

***Spatial Extension for
GeoServer WPS v1.0***

Geoserver
mango



Spatial Extension for GeoServer WPS v1.0 User Manual (TBD)

This manual is supported by Ministry of Land Infrastructure Transport (R&D 14NSIP-B080144-01).

9 August 2016. 1st edition published

1 October 2017. 2nd edition published.

Author: Minpa Lee, Jihyun Kim

Reviewer: TBD

Cover Design: TBD

Editor: TBD

Copyright: Mangosystem(ltd.)

Publisher: Gia3D(ltd.)

Address: TBD

Tel: 042-330-0400

Fax: 042-330-0410

Publication Code: 2012-000016

ISBN: 978-89-969532-6-5(95000)

NOTICE: It is not for sale.

Copyright Notice

Copyright © 2016 MangoSystem Inc. All Rights Reserved.

Address: 2307-ho, Pyengchon O'bizztower, 126, Beolmal-ro, Dongan-gu, Anyang-si, Gyeonggi-do, 431-060 South Korea

Tel: 82-31-450-3411 Fax: 82-31-450-3414

E-mail: master.mangosystem@gmail.com

Homepage: <http://www.mangosystem.com>

Online Documentation: <http://gxt.mangosystem.com>

Restricted Rights Legend

TBD

Trademarks

MangoSystem Spatial Extension for GeoServer WPS ® is belong to MangoSystem Inc. TBD.

Open Source Software Notice

TDB. It respects OSSL of "GeoTools", "GeoServer".

- GeoTools: <http://www.geotools.org>
- GeoServer: <http://geoserver.org>

User Manual Information

Title: User Manual of Spatial Extension for GeoServer WPS

Issue Date: 2016-12-31

SW Version: Spatial Extension for GeoServer WPS 1.0

User Manual Version: 1.0

목 차

1	Web Processing Service	6
2	Installation.....	6
3	Quick Start.....	6
4	WPS Processes for Spatial Analysis.....	7
4.1.	Spatial Analysis Processes	7
4.2.	Vector Analysis Processes.....	11
4.2.1.	Spatial Unit Creation.....	11
4.2.2.	Calculation	35
4.2.3.	Extract	54
4.2.4.	Overlay.....	65
4.2.5.	Proximity.....	82
4.2.6.	Aggregation	107
4.2.7.	Generalization	138
4.2.8.	Editing.....	153
4.2.9.	Feature Tools	169
4.3.	Raster Analysis.....	232
4.3.1.	Descriptive	232
4.3.2.	Conversion	238
4.3.3.	Distance	254
4.3.4.	Math	258
4.3.5.	Classification	264

4.3.6.	Extraction.....	267
4.3.7.	Density.....	281
4.3.8.	Interpolation	291
4.3.9.	Surface Analysis.....	298
4.3.10.	Zonal.....	310
4.3.11.	Projection	314
4.4.	Spatial Statistics Analysis	321
4.4.1.	Descriptive	321
4.4.2.	Distributions.....	333
4.4.3.	Point Pattern Analysis.....	350
4.4.4.	Global Spatial Auto-Correlation.....	361
4.4.5.	Local Spatial Auto-Correlation.....	379
4.4.6.	Global Spatial Modeling.....	394

1 Web Processing Service -- TBD

2 Installation -- TBD

3 Quick Start -- TBD

4 WPS Processes for Spatial Analysis

4.1. Spatial Analysis Processes

A full list of spatial analysis processes provided in Oct. 2017 is as follows.

Main category	Subcategory	Process
Vector Analysis	Spatial Unit Creation	Fishnet Grids by Count
		Fishnet Grids by Size
		Hexagonal Grids
		Triangular Grids
		Circular Grids
		Thiessen Polygon
		Delaunay Triangulation polygon
		Random Points
		Random Points per Features
	Calculation	Calculate XY Coordinate
		Calculate Area
		Calculate Length
		Calculate Field
		Calculate Count
Raster Analysis	Extract	Sum Polygon Areas
		Extract Values to Points
		Select Features
		Clip with Geometry
		Clip with Features
	Overlay	Merge Features
		Union
		Intersect
		Symmetrical Difference
		Difference
Proximity	Proximity	Identity
		Update
		Buffer Features using Expression
		Single Sided Buffer
		Multiple Ring Buffer
	Geometric	Wedge Buffer
		Point in Polygon
		Point in Polygons
		Point in Polyline
		Point in Polygons and Polyline

