimeSeries (self, pd_series, ct)

  Fits the amplitude and offset of an inflation event given the time and length of the event.
- def process (self, obj_data)

  Finds the magma source (default-mogi) from PBO GPS data.

Public Attributes

- pca_name
- column_names
- ap_paramNames

6.30.1 Detailed Description

Perform a mogi source inversion on a set of gps table data.

The source is assumed to be a mogi source (point source), but other source models can be selected. Assumes directions are named ('dN', 'dE', 'dU').

6.30.2 Constructor & Destructor Documentation

6.30.2.1 __init__()

```
def skdiscovery.table.analysis.Mogi_Inversion.__init__ (
              self,
              str_description,
              ap_paramList,
              pca_name,
              column_names = ['dN',
              dE,
              dU )
```

Initialize Mogi analysis item.

Parameters

str_description	Description of item
ap_paramList[source_type]	Type of magma chamber source model to use (default-mogi,finite_sphere,closed_pipe,constant_open_pipe,rising_open_pipe,sill)
pca_name	Name of pca result
column_names	The data direction column names

6.30.3 Member Function Documentation

6.30.3.1 FitPCA()

```
def skdiscovery.table.analysis.Mogi_Inversion.FitPCA (
              self,
              hPCA_Proj )
```

Determine the timing of the inflation event from the first component of the pca projection.

fits A ∗ arctan((t - t0) / c) + B to the first pca projection, in order to estimate source amplitude parameters

Parameters

hPCA_Proj	The sklearn PCA

Returns

ct: the t0, c, and B parameters from the fit
pA[0]: the fitted amplitude parameter

6.30.3.2 FitTimeSeries()

```
def skdiscovery.table.analysis.Mogi_Inversion.FitTimeSeries (
              self,
              pd_series,
              ct )
```

Fits the amplitude and offset of an inflation event given the time and length of the event.

Fits A and B in A ∗ arctan((t - t0) / c) + B

Parameters

pd_series	Time series to be fit
ct	the time constants for the arctan

Returns

res: Amplitude of the fit
perr_leastsq: Error of the fit

6.30.3.3 process()

```
def skdiscovery.table.analysis.Mogi_Inversion.process (
              self,
              obj_data )
```

Finds the magma source (default-mogi) from PBO GPS data.

Assumes time series columns are named ('dN', 'dE', 'dU'). Predicts the location of the magma source using scipy.↩ optimize.curve_fit

The result is added to the data wrapper as a list, with the four elements describing the location of the magma source: res[0] = latitude res[1] = longitude res[2] = source depth (km) res[3] = volume change (meters^3)

Parameters

obj_data	

6.30.4 Member Data Documentation

6.30.4.1 ap_paramNames

```
skdiscovery.table.analysis.Mogi_Inversion.ap_paramNames
```

6.30.4.2 column_names

```
skdiscovery.table.analysis.Mogi_Inversion.column_names
```

6.30.4.3 pca_name

`skdiscovery.table.analysis.Mogi_Inversion.pca_name`

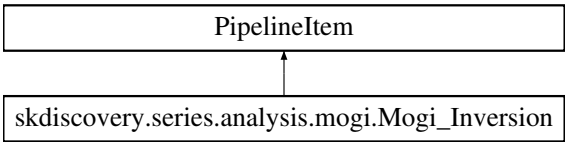
The documentation for this class was generated from the following file:

- table/analysis/mogi.py

6.31 skdiscovery.series.analysis.Mogi_Inversion Class Reference

Perform a Mogi source inversion on a set of gps series data.

Inheritance diagram for skdiscovery.series.analysis.Mogi_Inversion:



Public Member Functions

- def __init__ (self, str_description, ap_paramList)

  Initialize Mogi analysis item.
- def FitPCA (self, hPCA_Proj)

  Determine the timing of the inflation event.
- def FitTimeSeries (self, pd_series, ct)

  Fits the amplitude and offset of an inflation event given the time and length of the event.
- def process (self, obj_data)

  Finds the magma source (default-mogi) from PBO GPS data.

Public Attributes

- ap_paramNames

6.31.1 Detailed Description

Perform a Mogi source inversion on a set of gps series data.

The source is assumed to be a Mogi source (point source), but other source models can be selected. Assumes directions are named ('dN', 'dE', 'dU').

6.31.2 Constructor & Destructor Documentation

6.31.2.1 __init__()

```
def skdiscovery.series.analysis.Mogi_Inversion.__init__ (
              self,
              str_description,
              ap_paramList )
```

Initialize Mogi analysis item.

Parameters

str_description	Description of the item
ap_paramList[h_pca_name]	Name of the pca computed by General_Component_Analysis. Gets start and end date from the PCA fit.
ap_paramList[source_type]	Type of magma chamber source model to use (mogi [default],finite_sphere,closed_pipe,constant_open_pipe,rising_open_pipe,sill)

6.31.3 Member Function Documentation

6.31.3.1 FitPCA()

```
def skdiscovery.series.analysis.Mogi_Inversion.FitPCA (
              self,
              hPCA_Proj )
```

Determine the timing of the inflation event.

Uses the first component of the pca projection and fits A $*$ arctan((t - t0) / c) + B to the first pca projection.

Parameters

hPCA_Proj	The sklearn PCA projection

Returns

[t0, c]

6.31.3.2 FitTimeSeries()

```
def skdiscovery.series.analysis.Mogi_Inversion.FitTimeSeries (
              self,
              pd_series,
              ct )
```

Fits the amplitude and offset of an inflation event given the time and length of the event.

Fits A and B in A ∗ arctan((t - t0) / c) + B

Parameters

pd_series	Time series to be fit
ct	[t0, c]

Returns

Amplitude of fit

6.31.3.3 process()

```
def skdiscovery.series.analysis.Mogi_Inversion.process (
              self,
              obj_data )
```

Finds the magma source (default-mogi) from PBO GPS data.

Assumes time series columns are named ('dN', 'dE', 'dU'). Predicts location of the magma source using scipy.optimize.↩ curve_fit

The location of the magma source is stored in the data wrapper as a list res[0] = latitude res[1] = longitude res[2] = source depth (km) res[3] = volume change (meters^3) res[4] = extra parameters (depends on mogi fit type)

Parameters

obj_data	Data object containing the results from the PCA stage

6.31.4 Member Data Documentation

6.31.4.1 ap_paramNames

`skdiscovery.series.analysis.Mogi_Inversion.ap_paramNames`

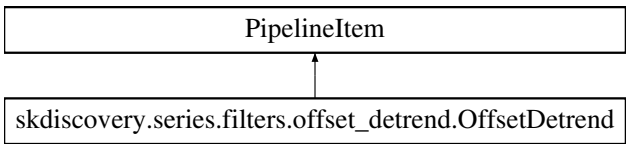
The documentation for this class was generated from the following file:

- series/analysis/mogi.py

6.32 skdiscovery.series.filters.OffsetDetrend Class Reference

Trend filter that fits a stepwise function to linearly detrended series data.

Inheritance diagram for skdiscovery.series.filters.OffsetDetrend:



Public Member Functions

- def __init__ (self, str_description, ap_paramList=[], labels=None, column_names=None, time_point=None, time_interval=None)

  Initialize OffsetDetrend filter.
- def process (self, obj_data)

  Apply offset estimation and detrending filter to data set.

Public Attributes

- labels
- column_names
- time_point
- time_interval
- ap_paramNames

6.32.1 Detailed Description

Trend filter that fits a stepwise function to linearly detrended series data.

On detrended data this filter fits a stepwise function (number of steps provided by the user) to correct the linear fit by accounting for discontinuous offsets, such as due to a change in the antenna or from an earthquake. The final linear fit handles each portion of the offset independently. If the number of discontinuities is not provided as an autoparam, the filter assumes a single discontinuity.

6.32.2 Constructor & Destructor Documentation

6.32.2.1 __init__()

```
def skdiscovery.series.filters.OffsetDetrend.__init__ (
              self,
              str_description,
              ap_paramList = [],
              labels = None,
              column_names = None,
              time_point = None,
              time_interval = None )
```

Initialize OffsetDetrend filter.

Parameters

str_description	String describing filter
ap_paramList[step_count]	Number of steps to remove from data (Default: 1)
labels	List of labels used to select data to be removed (using None will apply to all labels)
column_names	List of column names to select data to be removed (using None will apply to all columns)
time_point	Time of offset
time_interval	Interval within which the offset occurs

6.32.3 Member Function Documentation

6.32.3.1 process()

```
def skdiscovery.series.filters.OffsetDetrend.process (
              self,
              obj_data )
```

Apply offset estimation and detrending filter to data set.

Parameters

obj_data	Input data. Changes are made in place.

6.32.4 Member Data Documentation

6.32.4.1 ap_paramNames

skdiscovery.series.filters.OffsetDetrend.ap_paramNames

6.32.4.2 column_names

skdiscovery.series.filters.OffsetDetrend.column_names

6.32.4.3 labels

skdiscovery.series.filters.OffsetDetrend.labels

6.32.4.4 time_interval

skdiscovery.series.filters.OffsetDetrend.time_interval

6.32.4.5 time_point

skdiscovery.series.filters.OffsetDetrend.time_point

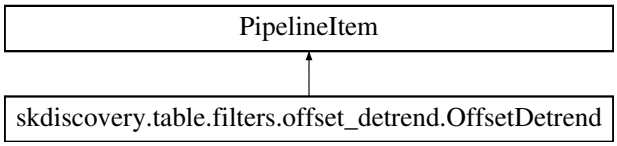
The documentation for this class was generated from the following file:

- series/filters/offset_detrend.py

6.33 skdiscovery.table.filters.OffsetDetrend Class Reference

Trend filter that fits a stepwise function to linearly detrended table data.

Inheritance diagram for skdiscovery.table.filters.OffsetDetrend:



Public Member Functions

- def __init__ (self, str_description, column_names, ap_paramList=[], labels=None, time_point=None, time_interval=None)

  Initialize OffsetDetrend filter for use on table data.
- def process (self, obj_data)

  Apply offset estimation and detrending filter to data set.

Public Attributes

- labels
- column_names
- time_point
- time_interval
- ap_paramNames

6.33.1 Detailed Description

Trend filter that fits a stepwise function to linearly detrended table data.

