Module **`fetch_data`**

Classes

`class FetchLidarData (polygon: shapely.geometry.polygon.Polygon, public_data_url: https://s3-us-west-2.amazonaws.com/usgs-lidar-public/, pipeline_json_path: ../data/pipeline.json)`

This class contains functons useful for fetching, manipulating, and visualizing LIDAR point cloud data.

This method is used to instantiate the class.

Args

`polygon`
: polygon of the the area we need to crop

`public_data_url` : `str`, optional
: [the url where the dataset can be accessed from]. Defaults to "https://s3-us-west-2.amazonaws.com/usgs-lidar-public/".

`pipeline_json_path` : `str`, optional
: [the json file describing the pipeline structure]. Defaults to "../data/pipeline.json".

Methods

`def elevation(self, x, y, z)`

`def get_bounds_and_polygon(self)`

This method returns the bounds and exterior coordinates of a polygon as strings.

Args

`polygon` : `Polygon`
: [a polygon object]

Returns

`[tuple]`

[bounds string and polygon exterior coordinates string]

```
def get_elevation(self, region: str = 'IA_FullState')
```

This method get elevation from all regions

Args

`region`

[the filename of the region where the data is extracted from]. Defaults to "IA_FullState".

Returns

`[Geopandas.GeoDataFrame]`

[a geopandas dataframe]

```
def get_raster_terrain(self, region: str = 'IA_FullState',
                       OUTPUT_FILENAME_LAZ: str = 'IA_FullState',
                       OUTPUT_FILENAME_TIF: str = 'IA_FullState',
                       pipeline_path: str = '../data/pipeline.json') -> None
```

```
def plot_terrain_3d(self, gdf: geopandas.geodataframe.GeoDataFrame,
                    fig_size: tuple = (12, 10), size: float = 0.01)
```

This method displays points in a geodataframe as a 3d scatter plot.

Args

`gdf` : `gpd.GeoDataFrame`

[a geopandas dataframe containing points in the geometry column and height in the elevation column.]

`fig_size` : `tuple`, optional

[filesze of the figure to be displayed]. Defaults to (12, 10).

`size` : `float`, optional

[size of the points to be plotted]. Defaults to 0.01.

```
def subsample(self, gdf: geopandas.geodataframe.GeoDataFrame, res: int = 6)
```

This method subsamples the points in a point cloud data using some resolution.

Args

`gdf` : `gpd.GeoDataFrame`

[a geopandas dataframe containing points in the geometry column and height in the elevation column.]

`res` : `int` , optional

[resolution]. Defaults to 3.

Returns

`[Geopandas.GeoDataFrame]`

[a geopandas dataframe]

Index

Classes

Generated by *pdoc* 0.10.0.