	Near
	Nearest Neighbor Count
	Polar Grids from Geometry
	Polar Grids from Features
Aggregation	Point Statistics
	Aggregate Polygons
	Collect Events
	Spatial Join
	Attribute Join
	Buffer Point Statistics
	Sum Line Lengths
	Hexagonal Binning
	Rectangular Binning
	Circular Binning
Generalization	Spatial Clump Map
	Dissolve
	Remove Polygon Holes
	Remove Polygon Part
	Simplify
	Densify
	Eliminate
	Reverse Line Direction
	Offset Features
	Snap Points To Lines
Editing	Extend Line
	Trim Line
	Delete Duplicated Geometries
	Feature to Point
	Singlepart to Multipart
	Multipart to Singlepart
	Feature Envelope to Polygon
	Points to Line
	Ring Maps
	Wind Rose Maps
Feature Tools	Hub Lines by ID
	Hub Lines by Nearest Distance
	Feature To Line
	Feature To Polygon

Raster Analysis	Descriptive	Feature Vertices To Points Repair Geometry Create Points along Line Split Line At Point Split Line At Vertices Split Line By Distance Expression Intersection Points from Lines Feature To Octagonal Envelope Feature To Minimum Rectangle Feature To ConvexHull Feature To Minimum Bounding Circle Create Flow Map from Line Features
	Conversion	Basic Statistics Histogram Features To Coverage Points To Coverage Geometry To Coverage GridCoverage To Point GridCoverage To Polygon GridCoverage To Image
	Classification	Reclass Extract by Attributes Conditional Expression
	Extract	Extract by Geometry Extract by Extent Extract by Circle Raster Conditional Expression
	Density	Kernel Density Point Density Line Density
	Distance	Euclidean Distance
	Math	Math
	Interpolation	Inverse Distance Weighted Thin Plate Spline
	Zonal	Zonal Statistics
	Projection	Resample Redefine Projection Reproject

Spatial Statistics	Surface Analysis	Raster Profile Radial Line Of Sight Linear Line Of Sight Find Highest/Lowest Points
	Descriptive	Basic Statistics Pearson Correlation Coefficient Standardized Score of Dissimilarity Focal Location Quotients
	Distributions	Mean Center Median Center Central Feature Standard Distance Standard Deviational Mean Linear Directional Mean
	Point Pattern Analysis	Nearest Neighbor Statistic K-Nearest Neighbor Map Quadrat Method K-Means Clustering
	Global Spatial Auto-Correlation	Join Count Statistic Moran's I Geary's c Getis-Ord's General G
	Local Spatial Auto-Correlation	Lee's S Lee's L Local Moran's I Local Geary's c Local G(Gi*) Lee's Si Lee's Li
	Global Spatial Modeling	Ordinary Least Squares (OLS)

Note: It will be updated continuously.

4.2. Vector Analysis Processes

These processes are for vector data analysis and processing.

4.2.1. Spatial Unit Creation

These processes are for creating various spatial analysis units like grid, hexagon, triangular, circle, and Thissen Polygon.

4.2.1.1. Fishnet Grid by Count

Creates a Fishnet Grid based on the specified spatial extent (**extent**) and the number of columns and rows (**columns**, **rows**).

■ Syntax

```
FishnetCount (ReferencedEnvelope extent, SimpleFeatureCollection boundsSource,  
Boolean boundaryInside, Integer columns, Integer rows): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
extent	The extent of the grids.	Complex	✓
boundsSource	Bounds Source Features.	Complex	-
boundaryInside	Bounds Inside.	Literal	-
columns	Number of columns.	Literal	✓
rows	Number of Rows.	Literal	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If the **boundsSource** parameter is set, creates a fishnet grid only intersecting with the boundary of **boundsSource**.

- If the **boundsSource** parameter is set and the **boundaryInside** parameter is set to True, creates only a fishnet grid contained within the boundary of **boundsSource**.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:FishnetCount</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>extent</ows:Identifier>
      <wps:Data>
        <wps:BoundingBoxData crs="EPSG:3857" dimensions="2">
          <ows:LowerCorner>1.4111357E7 4498975.0</ows:LowerCorner>
          <ows:UpperCorner>1.4158036E7 4537337.0</ows:UpperCorner>
        </wps:BoundingBoxData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>boundsSource</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:sid"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>columns</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>25</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>rows</ows:Identifier>
      <wps:Data>
```