On detrended data this filter fits a stepwise function (number of steps provided by the user) to correct the linear fit by accounting for discontinuous offsets, such as due to a change in the antenna or from an earthquake. The final linear fit handles each portion of the offset independently. If the number of discontinuities is not provided as an autoparam, the filter assumes a single discontinuity.

6.33.2 Constructor & Destructor Documentation

6.33.2.1 __init__()

```
def skdiscovery.table.filters.OffsetDetrend.__init__ (
              self,
              str_description,
              column_names,
              ap_paramList = [],
              labels = None,
              time_point = None,
              time_interval = None )
```

Initialize OffsetDetrend filter for use on table data.

Parameters

str_description	String describing filter
column_names	List of column names to select data to be removed (using None will apply to all columns)
ap_paramList[step_count]	Number of steps to remove from data (Default: 1)
labels	List of labels used to select data to be removed (using None will apply to all labels)
time_point	Time of offset
time_interval	Interval within which the offset occurs

6.33.3 Member Function Documentation

6.33.3.1 process()

```
def skdiscovery.table.filters.OffsetDetrend.process (
              self,
              obj_data )
```

Apply offset estimation and detrending filter to data set.

Parameters

obj_data	Input data. Changes are made in place.

6.33.4 Member Data Documentation

6.33.4.1 ap_paramNames

```
skdiscovery.table.filters.OffsetDetrend.ap_paramNames
```

6.33.4.2 column_names

```
skdiscovery.table.filters.OffsetDetrend.column_names
```

6.33.4.3 labels

`skdiscovery.table.filters.OffsetDetrend.labels`

6.33.4.4 time_interval

`skdiscovery.table.filters.OffsetDetrend.time_interval`

6.33.4.5 time_point

`skdiscovery.table.filters.OffsetDetrend.time_point`

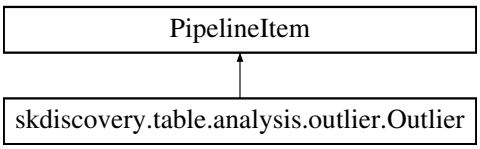
The documentation for this class was generated from the following file:

- table/filters/offset_detrend.py

6.34 skdiscovery.table.analysis.Outlier Class Reference

Computes (data / mad(data)) for outlier detection.

Inheritance diagram for skdiscovery.table.analysis.Outlier:



Public Member Functions

- def __init__ (self, str_description, columns=None, name_prefix='MAD_Scale_')

  Initalize Outlier Item.
- def process (self, obj_data)

  Process the data object to add a column with the outlier scores.

Public Attributes

- columns
- name_prefix

6.34.1 Detailed Description

Computes (data / mad(data)) for outlier detection.

Creates a new column for the result

6.34.2 Constructor & Destructor Documentation

6.34.2.1 __init__()

```
def skdiscovery.table.analysis.Outlier.__init__ (
              self,
              str_description,
              columns = None,
              name_prefix = 'MAD_Scale_' )
```

Initalize Outlier Item.

Parameters

str_description	Name of Item
columns	List of of column names
name_prefix	Prefix of newly created column

6.34.3 Member Function Documentation

6.34.3.1 process()

```
def skdiscovery.table.analysis.Outlier.process (
              self,
              obj_data )
```

Process the data object to add a column with the outlier scores.

Parameters

obj_data	Input table data wrapper

6.34.4 Member Data Documentation

6.34.4.1 columns

`skdiscovery.table.analysis.Outlier.columns`

6.34.4.2 name_prefix

`skdiscovery.table.analysis.Outlier.name_prefix`

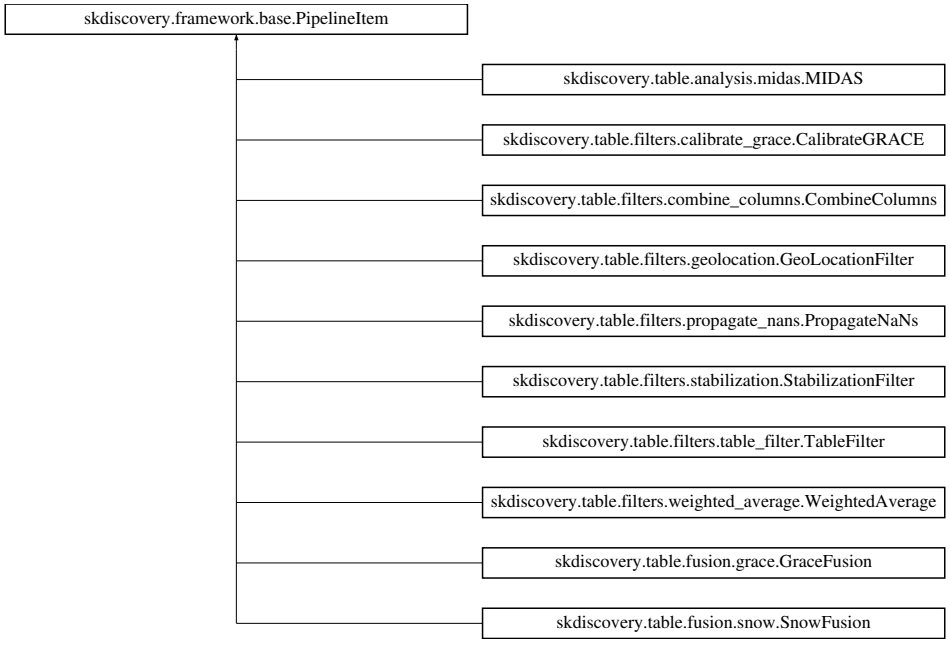
The documentation for this class was generated from the following file:

- table/analysis/outlier.py

6.35 skdiscovery.framework.PipelineItem Class Reference

The general class used to create pipeline items.

Inheritance diagram for skdiscovery.framework.PipelineItem:



Public Member Functions

- def __init__ (self, str_description, ap_paramList=[])
  Initialize an object.
- def perturbParams (self)
  choose other random value for all parameters
- def resetParams (self)
  set all parameters to initial value
- def process (self, obj_data)
  The actual filter processing.
- def __str__ (self)
  String represntation of object.
- def getMetadata (self)
  Retrieve metadata about filter.

Public Attributes

- str_description
- ap_paramList
- ap_paramNames

6.35.1 Detailed Description

The general class used to create pipeline items.

6.35.2 Constructor & Destructor Documentation

6.35.2.1 __init__()

```
def skdiscovery.framework.PipelineItem.__init__ (
              self,
              str_description,
              ap_paramList = [] )
```

Initialize an object.

Parameters

str_description	String description of filter
ap_paramList	List of AutoParam parameters.

6.35.3 Member Function Documentation

6.35.3.1 __str__()

```
def skdiscovery.framework.PipelineItem.__str__ (
              self )
```

String represntation of object.

Returns

String listing all currenter parameters

6.35.3.2 getMetadata()

```
def skdiscovery.framework.PipelineItem.getMetadata (
              self )
```

Retrieve metadata about filter.

Returns

String containing the item description and current parameters for filter.

6.35.3.3 perturbParams()

```
def skdiscovery.framework.PipelineItem.perturbParams (
              self )
```

choose other random value for all parameters

6.35.3.4 process()

```
def skdiscovery.framework.PipelineItem.process (
              self,
              obj_data )
```

The actual filter processing.

Empty in this generic filter.

```
   @param obj_data: Data wrapper that will be processed
```

6.35.3.5 resetParams()

```
def skdiscovery.framework.PipelineItem.resetParams (
              self )
```

set all parameters to initial value

6.35.4 Member Data Documentation

6.35.4.1 ap_paramList

```
skdiscovery.framework.PipelineItem.ap_paramList
```

6.35.4.2 ap_paramNames

```
skdiscovery.framework.PipelineItem.ap_paramNames
```

6.35.4.3 str_description

```
skdiscovery.framework.PipelineItem.str_description
```

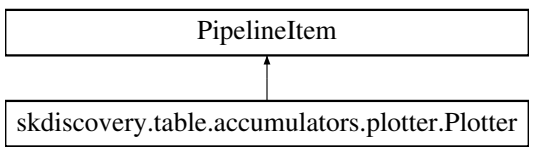
The documentation for this class was generated from the following file:

- framework/base.py

6.36 skdiscovery.table.accumulators.Plotter Class Reference

Make a plot of table data.

Inheritance diagram for skdiscovery.table.accumulators.Plotter:



Public Member Functions

- def __init__ (self, str_description, column_names=None, error_column_names=None, num_columns=3, width=13, height=4, columns_together=False, annotate_column=None, annotate_data=None, xlim=None, ylim=None, kwargs)

  Initialize Plotter.
- def process (self, obj_data)

  *Plot each column in obj_*

Public Attributes

- xlim
- ylim
- kwargs
- num_columns
- height
- width
- column_names
- annotate_column
- annotate_data
- error_column_names
- columns_together

6.36.1 Detailed Description

Make a plot of table data.

6.36.2 Constructor & Destructor Documentation

6.36.2.1 __init__()

```
def skdiscovery.table.accumulators.Plotter.__init__ (
              self,
              str_description,
              column_names = None,
              error_column_names = None,
              num_columns = 3,
              width = 13,
              height = 4,
              columns_together = False,
              annotate_column = None,
              annotate_data = None,
              xlim = None,
              ylim = None,
              kwargs )
```

Initialize Plotter.

Parameters

str_description	String describing accumulator
column_names	Columns to be plot
error_column_names	Columns containing uncertainties to be plot, no errorbars if None
num_columns	Number of columns to use when plotting data
width	Total width of all columns combined
height	Height of single row of plots
columns_together	If true, plot the columns on the same graph
annotate_column	Column of annotation data to use for annotation
annotate_data	Annotation data
xlim	The x limit
ylim	The y limit
∗∗*kwargs*	Any additional keyword arguments are passed on to matplotlib

6.36.3 Member Function Documentation

6.36.3.1 process()

```
def skdiscovery.table.accumulators.Plotter.process (
              self,
              obj_data )
```

Plot each column in obj_

Parameters

obj_data	Data Wrapper

6.36.4 Member Data Documentation

6.36.4.1 annotate_column

```
skdiscovery.table.accumulators.Plotter.annotate_column
```

6.36.4.2 annotate_data

skdiscovery.table.accumulators.Plotter.annotate_data

6.36.4.3 column_names

skdiscovery.table.accumulators.Plotter.column_names

6.36.4.4 columns_together

skdiscovery.table.accumulators.Plotter.columns_together

6.36.4.5 error_column_names

skdiscovery.table.accumulators.Plotter.error_column_names

6.36.4.6 height

skdiscovery.table.accumulators.Plotter.height

6.36.4.7 kwargs

skdiscovery.table.accumulators.Plotter.kwargs

6.36.4.8 num_columns

skdiscovery.table.accumulators.Plotter.num_columns

6.36.4.9 width

skdiscovery.table.accumulators.Plotter.width

6.36.4.10 xlim

skdiscovery.table.accumulators.Plotter.xlim

6.36.4.11 ylim

skdiscovery.table.accumulators.Plotter.ylim

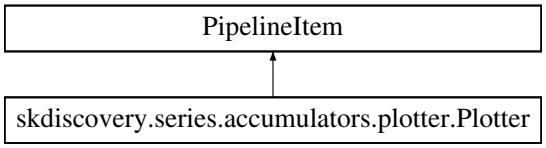
The documentation for this class was generated from the following file:

- table/accumulators/plotter.py

6.37 skdiscovery.series.accumulators.Plotter Class Reference

Make a plot of series data.

Inheritance diagram for skdiscovery.series.accumulators.Plotter:



Public Member Functions

- def __init__ (self, str_description, num_columns=3, errorbars=False, width=13, height=4, kwargs)
  Initialize Plotter.
- def process (self, obj_data)
  *Plot each column in obj_*

Public Attributes

- kwargs
- num_columns
- errorbars
- height
- width

6.37.1 Detailed Description

Make a plot of series data.

6.37.2 Constructor & Destructor Documentation

6.37.2.1 __init__()

```
def skdiscovery.series.accumulators.Plotter.__init__ (
              self,
              str_description,
              num_columns = 3,
              errorbars = False,
              width = 13,
              height = 4,
              kwargs )
```

Initialize Plotter.

Parameters

str_description	String describing accumulator
num_columns	Number of columns to use when plotting data
errorbars	Flag indicating if errorbars should be used
width	Total width of all columns combined
height	Height of single row of plots
∗∗*kwargs*	Any additional keyword arguments are passed on to matplotlib

6.37.3 Member Function Documentation

6.37.3.1 process()

```
def skdiscovery.series.accumulators.Plotter.process (
              self,
              obj_data )
```

Plot each column in obj_

Parameters

obj_data	Data Wrapper

6.37.4 Member Data Documentation

6.37.4.1 errorbars

`skdiscovery.series.accumulators.Plotter.errorbars`

6.37.4.2 height

`skdiscovery.series.accumulators.Plotter.height`

6.37.4.3 kwargs

`skdiscovery.series.accumulators.Plotter.kwargs`

6.37.4.4 num_columns

`skdiscovery.series.accumulators.Plotter.num_columns`

6.37.4.5 width

`skdiscovery.series.accumulators.Plotter.width`

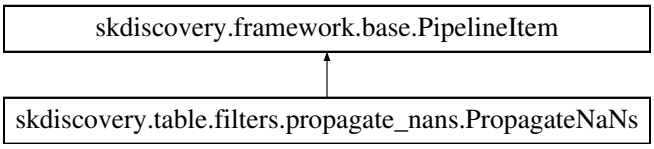
The documentation for this class was generated from the following file:

- series/accumulators/plotter.py

6.38 skdiscovery.table.filters.PropagateNaNs Class Reference

Propagates NaN's from one column to other columns.

Inheritance diagram for skdiscovery.table.filters.PropagateNaNs:



Public Member Functions

- def __init__ (self, str_description, nan_column, target_columns)

  Initialize PropagateNaNs Filter.
- def process (self, obj_data)

  PropagateNaNs on table data wrapper.
- def perturbParams (self)

  choose other random value for all parameters
- def resetParams (self)

  set all parameters to initial value
- def __str__ (self)

  String represntation of object.
- def getMetadata (self)

  Retrieve metadata about filter.

Public Attributes

- nan_column
- target_columns
- str_description
- ap_paramList
- ap_paramNames

6.38.1 Detailed Description

Propagates NaN's from one column to other columns.

6.38.2 Constructor & Destructor Documentation

6.38.2.1 __init__()

```
def skdiscovery.table.filters.PropagateNaNs.__init__ (
              self,
              str_description,
              nan_column,
              target_columns )
```

Initialize PropagateNaNs Filter.

Parameters

str_description	String describing filter
nan_column	Column used to select which rows should be NaN's
target_columns	Rows in these column will be set to NaN's based on nan_column

6.38.3 Member Function Documentation

6.38.3.1 __str__()

```
def skdiscovery.framework.PipelineItem.__str__ (
              self )  [inherited]
```

String represntation of object.

Returns

String listing all currenter parameters

6.38.3.2 getMetadata()

```
def skdiscovery.framework.PipelineItem.getMetadata (
              self )  [inherited]
```

Retrieve metadata about filter.

Returns

String containing the item description and current parameters for filter.

6.38.3.3 perturbParams()

```
def skdiscovery.framework.PipelineItem.perturbParams (
              self )  [inherited]
```

choose other random value for all parameters

6.38.3.4 process()

```
def skdiscovery.table.filters.PropagateNaNs.process (
              self,
              obj_data )
```

PropagateNaNs on table data wrapper.

Parameters

obj_data	Input table data wrapper

6.38.3.5 resetParams()

```
def skdiscovery.framework.PipelineItem.resetParams (
              self )  [inherited]
```

set all parameters to initial value

6.38.4 Member Data Documentation

6.38.4.1 ap_paramList

```
skdiscovery.framework.PipelineItem.ap_paramList  [inherited]
```

6.38.4.2 ap_paramNames

```
skdiscovery.framework.PipelineItem.ap_paramNames  [inherited]
```

6.38.4.3 nan_column

skdiscovery.table.filters.PropagateNaNs.nan_column

6.38.4.4 str_description

skdiscovery.framework.PipelineItem.str_description [inherited]

6.38.4.5 target_columns

skdiscovery.table.filters.PropagateNaNs.target_columns

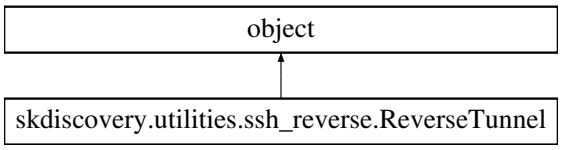
The documentation for this class was generated from the following file:

- table/filters/propagate_nans.py

6.39 skdiscovery.utilities.ssh_reverse.ReverseTunnel Class Reference

Create a reverse ssh tunnel.

Inheritance diagram for skdiscovery.utilities.ssh_reverse.ReverseTunnel:



Public Member Functions

- def __init__ (self, server_address, username, key_filename, server_port, remote_host, remote_port, check=30, verbose=False)

  Initialize ReverseTunnel object.
- def create_reverse_tunnel (self)

  Create the reverse tunnel.
- def __del__ (self)

  Deconstructor.

Public Attributes

- server_address
- username
- key_filename
- server_port
- remote_host
- remote_port
- check
- verbose
- ssh
- event
- child_threads

6.39.1 Detailed Description

Create a reverse ssh tunnel.

6.39.2 Constructor & Destructor Documentation

6.39.2.1 __init__()

```
def skdiscovery.utilities.ssh_reverse.ReverseTunnel.__init__ (
              self,
              server_address,
              username,
              key_filename,
              server_port,
              remote_host,
              remote_port,
              check = 30,
              verbose = False )
```

Initialize ReverseTunnel object.

Parameters

server_address	Local server address
username	Valid username on remote host
key_filename	Filename of ssh key associated with remote host
server_port	Local port
remote_host	Address of remote host
remote_port	Remote port
check	Amount of time to wait in seconds when opening up a channel
verbose	Print status information

6.39.2.2 __del__()

```
def skdiscovery.utilities.ssh_reverse.ReverseTunnel.__del__ (
              self )
```

Deconstructor.

6.39.3 Member Function Documentation

6.39.3.1 create_reverse_tunnel()

```
def skdiscovery.utilities.ssh_reverse.ReverseTunnel.create_reverse_tunnel (
              self )
```

Create the reverse tunnel.

6.39.4 Member Data Documentation

6.39.4.1 check

```
skdiscovery.utilities.ssh_reverse.ReverseTunnel.check
```

6.39.4.2 child_threads

```
skdiscovery.utilities.ssh_reverse.ReverseTunnel.child_threads
```

6.39.4.3 event

```
skdiscovery.utilities.ssh_reverse.ReverseTunnel.event
```

6.39.4.4 key_filename

skdiscovery.utilities.ssh_reverse.ReverseTunnel.key_filename

6.39.4.5 remote_host

skdiscovery.utilities.ssh_reverse.ReverseTunnel.remote_host

6.39.4.6 remote_port

skdiscovery.utilities.ssh_reverse.ReverseTunnel.remote_port

6.39.4.7 server_address

skdiscovery.utilities.ssh_reverse.ReverseTunnel.server_address

6.39.4.8 server_port

skdiscovery.utilities.ssh_reverse.ReverseTunnel.server_port

6.39.4.9 ssh

skdiscovery.utilities.ssh_reverse.ReverseTunnel.ssh

6.39.4.10 username

skdiscovery.utilities.ssh_reverse.ReverseTunnel.username

6.39.4.11 verbose

```
skdiscovery.utilities.ssh_reverse.ReverseTunnel.verbose
```

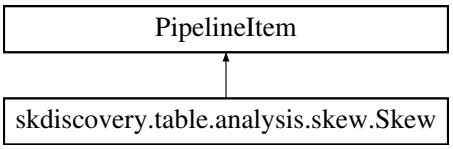
The documentation for this class was generated from the following file:

- utilities/ssh_reverse.py

6.40 skdiscovery.table.analysis.Skew Class Reference

Calculates the skew of table data.

Inheritance diagram for skdiscovery.table.analysis.Skew:



Public Member Functions

- def process (self, obj_data)
  Apply Skew analysis with results added to the data wrapper.