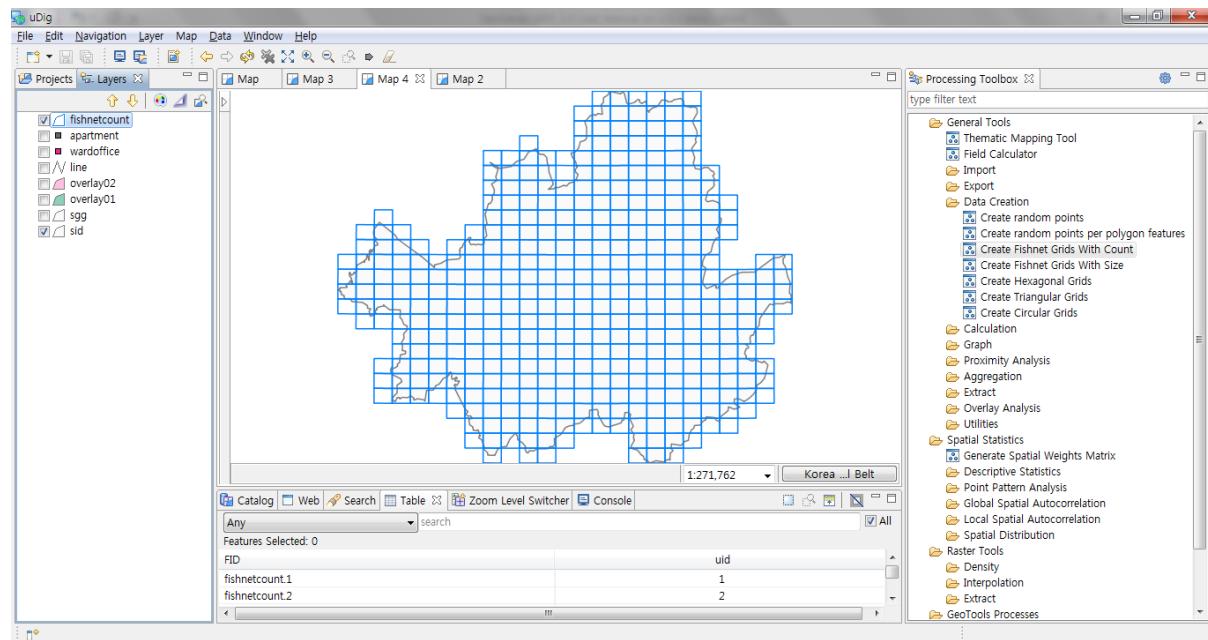
```

<wps:LiteralData>25</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows an example of creating a 25 by 25 fishnet grid based on the administrative boundary.



4.2.1.2. Fishnet Grids by Size

Creates a Fishnet Grid based on the specified spatial extent (**extent**) and the cell size (**width, height**).

■ Syntax

FishnetSize (ReferencedEnvelope extent, SimpleFeatureCollection boundsSource, Boolean boundaryInside, Double width, Double height): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
extent	The extent of the grids.	Complex	✓
boundsSource	Bounds Source Features.	Complex	-
boundaryInside	Bounds Inside.	Literal	-
width	Width of Each Cell.	Literal	✓
height	Height of Each Cell.	Literal	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If the **boundsSource** parameter is set, creates a fishnet grid only intersecting with the boundary of **boundsSource**.
- If the **boundsSource** parameter is set and the **boundaryInside** parameter is set to True, creates only a fishnet grid contained within the boundary of **boundsSource**.

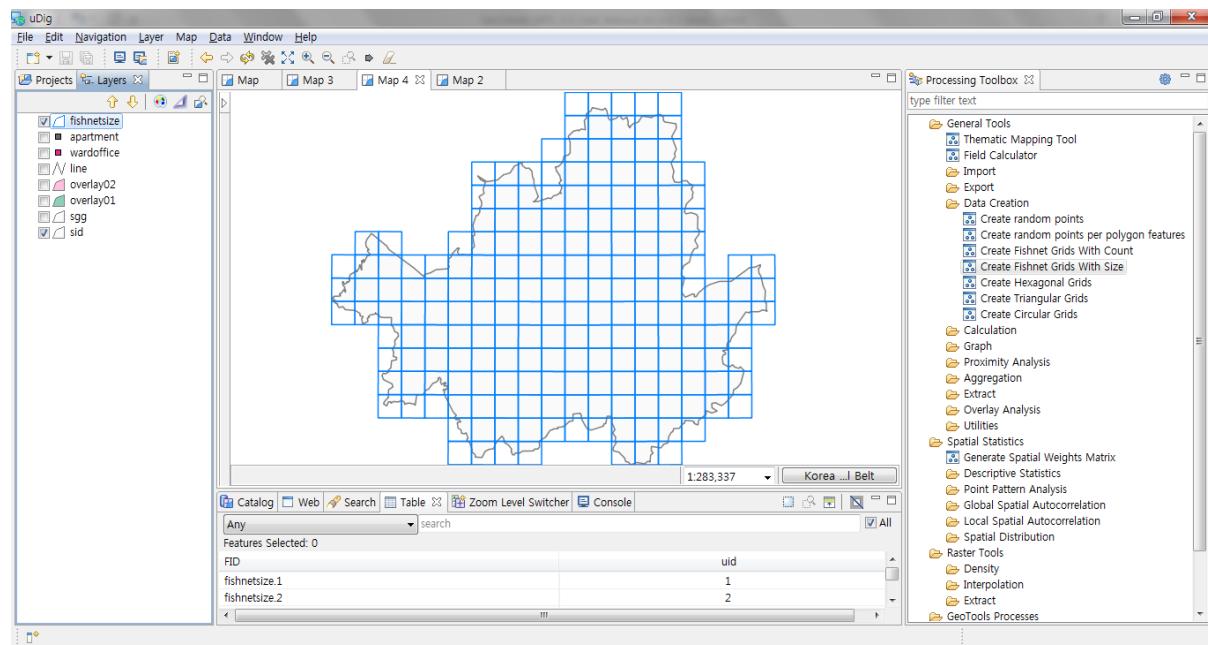
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
```

```
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:FishnetSize</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>extent</ows:Identifier>
<wps:Data>
<wps:BoundingBoxData crs="EPSG:3857" dimensions="2">
<ows:LowerCorner>1.4111357E7 4498975.0</ows:LowerCorner>
<ows:UpperCorner>1.4158036E7 4537337.0</ows:UpperCorner>
</wps:BoundingBoxData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>boundsSource</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:sid"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>width</ows:Identifier>
<wps:Data>
<wps:LiteralData>2500</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>height</ows:Identifier>
<wps:Data>
<wps:LiteralData>2500</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows an example of creating a fishnet grid of 2500×2500 meters based on the administrative boundary (Si-Do).



4.2.1.3. Hexagonal Grids

Creates a Hexagonal Grid based on the specified spatial extent (**extent**) and the cell size (**sideLen**).

■ Syntax

Hexagon (ReferencedEnvelope extent, SimpleFeatureCollection boundsSource, Double sideLen, HexagonOrientation orientation): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
extent	The extent of the grids.	Complex	✓
boundsSource	Bounds Source Features.	Complex	-
sideLen	Side length, radius.	Literal	✓
orientation	Hexagon Orientation: FLAT (default), ANGLED.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If the **boundsSource** parameter is set, creates hexagons only intersecting with the boundary of the **boundsSource**.
- The **sideLen** parameter defines the distance from the hexagon's center to its edge.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
```

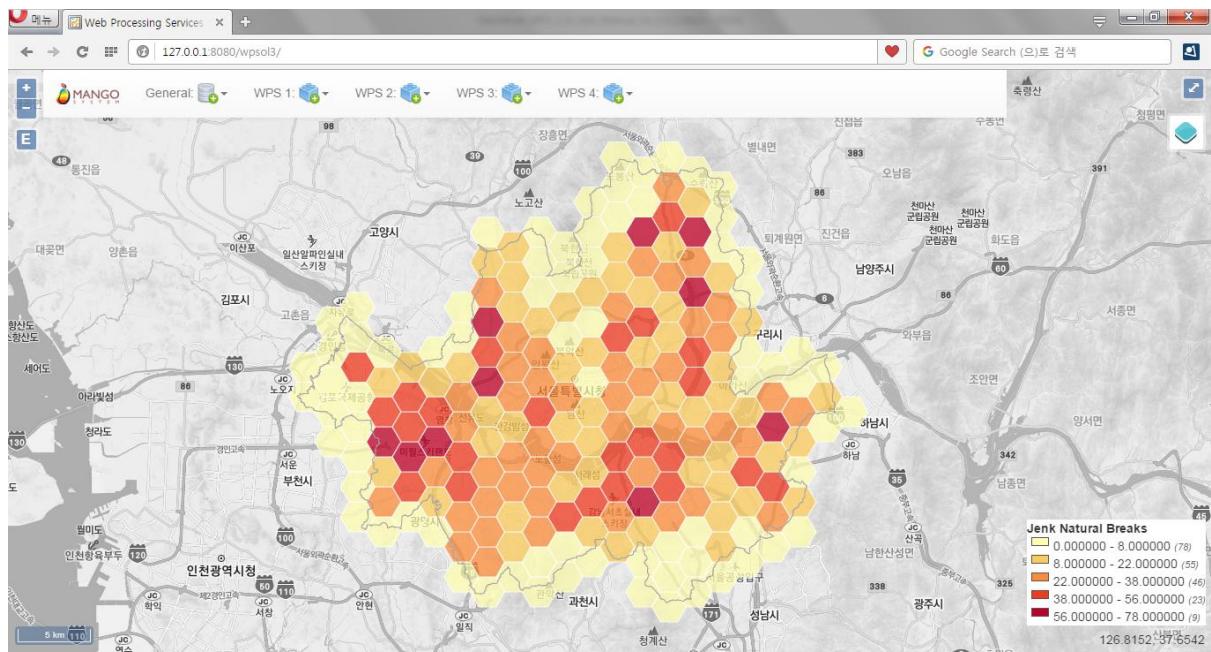
```

<ows:Identifier>statistics:Hexagon</ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>extent</ows:Identifier>
    <wps>Data>
      <wps:BoundingBoxData crs="EPSG:3857" dimensions="2">
        <ows:LowerCorner>1.4111357E7 4498975.0</ows:LowerCorner>
        <ows:UpperCorner>1.4158036E7 4537337.0</ows:UpperCorner>
      </wps:BoundingBoxData>
    </wps>Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>boundsSource</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
          <wfs:Query typeName="foss:sid"/>
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>sideLen</ows:Identifier>
    <wps>Data>
      <wps:LiteralData>1500</wps:LiteralData>
    </wps>Data>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows an example of the choropleth map of the apartment density calculated on the hexagonal grid with the hexagon size of 1500 meters based on the administrative boundary.



4.2.1.4. Triangular Grids

Create a Triangular Grid based on the specified extent (**extent**) and the cell size (**size**).

■ Syntax

TriangularGrid (ReferencedEnvelope extent, SimpleFeatureCollection boundsSource, Double size, HexagonOrientation orientation): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
extent	The extent of the grids.	Complex	✓
boundsSource	Bounds Source Features.	Complex	-
size	Grid Size.	Literal	✓
orientation	Orientation: FLAT (default), ANGLED.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If the **boundsSource** parameter is set, creates triangular only intersecting with the boundary of the **boundsSource**.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
    <ows:Identifier>statistics:TriangularGrid</ows:Identifier>
    <wps:DataInputs>
        <wps:Input>
            <ows:Identifier>extent</ows:Identifier>
```