6.40.1 Detailed Description

Calculates the skew of table data.

6.40.2 Member Function Documentation

6.40.2.1 process()

```
def skdiscovery.table.analysis.Skew.process (
              self,
              obj_data )
```

Apply Skew analysis with results added to the data wrapper.

Parameters

obj_data	Data wrapper

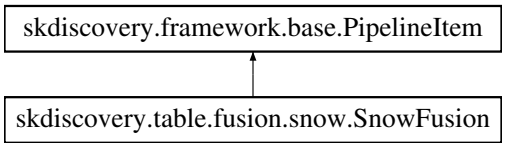
The documentation for this class was generated from the following file:

- table/analysis/skew.py

6.41 skdiscovery.table.fusion.SnowFusion Class Reference

Adds snow time series data to table based on geographic coordinates.

Inheritance diagram for skdiscovery.table.fusion.SnowFusion:



Public Member Functions

- def __init__ (self, str_description, metadata, column_data_name='Snow')

  Initialize Snow Fusion item.
- def process (self, obj_data)

  Adds column for snow (g02156) data.
- def perturbParams (self)

  choose other random value for all parameters
- def resetParams (self)

  set all parameters to initial value
- def __str__ (self)

  String represntation of object.
- def getMetadata (self)

  Retrieve metadata about filter.

Public Attributes

- metadata
- column_data_name
- str_description
- ap_paramList
- ap_paramNames

6.41.1 Detailed Description

Adds snow time series data to table based on geographic coordinates.

Works on table data (original data from http://nsidc.org/data/g02156)

6.41.2 Constructor & Destructor Documentation

6.41.2.1 __init__()

```
def skdiscovery.table.fusion.SnowFusion.__init__ (
              self,
              str_description,
              metadata,
              column_data_name = 'Snow' )
```

Initialize Snow Fusion item.

Parameters

str_description	String describing item
metadata	Metadata that contains lat,lon coordinates based on data labels
column_data_name	Name of column for Snow data

6.41.3 Member Function Documentation

6.41.3.1 __str__()

```
def skdiscovery.framework.PipelineItem.__str__ (
              self )  [inherited]
```

String represntation of object.

Returns

String listing all currenter parameters

6.41.3.2 getMetadata()

```
def skdiscovery.framework.PipelineItem.getMetadata (
              self )  [inherited]
```

Retrieve metadata about filter.

Returns

String containing the item description and current parameters for filter.

6.41.3.3 perturbParams()

```
def skdiscovery.framework.PipelineItem.perturbParams (
              self )  [inherited]
```

choose other random value for all parameters

6.41.3.4 process()

```
def skdiscovery.table.fusion.SnowFusion.process (
              self,
              obj_data )
```

Adds column for snow (g02156) data.

Parameters

obj_data	Input DataWrapper, will be modified in place

6.41.3.5 resetParams()

```
def skdiscovery.framework.PipelineItem.resetParams (
              self )  [inherited]
```

set all parameters to initial value

6.41.4 Member Data Documentation

6.41.4.1 ap_paramList

skdiscovery.framework.PipelineItem.ap_paramList [inherited]

6.41.4.2 ap_paramNames

skdiscovery.framework.PipelineItem.ap_paramNames [inherited]

6.41.4.3 column_data_name

skdiscovery.table.fusion.SnowFusion.column_data_name

6.41.4.4 metadata

skdiscovery.table.fusion.SnowFusion.metadata

6.41.4.5 str_description

skdiscovery.framework.PipelineItem.str_description [inherited]

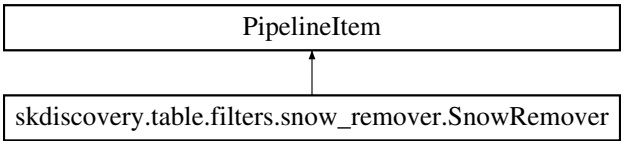
The documentation for this class was generated from the following file:

- table/fusion/snow.py

6.42 skdiscovery.table.filters.SnowRemover Class Reference

Removes data with snow errors.

Inheritance diagram for skdiscovery.table.filters.SnowRemover:



Public Member Functions

- def __init__ (self, str_description, ap_paramList=[AutoParam(1.5)], column_name='dN', snow_column='Snow')

  Initialize snow remover for use on table data.
- def process (self, obj_data)

  Removes table data with large snow errors.

Public Attributes

- column_name
- snow_column

6.42.1 Detailed Description

Removes data with snow errors.

6.42.2 Constructor & Destructor Documentation

6.42.2.1 __init__()

```
def skdiscovery.table.filters.SnowRemover.__init__ (
              self,
              str_description,
              ap_paramList = [AutoParam(1.5)],
              column_name = 'dN',
              snow_column = 'Snow' )
```

Initialize snow remover for use on table data.

Parameters

str_description	String describing filter
ap_paramList[sigma_clip]	remove station if the stddev of snowdays is sigma_clip times greater than non-snow days, default 1.5
column_name	Name of column to check
snow_column	Name of snow column to determine snowdays/non snow days

6.42.3 Member Function Documentation

6.42.3.1 process()

```
def skdiscovery.table.filters.SnowRemover.process (
              self,
              obj_data )
```

Removes table data with large snow errors.

Parameters

obj_data	Input DataWrapper, will be modified in place

6.42.4 Member Data Documentation

6.42.4.1 column_name

```
skdiscovery.table.filters.SnowRemover.column_name
```

6.42.4.2 snow_column

```
skdiscovery.table.filters.SnowRemover.snow_column
```

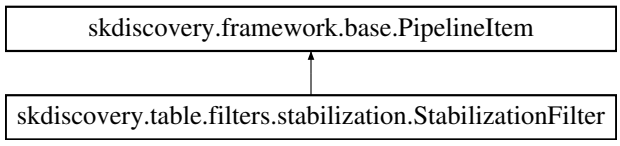
The documentation for this class was generated from the following file:

- table/filters/snow_remover.py

6.43 skdiscovery.table.filters.StabilizationFilter Class Reference

This filter transforms GPS stations in a region to a local reference frame.

Inheritance diagram for skdiscovery.table.filters.StabilizationFilter:



Public Member Functions

- def process (self, obj_data)

  Apply stabilization filter to data set.
- def perturbParams (self)

  choose other random value for all parameters
- def resetParams (self)

  set all parameters to initial value
- def __str__ (self)

  String represntation of object.
- def getMetadata (self)

  Retrieve metadata about filter.

Public Attributes

- str_description
- ap_paramList
- ap_paramNames

6.43.1 Detailed Description

This filter transforms GPS stations in a region to a local reference frame.

6.43.2 Member Function Documentation

6.43.2.1 __str__()

```
def skdiscovery.framework.PipelineItem.__str__ (
             self )  [inherited]
```

String represntation of object.

Returns

String listing all currenter parameters

6.43.2.2 getMetadata()

```
def skdiscovery.framework.PipelineItem.getMetadata (
              self )  [inherited]
```

Retrieve metadata about filter.

Returns

String containing the item description and current parameters for filter.

6.43.2.3 perturbParams()

```
def skdiscovery.framework.PipelineItem.perturbParams (
              self )  [inherited]
```

choose other random value for all parameters

6.43.2.4 process()

```
def skdiscovery.table.filters.StabilizationFilter.process (
              self,
              obj_data )
```

Apply stabilization filter to data set.

Parameters

obj_data	Table data wrapper.

6.43.2.5 resetParams()

```
def skdiscovery.framework.PipelineItem.resetParams (
              self )  [inherited]
```

set all parameters to initial value

6.43.3 Member Data Documentation

6.43.3.1 ap_paramList

```
skdiscovery.framework.PipelineItem.ap_paramList  [inherited]
```

6.43.3.2 ap_paramNames

```
skdiscovery.framework.PipelineItem.ap_paramNames  [inherited]
```

6.43.3.3 str_description

```
skdiscovery.framework.PipelineItem.str_description  [inherited]
```

The documentation for this class was generated from the following file:

- table/filters/stabilization.py

6.44 skdiscovery.framework.StageContainer Class Reference

Container to hold a stage for the DiscoveryPipeline.

Public Member Functions

- def __init__ (self, obj_content, obj_runmethod=None, obj_perturbmethod=None, obj_reset=None)

  Get the object and its run method into this conainer.
- def run (self, obj_data_container)

  Execute the obj_content run method.
- def perturb (self)

  Execute the obj_content peturb method.
- def reset (self)

  Execute the obj_content reset method.
- def getMetadata (self)

  Retrieves the obj_content metadata.
- def getObjects (self)

  Return the obj_content in a list.
- def getMetadataType (self)

  Get metadata type.
- def getMetadataNestedTypes (self)

  Get the metadata along with container type.
- def getMetadataNestedGraph (self)

  Get the nested graph for the container.

Public Attributes

- obj_content
- runmethod
- perturbmethod
- resetmethod

6.44.1 Detailed Description

Container to hold a stage for the DiscoveryPipeline.

6.44.2 Constructor & Destructor Documentation

6.44.2.1 __init__()

```
def skdiscovery.framework.StageContainer.__init__ (
              self,
              obj_content,
              obj_runmethod = None,
              obj_perturbmethod = None,
              obj_reset = None )
```

Get the object and its run method into this conainer.

Parameters

obj_content	filter, analysis, or accumlator
obj_runmethod	Run method of the obj_content (default process)
obj_perturbmethod	Perturb method of the obj_content (default peturbParams)
obj_reset	Reset method of the obj_content (default resetParams)

6.44.3 Member Function Documentation

6.44.3.1 getMetadata()

```
def skdiscovery.framework.StageContainer.getMetadata (
              self )
```

Retrieves the obj_content metadata.

Returns

obj_content metadata

6.44.3.2 getMetadataNestedGraph()

```
def skdiscovery.framework.StageContainer.getMetadataNestedGraph (
              self )
```

Get the nested graph for the container.

Returns

String: Stage container subgraph

6.44.3.3 getMetadataNestedTypes()

```
def skdiscovery.framework.StageContainer.getMetadataNestedTypes (
              self )
```

Get the metadata along with container type.

Returns

string of container and metadata

6.44.3.4 getMetadataType()

```
def skdiscovery.framework.StageContainer.getMetadataType (
              self )
```

Get metadata type.