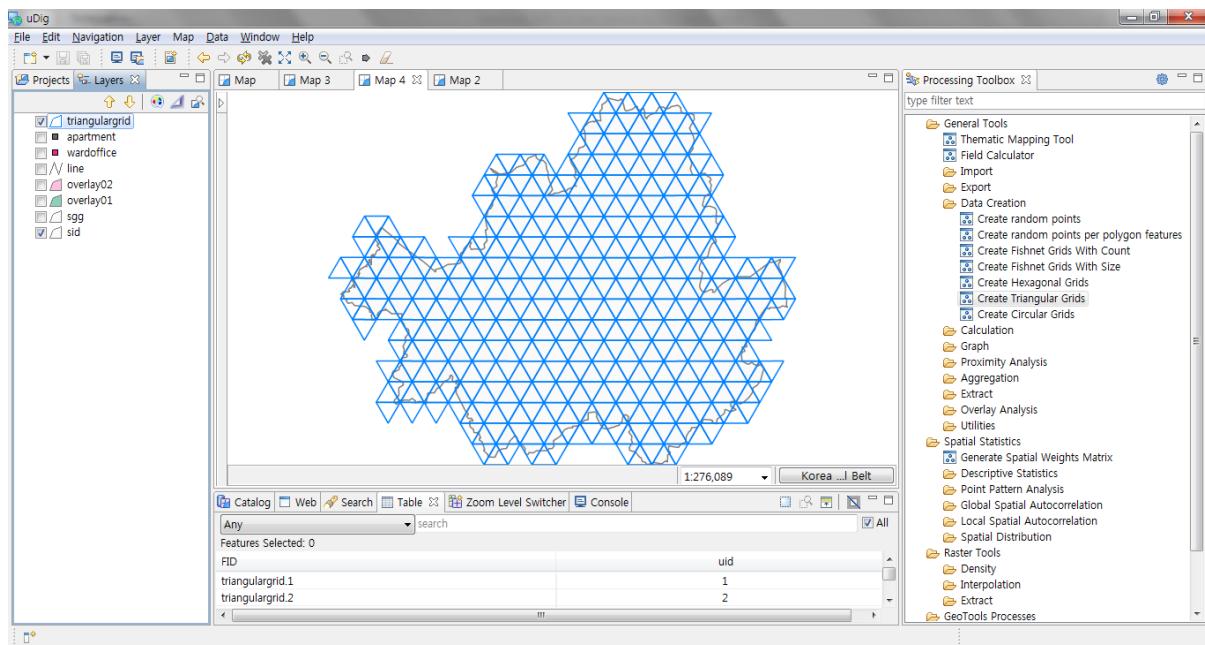
```

<wps:Data>
  <wps:BoundingBoxData crs="EPSG:3857" dimensions="2">
    <ows:LowerCorner>1.4111357E7 4498975.0</ows:LowerCorner>
    <ows:UpperCorner>1.4158036E7 4537337.0</ows:UpperCorner>
  </wps:BoundingBoxData>
</wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>boundsSource</ows:Identifier>
  <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
    <wps:Body>
      <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
      xmlns:foss="http://www.opengeospatial.net/foss">
        <wfs:Query typeName="foss:sid"/>
      </wfs:GetFeature>
    </wps:Body>
  </wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>size</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>2500</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows an example of creating a Triangular grid with a side length of 2500 meter based on administrative boundary (Si-Do).



4.2.1.5. Circular Grids

- Creates a Circular grid based on the specified extent (**extent**) and the cell size (**radius**).

■ Syntax

CircularGrid (ReferencedEnvelope extent, SimpleFeatureCollection boundsSource, Double radius, CircularType circularType): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
extent	The extent of the grids.	Complex	✓
boundsSource	Bounds Source Features.	Complex	-
radius	Radius of the circle.	Literal	✓
circularType	Circular Type: Grid (default), Hex.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If the **boundsSource** parameter is set, creates circular intersecting with the boundary of the **boundsSource**.
- If the **circularType** is Grid, the rules of creating circular in grid follows the rules of creating hexagons when the **circularType** is Hex.

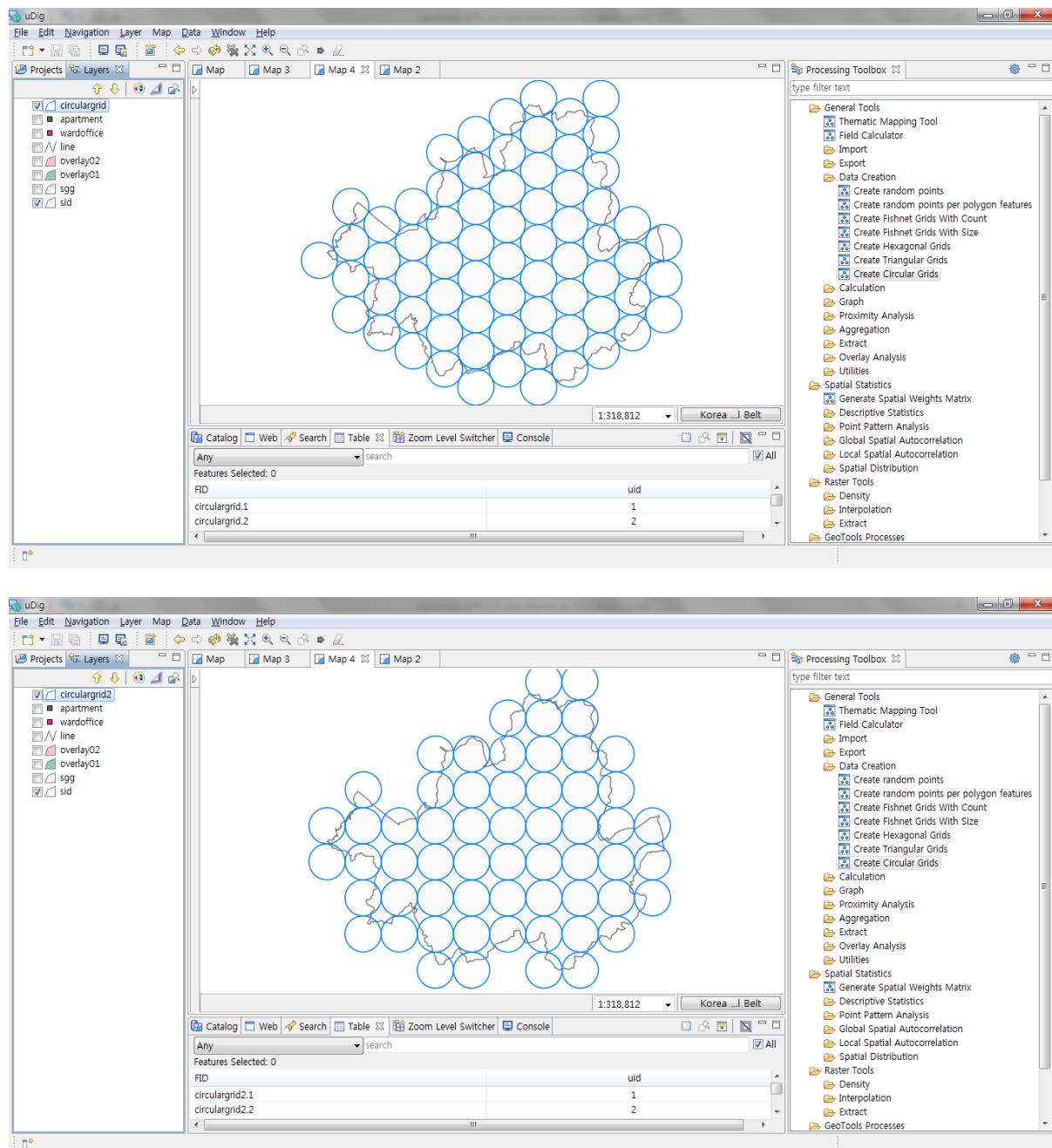
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
```

```
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:CircularGrid</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>extent</ows:Identifier>
<wps:Data>
<wps:BoundingBoxData crs="EPSG:3857" dimensions="2">
<ows:LowerCorner>1.4111357E7 4498975.0</ows:LowerCorner>
<ows:UpperCorner>1.4158036E7 4537337.0</ows:UpperCorner>
</wps:BoundingBoxData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>boundsSource</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:sid"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>radius</ows:Identifier>
<wps:Data>
<wps:LiteralData>2500</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>circularType</ows:Identifier>
<wps:Data>
<wps:LiteralData>Hex</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows an example of creating a Circular grid with a radius of 2500 meter in both Hexagon and Grid options based on the administrative boundary (Si-Do).



4.2.1.6. Thiessen Polygon

Creates Thiessen polygon using input feature layers (**inputFeatures**).

■ Syntax

ThiessenPolygon (SimpleFeatureCollection **inputFeatures**, ThiessenAttributeMode attributes, Geometry **clipArea**): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The point input features from which thiessen polygons will be generated.	Complex	✓
attributes	Attribute mode: ONLY_FID (default), ALL (retain input feature's attribute).	Literal	-
clipArea	The clip area polygon geometry.	Complex	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Though the **inputFeatures** can be points, lines and polygons, Thiessen polygon will be output after creating gravity center.