Returns

String: container type

6.44.3.5 getObjects()

```
def skdiscovery.framework.StageContainer.getObjects (
              self )
```

Return the obj_content in a list.

Returns

Contained object in a list

6.44.3.6 perturb()

```
def skdiscovery.framework.StageContainer.perturb (
              self )
```

Execute the obj_content peturb method.

6.44.3.7 reset()

```
def skdiscovery.framework.StageContainer.reset (
              self )
```

Execute the obj_content reset method.

6.44.3.8 run()

```
def skdiscovery.framework.StageContainer.run (
              self,
              obj_data_container )
```

Execute the obj_content run method.

Parameters

obj_data_container	Data container to be passed to the held obj_content's run method

6.44.4 Member Data Documentation

6.44.4.1 obj_content

skdiscovery.framework.StageContainer.obj_content

6.44.4.2 perturbmethod

skdiscovery.framework.StageContainer.perturbmethod

6.44.4.3 resetmethod

skdiscovery.framework.StageContainer.resetmethod

6.44.4.4 runmethod

skdiscovery.framework.StageContainer.runmethod

The documentation for this class was generated from the following file:

- framework/stagecontainers.py

6.45 skdiscovery.framework.StageContainerAlternative Class Reference

Stage Container that holds a list of stage containers and randomly chooses one to use.

Public Member Functions

- def __init__ (self, list_stagecontainers)

  Initialize the StageContainerAlternative.
- def run (self, obj_data_container)

  Run the currently selected stage container.
- def perturb (self)

  choose one of the containers as an alternative and perturb its parameters
- def getMetadata (self)

  Return metadata from the current container.
- def getObjects (self)

  retrieve the current container as a list
- def reset (self)

  Reset the current chosen StageContainer.
- def getMetadataType (self)

  Get metadata type.
- def getMetadataNestedTypes (self)

  Get the metadata along with container type.
- def getMetadataNestedGraph (self)

  Get the nested graph for the container.

Public Attributes

- list_stagecontainers
- currentContainer

Static Public Attributes

- list currentContainer = [ ]

6.45.1 Detailed Description

Stage Container that holds a list of stage containers and randomly chooses one to use.

6.45.2 Constructor & Destructor Documentation

6.45.2.1 __init__()

```
def skdiscovery.framework.StageContainerAlternative.__init__ (
            self,
            list_stagecontainers )
```

Initialize the StageContainerAlternative.

Parameters

list_stagecontainers	List of stage containers

6.45.3 Member Function Documentation

6.45.3.1 getMetadata()

```
def skdiscovery.framework.StageContainerAlternative.getMetadata (
              self )
```

Return metadata from the current container.

Returns

metadata from the currently selected container

6.45.3.2 getMetadataNestedGraph()

```
def skdiscovery.framework.StageContainerAlternative.getMetadataNestedGraph (
              self )
```

Get the nested graph for the container.

Returns

String: Container subgraph

6.45.3.3 getMetadataNestedTypes()

```
def skdiscovery.framework.StageContainerAlternative.getMetadataNestedTypes (
              self )
```

Get the metadata along with container type.

Returns

string of container and metadata

6.45.3.4 getMetadataType()

```
def skdiscovery.framework.StageContainerAlternative.getMetadataType (
              self )
```

Get metadata type.

Returns

String: container type

6.45.3.5 getObjects()

```
def skdiscovery.framework.StageContainerAlternative.getObjects (
              self )
```

retrieve the current container as a list

Returns

Current container being used as a list

6.45.3.6 perturb()

```
def skdiscovery.framework.StageContainerAlternative.perturb (
              self )
```

choose one of the containers as an alternative and perturb its parameters

6.45.3.7 reset()

```
def skdiscovery.framework.StageContainerAlternative.reset (
              self )
```

Reset the current chosen StageContainer.

self.currentContainer.reset()

6.45.3.8 run()

```
def skdiscovery.framework.StageContainerAlternative.run (
              self,
              obj_data_container )
```

Run the currently selected stage container.

Parameters

obj_data_container	Data container to be passed to the current stagecontainer

6.45.4 Member Data Documentation

6.45.4.1 currentContainer [1/2]

```
list skdiscovery.framework.StageContainerAlternative.currentContainer = []  [static]
```

6.45.4.2 currentContainer [2/2]

```
skdiscovery.framework.StageContainerAlternative.currentContainer
```

6.45.4.3 list_stagecontainers

```
skdiscovery.framework.StageContainerAlternative.list_stagecontainers
```

The documentation for this class was generated from the following file:

- framework/stagecontainers.py

6.46 skdiscovery.framework.StageContainerIncrementalAdd Class Reference

In each perturb call, it incrementally adds one of the filters specified in the constructor.

Public Member Functions

- def __init__ (self, list_stagecontainers)

  Initialize the container.
- def reset (self)

  Reset the container so that it will only run the first stage container again.
- def run (self, obj_data_container)

  Run the current list of stage containers.
- def perturb (self)

  Add another stage container to the current list of stage containers.
- def getMetadata (self)

  Return the metadata from the currently used stage containers.
- def getObjects (self)

  Retrieve objects in the current list of stage containers.
- def getMetadataType (self)

  Get metadata type.
- def getMetadataNestedTypes (self)

  Get the metadata along with container type.
- def getMetadataNestedGraph (self)

  Get the nested graph for the container.

Public Attributes

- length
- list_AllStagecontainers
- list_currentContainers
- currentindex

Static Public Attributes

- int length = 0
- int currentindex = 0
- list list_currentContainers = [ ]

6.46.1 Detailed Description

In each perturb call, it incrementally adds one of the filters specified in the constructor.

6.46.2 Constructor & Destructor Documentation

6.46.2.1 __init__()

```
def skdiscovery.framework.StageContainerIncrementalAdd.__init__ (
              self,
              list_stagecontainers )
```

Initialize the container.

Parameters

list_stagecontainers	List of stage containers.

6.46.3 Member Function Documentation

6.46.3.1 getMetadata()

```
def skdiscovery.framework.StageContainerIncrementalAdd.getMetadata (
              self )
```

Return the metadata from the currently used stage containers.

Returns

List of metadata from current containers

6.46.3.2 getMetadataNestedGraph()

```
def skdiscovery.framework.StageContainerIncrementalAdd.getMetadataNestedGraph (
              self )
```

Get the nested graph for the container.

Returns

String: Container subgraph

6.46.3.3 getMetadataNestedTypes()

```
def skdiscovery.framework.StageContainerIncrementalAdd.getMetadataNestedTypes (
              self )
```

Get the metadata along with container type.

Returns

string of container and metadata

6.46.3.4 getMetadataType()

```
def skdiscovery.framework.StageContainerIncrementalAdd.getMetadataType (
              self )
```

Get metadata type.

Returns

String: container type

6.46.3.5 getObjects()

```
def skdiscovery.framework.StageContainerIncrementalAdd.getObjects (
              self )
```

Retrieve objects in the current list of stage containers.

Returns

List of current obj_content from the current list of stage containers

6.46.3.6 perturb()

```
def skdiscovery.framework.StageContainerIncrementalAdd.perturb (
              self )
```

Add another stage container to the current list of stage containers.

6.46.3.7 reset()

```
def skdiscovery.framework.StageContainerIncrementalAdd.reset (
              self )
```

Reset the container so that it will only run the first stage container again.

6.46.3.8 run()

```
def skdiscovery.framework.StageContainerIncrementalAdd.run (
              self,
              obj_data_container )
```

Run the current list of stage containers.

6.46.4 Member Data Documentation

6.46.4.1 currentindex [1/2]

```
int skdiscovery.framework.StageContainerIncrementalAdd.currentindex = 0  [static]
```

6.46.4.2 currentindex [2/2]

```
skdiscovery.framework.StageContainerIncrementalAdd.currentindex
```

6.46.4.3 length [1/2]

```
int skdiscovery.framework.StageContainerIncrementalAdd.length = 0  [static]
```

6.46.4.4 length [2/2]

```
skdiscovery.framework.StageContainerIncrementalAdd.length
```

6.46.4.5 list_AllStagecontainers

```
skdiscovery.framework.StageContainerIncrementalAdd.list_AllStagecontainers
```

6.46.4.6 list_currentContainers [1/2]

```
list skdiscovery.framework.StageContainerIncrementalAdd.list_currentContainers = []  [static]
```

6.46.4.7 list_currentContainers [2/2]

```
skdiscovery.framework.StageContainerIncrementalAdd.list_currentContainers
```

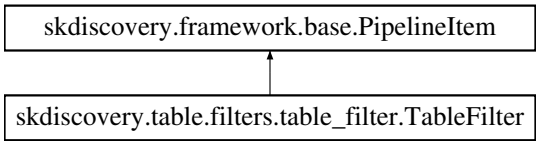
The documentation for this class was generated from the following file:

- framework/stagecontainers.py

6.47 skdiscovery.table.filters.TableFilter Class Reference

This class removes tables based on their label.

Inheritance diagram for skdiscovery.table.filters.TableFilter:



Public Member Functions

- def __init__ (self, str_description, ap_paramList)

  Initialize Table FIlter.
- def process (self, obj_data)

  Apply geolocation filter to data set.
- def perturbParams (self)

  choose other random value for all parameters
- def resetParams (self)

  set all parameters to initial value
- def __str__ (self)

  String represntation of object.
- def getMetadata (self)

  Retrieve metadata about filter.

Public Attributes

- str_description
- ap_paramList
- ap_paramNames

6.47.1 Detailed Description

This class removes tables based on their label.

6.47.2 Constructor & Destructor Documentation

6.47.2.1 __init__()

```
def skdiscovery.table.filters.TableFilter.__init__ (
              self,
              str_description,
              ap_paramList )
```

Initialize Table FIlter.

Parameters

str_description	String describing this filter
ap_paramList[ap_label_list]	AutoList of table labels to remove

6.47.3 Member Function Documentation

6.47.3.1 __str__()

```
def skdiscovery.framework.PipelineItem.__str__ (
              self )  [inherited]
```

String represntation of object.

Returns

String listing all currenter parameters

6.47.3.2 getMetadata()

```
def skdiscovery.framework.PipelineItem.getMetadata (
              self )  [inherited]
```

Retrieve metadata about filter.