- If the **attributes** parameter is All, maintains all attribute value of **inputFeatures**.
- If the **clipArea** parameter is given, returns clipped polygons in the relevant areas.

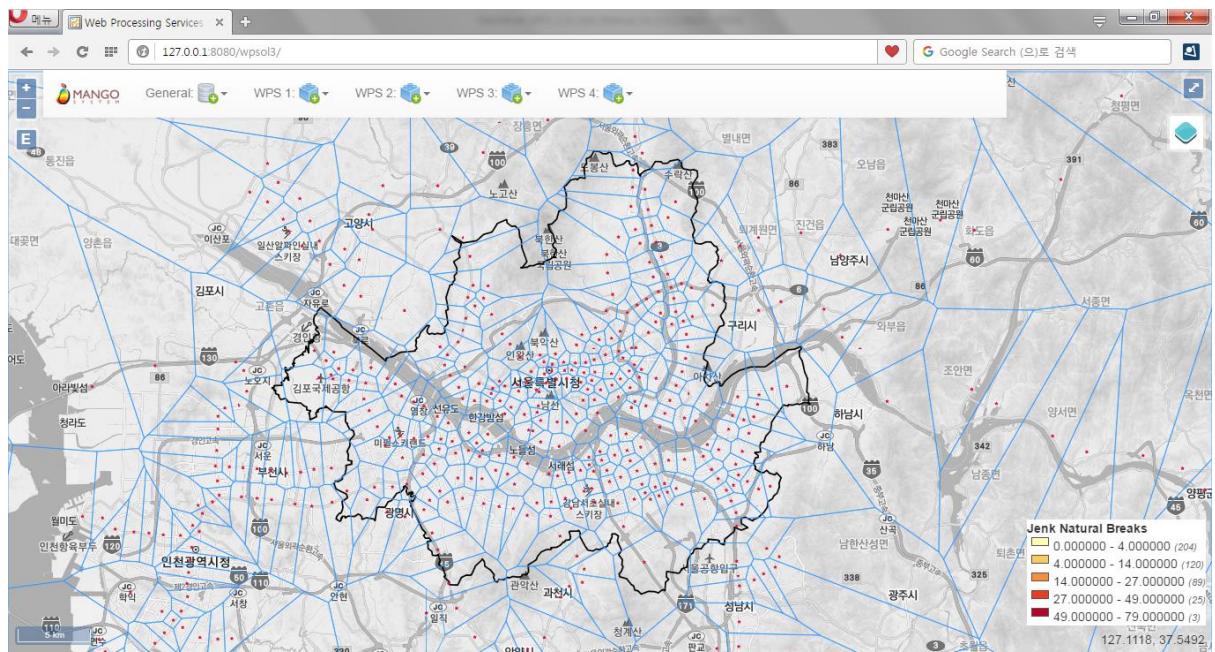
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0"
```

```
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:ThiessenPolygon</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:stores"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>attributes</ows:Identifier>
<wps:Data>
<wps:LiteralData>ALL</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows an example of creating Thiessen Polygon within current map boundary based on national train/subway stations.



4.2.1.7. Delaunay Triangulation Polygons

Creates Delaunay Triangulation polygons using input point feature layers (**inputFeatures**).

■ Syntax

DelaunayTriangulation (SimpleFeatureCollection inputFeatures, Geometry clipArea):

SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The point input features from which delaunay triangulations will be generated.	Complex	✓
clipArea	Clip area polygon.	Complex	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If the **clipArea** parameter is given, returns clipped polygons in relevant areas.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
    <ows:Identifier>statistics:DelaunayTriangulation </ows:Identifier>
    <wps:DataInputs>
        <wps:Input>
            <ows:Identifier>inputFeatures</ows:Identifier>
            <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
                <wps:Body>
                    <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
```