Returns

String containing the item description and current parameters for filter.

6.47.3.3 perturbParams()

```
def skdiscovery.framework.PipelineItem.perturbParams (
              self )  [inherited]
```

choose other random value for all parameters

6.47.3.4 process()

```
def skdiscovery.table.filters.TableFilter.process (
              self,
              obj_data )
```

Apply geolocation filter to data set.

Parameters

obj_data	Table data wrapper

6.47.3.5 resetParams()

```
def skdiscovery.framework.PipelineItem.resetParams (
              self )  [inherited]
```

set all parameters to initial value

6.47.4 Member Data Documentation

6.47.4.1 ap_paramList

```
skdiscovery.framework.PipelineItem.ap_paramList  [inherited]
```

6.47.4.2 ap_paramNames

```
skdiscovery.framework.PipelineItem.ap_paramNames  [inherited]
```

6.47.4.3 str_description

```
skdiscovery.framework.PipelineItem.str_description  [inherited]
```

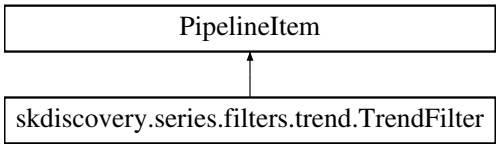
The documentation for this class was generated from the following file:

- table/filters/table_filter.py

6.48 skdiscovery.series.filters.TrendFilter Class Reference

Trend Filter that removes linear and sinusoidal (annual, semi-annual) trends on series data.

Inheritance diagram for skdiscovery.series.filters.TrendFilter:



Public Member Functions

- def __init__ (self, str_description, ap_paramList)
  Initialize Trend Filter.
- def process (self, obj_data)
  Apply trend filter to data set.

Public Attributes

- ap_paramNames

6.48.1 Detailed Description

Trend Filter that removes linear and sinusoidal (annual, semi-annual) trends on series data.

6.48.2 Constructor & Destructor Documentation

6.48.2.1 __init__()

```
def skdiscovery.series.filters.TrendFilter.__init__ (
              self,
              str_description,
              ap_paramList )
```

Initialize Trend Filter.

Parameters

str_description	String describing filter
ap_paramList[list_trendTypes]	List of trend types. List can contain any mix of "linear", "annual", or "semiannual". The default is to remove the linear, annual, and semiannual trends

6.48.3 Member Function Documentation

6.48.3.1 process()

```
def skdiscovery.series.filters.TrendFilter.process (
              self,
              obj_data )
```

Apply trend filter to data set.

Parameters

obj_data	Input data. Changes are made in place.

6.48.4 Member Data Documentation

6.48.4.1 ap_paramNames

`skdiscovery.series.filters.TrendFilter.ap_paramNames`

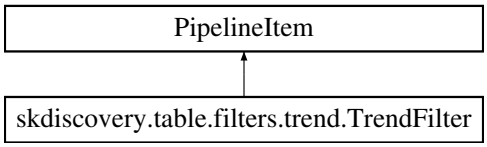
The documentation for this class was generated from the following file:

- series/filters/trend.py

6.49 skdiscovery.table.filters.TrendFilter Class Reference

Trend Filter that removes linear and sinusoidal (annual, semi-annual) trends on series data.

Inheritance diagram for skdiscovery.table.filters.TrendFilter:



Public Member Functions

- def __init__ (self, str_description, ap_paramList, columns=None)
  Initialize Trend Filter.
- def process (self, obj_data)
  Apply trend filter to data set.

Public Attributes

- columns
- ap_paramNames

6.49.1 Detailed Description

Trend Filter that removes linear and sinusoidal (annual, semi-annual) trends on series data.

Works on table data

6.49.2 Constructor & Destructor Documentation

6.49.2.1 __init__()

```
def skdiscovery.table.filters.TrendFilter.__init__ (
              self,
              str_description,
              ap_paramList,
              columns = None )
```

Initialize Trend Filter.

Parameters

str_description	String describing filter
ap_paramList[list_trendTypes]	List of trend types. List can contain "linear", "annual", or "semiannual"
columns	List of column names to filter

6.49.3 Member Function Documentation

6.49.3.1 process()

```
def skdiscovery.table.filters.TrendFilter.process (
              self,
              obj_data )
```

Apply trend filter to data set.

Parameters

obj_data	Input data. Changes are made in place.

6.49.4 Member Data Documentation

6.49.4.1 ap_paramNames

```
skdiscovery.table.filters.TrendFilter.ap_paramNames
```

6.49.4.2 columns

```
skdiscovery.table.filters.TrendFilter.columns
```

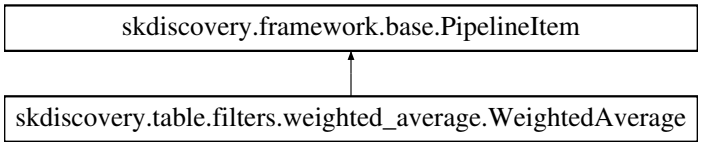
The documentation for this class was generated from the following file:

- table/filters/trend.py

6.50 skdiscovery.table.filters.WeightedAverage Class Reference

This filter performs a rolling weighted average using standard deviations as weight.

Inheritance diagram for skdiscovery.table.filters.WeightedAverage:



Public Member Functions

- def __init__ (self, str_description, ap_paramList, column_names, std_dev_column_names=None)

  Initializes a WeightedAverage object.
- def process (self, obj_data)

  Apply the moving (weighted) average filter to a table data wrapper.n.
- def perturbParams (self)

  choose other random value for all parameters
- def resetParams (self)

  set all parameters to initial value
- def __str__ (self)

  String represntation of object.
- def getMetadata (self)

  Retrieve metadata about filter.

Public Attributes

- column_names
- std_dev_column_names
- str_description
- ap_paramList
- ap_paramNames

6.50.1 Detailed Description

This filter performs a rolling weighted average using standard deviations as weight.

6.50.2 Constructor & Destructor Documentation

6.50.2.1 __init__()

```
def skdiscovery.table.filters.WeightedAverage.__init__ (
              self,
              str_description,
              ap_paramList,
              column_names,
              std_dev_column_names = None )
```

Initializes a WeightedAverage object.

Parameters

str_description	String describing filter
ap_paramList[window]	Window to use for computing rolling weighted average
column_names	Names of columns to apply the weighted average
std_dev_column_names	Names of columns of the standard deviations. If none a regular mean is computed.

6.50.3 Member Function Documentation

6.50.3.1 __str__()

```
def skdiscovery.framework.PipelineItem.__str__ (
              self )  [inherited]
```

String represntation of object.

Returns

String listing all currenter parameters

6.50.3.2 getMetadata()

```
def skdiscovery.framework.PipelineItem.getMetadata (
              self )  [inherited]
```

Retrieve metadata about filter.

Returns

String containing the item description and current parameters for filter.

6.50.3.3 perturbParams()

```
def skdiscovery.framework.PipelineItem.perturbParams (
              self )  [inherited]
```

choose other random value for all parameters

6.50.3.4 process()

```
def skdiscovery.table.filters.WeightedAverage.process (
              self,
              obj_data )
```

Apply the moving (weighted) average filter to a table data wrapper.n.

Changes are made in place.

Parameters

obj_data	Input table data wrapper

6.50.3.5 resetParams()

```
def skdiscovery.framework.PipelineItem.resetParams (
              self )  [inherited]
```

set all parameters to initial value

6.50.4 Member Data Documentation

6.50.4.1 ap_paramList

```
skdiscovery.framework.PipelineItem.ap_paramList  [inherited]
```

6.50.4.2 ap_paramNames

skdiscovery.framework.PipelineItem.ap_paramNames [inherited]

6.50.4.3 column_names

skdiscovery.table.filters.WeightedAverage.column_names

6.50.4.4 std_dev_column_names

skdiscovery.table.filters.WeightedAverage.std_dev_column_names

6.50.4.5 str_description

skdiscovery.framework.PipelineItem.str_description [inherited]

The documentation for this class was generated from the following file:

- table/filters/weighted_average.py

Chapter 7

File Documentation

7.1 framework/base.py File Reference

Classes

- class skdiscovery.framework.PipelineItem

  The general class used to create pipeline items.

Namespaces

- skdiscovery.framework.base

7.2 framework/discoverypipeline.py File Reference

Classes

- class skdiscovery.DiscoveryPipeline

  Pipeline for running the analysis.

Namespaces

- skdiscovery.framework.discoverypipeline

7.3 framework/stagecontainers.py File Reference

Classes

- class skdiscovery.framework.StageContainer

  Container to hold a stage for the DiscoveryPipeline.
- class skdiscovery.framework.StageContainerAlternative

  Stage Container that holds a list of stage containers and randomly chooses one to use.
- class skdiscovery.framework.StageContainerIncrementalAdd

  In each perturb call, it incrementally adds one of the filters specified in the constructor.

Namespaces

- skdiscovery.framework.stagecontainers

7.4 generic/accumulators/data.py File Reference

Classes

- class skdiscovery.generic.accumulators.DataAccumulator

  Stores a copy of the data in its current state in the pipeline.

Namespaces

- skdiscovery.generic.accumulators.data

7.5 generic/accumulators/gpshplotter.py File Reference

Classes

- class skdiscovery.generic.accumulators.GPSHPlotter

  Plots results from General_Component_Analysis, for the GPS horizontal or vertical components.

Namespaces

- skdiscovery.generic.accumulators.gpshplotter

7.6 generic/accumulators/hcluster.py File Reference

Classes

- class skdiscovery.generic.accumulators.HCluster

  Hierarchical Clustering function that produces a cluster map of the distance matrix.

Namespaces

- skdiscovery.generic.accumulators.hcluster

7.7 series/accumulators/plotter.py File Reference

Classes

- class skdiscovery.series.accumulators.Plotter

  Make a plot of series data.

Namespaces

- skdiscovery.series.accumulators.plotter

7.8 table/accumulators/plotter.py File Reference

Classes

- class skdiscovery.table.accumulators.Plotter

  Make a plot of table data.

Namespaces

- skdiscovery.table.accumulators.plotter

7.9 series/analysis/correlate.py File Reference

Classes

- class skdiscovery.series.analysis.Correlate

  Computes the correlation for series data.

Namespaces

- skdiscovery.series.analysis.correlate

7.10 table/analysis/correlate.py File Reference

Classes

- class skdiscovery.table.analysis.Correlate

  Computes the correlation for table data and stores the result as a matrix.

Namespaces

- skdiscovery.table.analysis.correlate

7.11 series/analysis/gca.py File Reference

Classes

- class skdiscovery.series.analysis.General_Component_Analysis

  Performs either ICA or PCA analysis on series data.

Namespaces

- skdiscovery.series.analysis.gca

7.12 table/analysis/gca.py File Reference

Classes

- class skdiscovery.table.analysis.General_Component_Analysis

  Performs a general component analysis on table data.

Namespaces

- skdiscovery.table.analysis.gca

7.13 series/analysis/mogi.py File Reference

Classes

- class skdiscovery.series.analysis.Mogi_Inversion

  Perform a Mogi source inversion on a set of gps series data.

Namespaces

- skdiscovery.series.analysis.mogi

Functions

- def skdiscovery.series.analysis.MogiVectors (mogi_res, station_lat_list, station_lon_list, flag3D=False)

  Creates a set of Mogi vectors for plotting.

7.14 table/analysis/mogi.py File Reference

Classes

- class skdiscovery.table.analysis.Mogi_Inversion

  Perform a mogi source inversion on a set of gps table data.

Namespaces

- skdiscovery.table.analysis.mogi

Functions

- def skdiscovery.table.analysis.MogiVectors (mogi_res, station_lat_list, station_lon_list, flag3D=False)

  Creates a set of mogi vectors for plotting.

7.15 series/filters/dataremover.py File Reference

Classes

- class skdiscovery.series.filters.DataRemover

  Sets specified series data to NaN.

Namespaces

- skdiscovery.series.filters.dataremover

7.16 table/filters/dataremover.py File Reference

Classes

- class skdiscovery.table.filters.DataRemover

  Sets specified table data to NaN.

Namespaces

- skdiscovery.table.filters.dataremover

7.17 series/filters/hyperbolictan.py File Reference

Classes

- class skdiscovery.series.filters.HTanFilter

  Filter to subtract arctan fit from data.

Namespaces

- skdiscovery.series.filters.hyperbolictan

7.18 table/filters/hyperbolictan.py File Reference

Classes

- class skdiscovery.table.filters.HTanFilter

  Filter to subtract an arctan fit from data.

Namespaces

- skdiscovery.table.filters.hyperbolictan

7.19 series/filters/interpolate.py File Reference

Classes

- class skdiscovery.series.filters.InterpolateFilter
  Interpolate missing values on series data.

Namespaces

- skdiscovery.series.filters.interpolate

7.20 table/filters/interpolate.py File Reference

Classes

- class skdiscovery.table.filters.InterpolateFilter
  Interpolate missing values on table data.

Namespaces

- skdiscovery.table.filters.interpolate

7.21 series/filters/kalman.py File Reference

Classes

- class skdiscovery.series.filters.KalmanFilter
  Runs a Kalman Smoother on series data.

Namespaces

- skdiscovery.series.filters.kalman

7.22 table/filters/kalman.py File Reference

Classes

- class skdiscovery.table.filters.KalmanFilter
  Runs a Kalman Smoother on table data.

Namespaces

- skdiscovery.table.filters.kalman

7.23 series/filters/lowpass.py File Reference

Classes

- class skdiscovery.series.filters.LowPassFilter

  A FIR Remez (Parks-McLellan) designed low pass filter for series data.

Namespaces

- skdiscovery.series.filters.lowpass

7.24 table/filters/lowpass.py File Reference

Classes

- class skdiscovery.table.filters.LowPassFilter

  A remez low pass filter for table data.

Namespaces

- skdiscovery.table.filters.lowpass

7.25 series/filters/median.py File Reference

Classes

- class skdiscovery.series.filters.MedianFilter

  A Median filter for series data.

Namespaces

- skdiscovery.series.filters.median

7.26 table/filters/median.py File Reference

Classes

- class skdiscovery.table.filters.MedianFilter

  A Median filter for table data.

Namespaces

- skdiscovery.table.filters.median

7.27 series/filters/offset_detrend.py File Reference

Classes

- class skdiscovery.series.filters.OffsetDetrend

  Trend filter that fits a stepwise function to linearly detrended series data.

Namespaces

- skdiscovery.series.filters.offset_detrend

7.28 table/filters/offset_detrend.py File Reference

Classes

- class skdiscovery.table.filters.OffsetDetrend

  Trend filter that fits a stepwise function to linearly detrended table data.

Namespaces

- skdiscovery.table.filters.offset_detrend

7.29 series/filters/trend.py File Reference

Classes

- class skdiscovery.series.filters.TrendFilter

  Trend Filter that removes linear and sinusoidal (annual, semi-annual) trends on series data.

Namespaces

- skdiscovery.series.filters.trend

7.30 table/filters/trend.py File Reference

Classes

- class skdiscovery.table.filters.TrendFilter

  Trend Filter that removes linear and sinusoidal (annual, semi-annual) trends on series data.

Namespaces

- skdiscovery.table.filters.trend

7.31 table/analysis/dbscan.py File Reference

Classes

- class skdiscovery.table.analysis.DBScan

  Runs DBScan on table data.

Namespaces

- skdiscovery.table.analysis.dbscan

7.32 table/analysis/midas.py File Reference

Classes

- class skdiscovery.table.analysis.MIDAS

  In A basic MIDAS trend estimator.

Namespaces

- skdiscovery.table.analysis.midas

7.33 table/analysis/outlier.py File Reference

Classes

- class skdiscovery.table.analysis.Outlier

  Computes (data / mad(data)) for outlier detection.

Namespaces

- skdiscovery.table.analysis.outlier

7.34 table/analysis/skew.py File Reference

Classes

- class skdiscovery.table.analysis.Skew

  Calculates the skew of table data.

Namespaces

- skdiscovery.table.analysis.skew

7.35 table/filters/antenna_offset.py File Reference

Classes

- class skdiscovery.table.filters.AntennaOffset

  Applies corrections to fix offsets in PBO GPS data induced by antenna changes.

Namespaces

- skdiscovery.table.filters.antenna_offset

7.36 table/filters/calibrate_py File Reference

Classes

- class skdiscovery.table.filters.CalibrateGRACE

  Calibrate Grace Data.

Namespaces

- skdiscovery.table.filters.calibrate_grace

7.37 table/filters/combine_columns.py File Reference

Classes

- class skdiscovery.table.filters.CombineColumns

  Create a new column by selecting data from a column.

Namespaces

- skdiscovery.table.filters.combine_columns

7.38 table/filters/geolocation.py File Reference

Classes

- class skdiscovery.table.filters.GeoLocationFilter

  Removes objects not located in a specified region.

Namespaces

- skdiscovery.table.filters.geolocation

7.39 table/filters/propagate_nans.py File Reference

Classes

- class skdiscovery.table.filters.PropagateNaNs

  Propagates NaN's from one column to other columns.

Namespaces

- skdiscovery.table.filters.propagate_nans

7.40 table/filters/snow_remover.py File Reference

Classes

- class skdiscovery.table.filters.SnowRemover
  Removes data with snow errors.

Namespaces

- skdiscovery.table.filters.snow_remover

7.41 table/filters/stabilization.py File Reference

Classes

- class skdiscovery.table.filters.StabilizationFilter
  This filter transforms GPS stations in a region to a local reference frame.

Namespaces

- skdiscovery.table.filters.stabilization

7.42 table/filters/table_filter.py File Reference

Classes

- class skdiscovery.table.filters.TableFilter
  This class removes tables based on their label.

Namespaces

- skdiscovery.table.filters.table_filter

7.43 table/filters/weighted_average.py File Reference

Classes

- class skdiscovery.table.filters.WeightedAverage
  This filter performs a rolling weighted average using standard deviations as weight.

Namespaces

- skdiscovery.table.filters.weighted_average

7.44 table/fusion/grace.py File Reference

Classes

- class skdiscovery.table.fusion.GraceFusion

  Fuses GRACE equivelent water depth time series.

Namespaces

- skdiscovery.table.fusion.grace

7.45 table/fusion/snow.py File Reference

Classes

- class skdiscovery.table.fusion.SnowFusion

  Adds snow time series data to table based on geographic coordinates.

Namespaces

- skdiscovery.table.fusion.snow

7.46 table/generators/catalog_generator.py File Reference

Classes

- class skdiscovery.table.generators.CatalogGenerator

  In Generates galaxy catalogs for use in DiscoveryPipeline.

Namespaces

- skdiscovery.table.generators.catalog_generator

7.47 table/generators/data_generator.py File Reference

Classes

- class skdiscovery.table.generators.DataGenerator

  In Class for generating random data.

Namespaces

- skdiscovery.table.generators.data_generator

7.48 utilities/amazon_control.py File Reference

Namespaces

- skdiscovery.utilities.amazon_control

Functions

- def skdiscovery.utilities.amazon_control.init (in_aws_access_key, in_aws_secret, in_aws_region, in_aws_↩ security_group, in_aws_key_name, in_pem_file)

  The underlying functionality for the Amazon GUI, the user should not need to directly interface with this function.
- def skdiscovery.utilities.amazon_control.closeDispyScheduler ()

  Close the Dispy Scheduler.
- def skdiscovery.utilities.amazon_control.startDispyScheduler ()

  Start the Dispy Scheduler.
- def skdiscovery.utilities.amazon_control.generateInfo (instance)

  Read metadata from an Amazon instance.
- def skdiscovery.utilities.amazon_control.updateStatus ()

  Update status information in amazon_list.
- def skdiscovery.utilities.amazon_control.setNumInstances (new_total_instances, instance_type, image_id)

  Change the number of running instances.
- def skdiscovery.utilities.amazon_control.createTunnels ()

  Create reverse ssh tunnels to all instances.
- def skdiscovery.utilities.amazon_control.startDispyNode ()

  Start dispy on each Amazon instance.
- def skdiscovery.utilities.amazon_control.resetInstances ()

  Reboot Amazon instances.
- def skdiscovery.utilities.amazon_control.reset ()

  Close and clear Amazon List.
- def skdiscovery.utilities.amazon_control.close ()

  Shutdown all instances, close dispy scheduler and clear Amazon list.
- def skdiscovery.utilities.amazon_control.clearAmazonList ()

  Shutdown connection tunnels to Amazon instances and clear amazon list.

Variables

- skdiscovery.utilities.amazon_control.aws_access_key = None
- skdiscovery.utilities.amazon_control.aws_secret = None
- skdiscovery.utilities.amazon_control.aws_region = None
- skdiscovery.utilities.amazon_control.aws_security_group = None
- skdiscovery.utilities.amazon_control.aws_key_name = None
- skdiscovery.utilities.amazon_control.pem_file = None
- skdiscovery.utilities.amazon_control.ec2_res = None
- skdiscovery.utilities.amazon_control.ec2_client = None
- list skdiscovery.utilities.amazon_control.amazon_list = [ ]
- skdiscovery.utilities.amazon_control.scheduler = None
- skdiscovery.utilities.amazon_control.popen = None

7.49 utilities/amazon_gui.py File Reference

Namespaces

- skdiscovery.utilities.amazon_gui

Functions

- def skdiscovery.utilities.amazon_gui.init ()

  Initialize GUI for controlling Amazon instances.
- def skdiscovery.utilities.amazon_gui.drawGUI ()

  Draw the GUI on the screen.
- def skdiscovery.utilities.amazon_gui.changeButtonState (enabled=True)

  Enable or disable the buttons and slider in the GUI.
- def skdiscovery.utilities.amazon_gui.checkValidValues ()

  Check if Amazon information is valid.

Variables

- skdiscovery.utilities.amazon_gui.widget_dict = OrderedDict()
- list skdiscovery.utilities.amazon_gui.disable_list
- list skdiscovery.utilities.amazon_gui.key_value_list

7.50 utilities/astro_tools.py File Reference

Namespaces

- skdiscovery.utilities.astro_tools

Functions

- def skdiscovery.utilities.astro_tools.z_to_v (z)

  Convert redshift to km/s assuming shift is due to velocity using special relativity.
- def skdiscovery.utilities.astro_tools.v_to_z (v)

  Convert km/s to redshift assuming all are using special relativity.
- def skdiscovery.utilities.astro_tools.angular_separation (ra1, dec1, ra2, dec2)

  Angular seperation between two objects via the haversine formula.
- def skdiscovery.utilities.astro_tools.move_point (ra, dec, ang_dist, bearing)

  Move a point along a great circle at a particular bearing.
- def skdiscovery.utilities.astro_tools.abs_mag (app_mag, z)

  Get the absolute magnitude from apparent magnitude.
- def skdiscovery.utilities.astro_tools.app_mag (abs_mag, z)

  Get the apparent magnitude from absolute magnitude.
- def skdiscovery.utilities.astro_tools.nfw (R, norm_constant, Rs, Rcore)

  2D Navarro-Frenk-White surface radial profile probability density
- def skdiscovery.utilities.astro_tools.lf (x, A, mstar, alpha)

  Schechter function.
- def skdiscovery.utilities.astro_tools.dlf (x, A, m1, a1, m2, a2)

  double Schechter function.
- def skdiscovery.utilities.astro_tools.cdf_dlf (x, A, m1, a1, m2, a2, start=-26)

  Cumulative Schechter function.
- def skdiscovery.utilities.astro_tools.inv_cdf_dlf (p, A, m1, a1, m2, a2, start=-26, end=-15)

  Inverse Cumulative Schechter function.

7.51 utilities/config.py File Reference

Namespaces

- skdiscovery.utilities.config

Functions

- def skdiscovery.utilities.config.getConfig ()

  Retrieve skdiscovery configuaration.
- def skdiscovery.utilities.config.writeConfigValue (section, key, value)

  Write config to disk.
- def skdiscovery.utilities.config.getDispyPassword ()

  Get dispy password.
- def skdiscovery.utilities.config.getHostName ()

  Get Host name for displaying link to dispy status.

7.52 utilities/kalman_smoother.py File Reference

Namespaces

- skdiscovery.utilities.kalman_smoother

Functions

- def skdiscovery.utilities.kalman_smoother.KalmanFilter (in_data, t, sigma_sq, R, Pinit, x0=0, invert=False, clipping=5)

  Runs the kalman filter on data.
- def skdiscovery.utilities.kalman_smoother.FitFOGMParameters (data, Pinit=100, R=1, method='brute', x0=0, clipping=5)

  Find best FOGM parameters for a given data set.
- def skdiscovery.utilities.kalman_smoother.IterativeGridSearch (f, args, intervals, max_iter=50, tol=0.↩1, bounds=None, prev_minimum=None, verbose=False)

  Find the minimum of f using an iterative grid search with 3 points per dimension.
- def skdiscovery.utilities.kalman_smoother.KalmanSmoother (in_data, Pinit=1e6, Restimate=1, clipping=5, method='simple', t=None, sigma_sq=None, R=1, verbose=False, max_clip_iter=10)

  Smoother based on a forward and a backward kalman filter.
- def skdiscovery.utilities.kalman_smoother.FOGM (size, t, sigma_sq, R)

  Generates data from a First Order Gaussian-Markov process.

7.53 utilities/pbo_tools.py File Reference

Namespaces

- skdiscovery.utilities.pbo_tools

Functions

- def skdiscovery.utilities.pbo_tools.mogi (xdata, lat, lon, source_depth, amplitude)

  Compute the surface deformation due to changes in a mogi source.
- def skdiscovery.utilities.pbo_tools.finite_sphere (xdata, lat, lon, source_depth, amplitude, alpha_rad)

  Compute the surface deformation due to changes in a finite sphere source.
- def skdiscovery.utilities.pbo_tools.closed_pipe (xdata, lat, lon, source_depth, amplitude, pipe_delta)

  Compute the surface deformation due to changes in a closed pipe source.
- def skdiscovery.utilities.pbo_tools.constant_open_pipe (xdata, lat, lon, source_depth, amplitude, pipe_delta)

  Compute the surface deformation due to changes in a constant width open pipe source.
- def skdiscovery.utilities.pbo_tools.rising_open_pipe (xdata, lat, lon, source_depth, amplitude, pipe_delta, open↩_pipe_top)

  Compute the surface deformation due to changes in a rising width amplitude open pipe source.
- def skdiscovery.utilities.pbo_tools.sill (xdata, lat, lon, source_depth, amplitude)

  Compute the surface deformation due to changes in a sill-like source.
- def skdiscovery.utilities.pbo_tools.dirEigenvectors (coord_list, pca_comps, pdir='H')

  Takes eigenvectors (north and east) and forces them to point "outward".
- def skdiscovery.utilities.pbo_tools.datetimeToNumber (in_time)

  Converts input pandas Timestamp or pandas DatetimeIndex to unix time.

7.54 utilities/random_walks.py File Reference

Namespaces

- skdiscovery.utilities.random_walks

Functions

- def skdiscovery.utilities.random_walks.uniform_walk (pos, grid, step_size=None)

  A uniform random walk function.
- def skdiscovery.utilities.random_walks.gaussian_walk (pos, grid, step_size=None)

  A gaussian random walk function.
- def skdiscovery.utilities.random_walks.keep_in_bound (pos, grid)

  Function for truncating and bounding the random walk to within the defined grid.

7.55 utilities/spherical_voronoi.py File Reference

Namespaces

- skdiscovery.utilities.spherical_voronoi

Functions

- def skdiscovery.utilities.spherical_voronoi.sphericalToXYZ (lat, lon, radius=1)

  Convert spherical coordinates to x,y,z.
- def skdiscovery.utilities.spherical_voronoi.xyzToSpherical (x, y, z)

  Convert x,y,z to spherical coordinates.
- def skdiscovery.utilities.spherical_voronoi.find_match (region_index, region_list)

  Find neighboring regions.
- def skdiscovery.utilities.spherical_voronoi.getVoronoiCollection (data, lat_name, lon_name, bmap=None, v_↩ name=None, full_sphere=False, max_v=.3, min_v=-0.3, cmap=matplotlib.cm.get_cmap('jet'))

  Perform a Spherical Voronoi Tessellation on the input data.

7.56 utilities/ssh_reverse.py File Reference

Classes

- class skdiscovery.utilities.ssh_reverse.ReverseTunnel

  Create a reverse ssh tunnel.

Namespaces

- skdiscovery.utilities.ssh_reverse

Functions

- def skdiscovery.utilities.ssh_reverse.print_verbose (s, verbose=False)

  Print statement if verbose is True.
- def skdiscovery.utilities.ssh_reverse.handler (chan, host, port, verbose=False)

  Handler is responsible for sending and receiving data through ssh tunnel.
- def skdiscovery.utilities.ssh_reverse.reverse_forward_tunnel (server_port, remote_host, remote_port, transport, check=30, verbose=False)

  Creates a reverse ssh tunnel.

7.57 utilities/trendTools.py File Reference

Namespaces

- skdiscovery.utilities.trendTools

Functions

- def skdiscovery.utilities.trendTools.getTrend (xdata)

  The getTrend function applies the signal.detrend function.
- def skdiscovery.utilities.trendTools.sinuFits (xdata, fitN=2, rmve=1)

  The sinuFits function fits annual and semi-annual sinusoid trends.
- def skdiscovery.utilities.trendTools.interpNaN (data)

  A simple wrapper for the linear interpolation function from Numpy to fill in NaN's.
- def skdiscovery.utilities.trendTools.medianFilter (data, window, interpolate=True)

  A median filter.

7.58 visualization/multi_ca_plot.py File Reference

Namespaces

- skdiscovery.visualization.multi_ca_plot

Functions

- def skdiscovery.visualization.multiCaPlot (pipeline, mogiFlag=False, offset=.15, direction='H', pca_comp=0, scaleFactor=2.5, map_res='i')

  The multiCaPlot function generates a geographic eigenvector plot of several pipeline runs.

7.59 visualization/multi_dist.py File Reference

Namespaces

- skdiscovery.visualization.multi_dist

Functions

- def skdiscovery.visualization.calc_distance_map (pipeline, ap_name, ca_name, ca_type, plotFlag=True, hist↩ Idx=False, fontsize=10)

  Calculates distances/similarities between pipeline runs.

Variables

- skdiscovery.visualization.font

Index