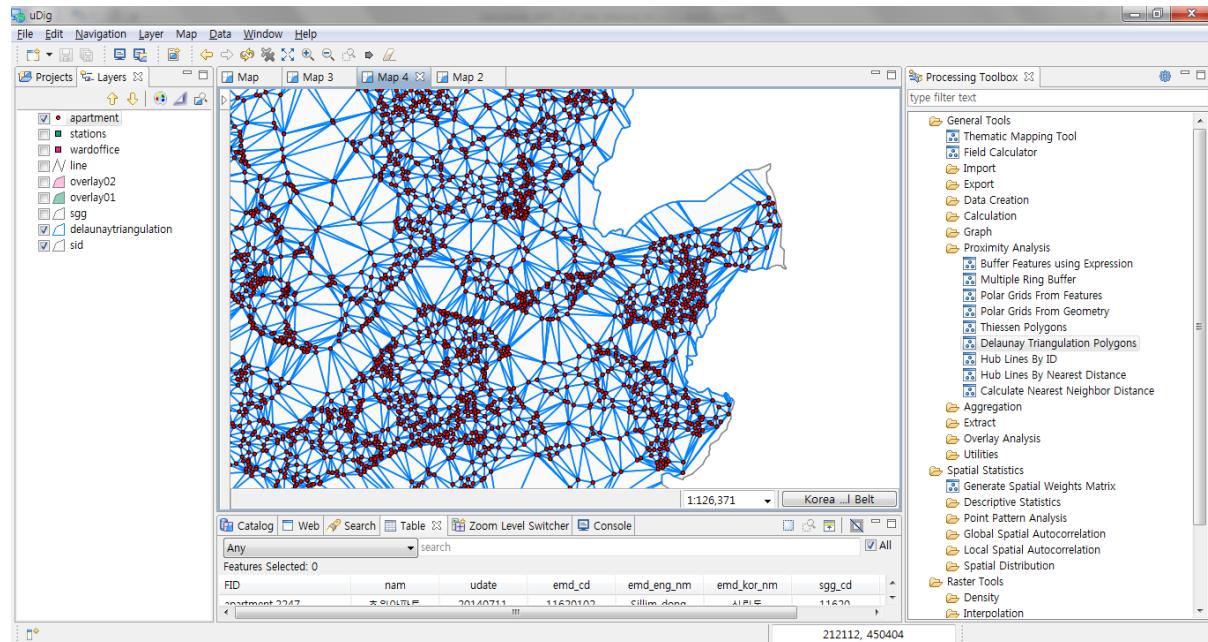
```

xmlns:foss="http://www.opengeospatial.net/foss"
    <wfs:Query typeName="foss:apartment"/>
    </wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
        <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows an example of creating Delaunay Triangulation Polygon using points within some specified boundary such as the administrative boundary (Si-Do).



4.2.1.8. Random Points

Creates random points according to settings using boundary of polygon features (**polygonFeatures**) or the specified Extent (**extent**).

■ Syntax

RandomPoints (ReferenceEnvelope extent, SimpleFeatureCollection polygonFeatures, Integer pointCount): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
extent	Random points will be generated inside the extent.	Complex	-
polygonFeatures	The features which contains the features into which the random points will be placed.	Complex	-
pointCount	The number of points to be randomly generated.	Literal	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- BoundingBox or polygon layers can be used as basic layers.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RandomPoints</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>polygonFeatures</ows:Identifier>
```

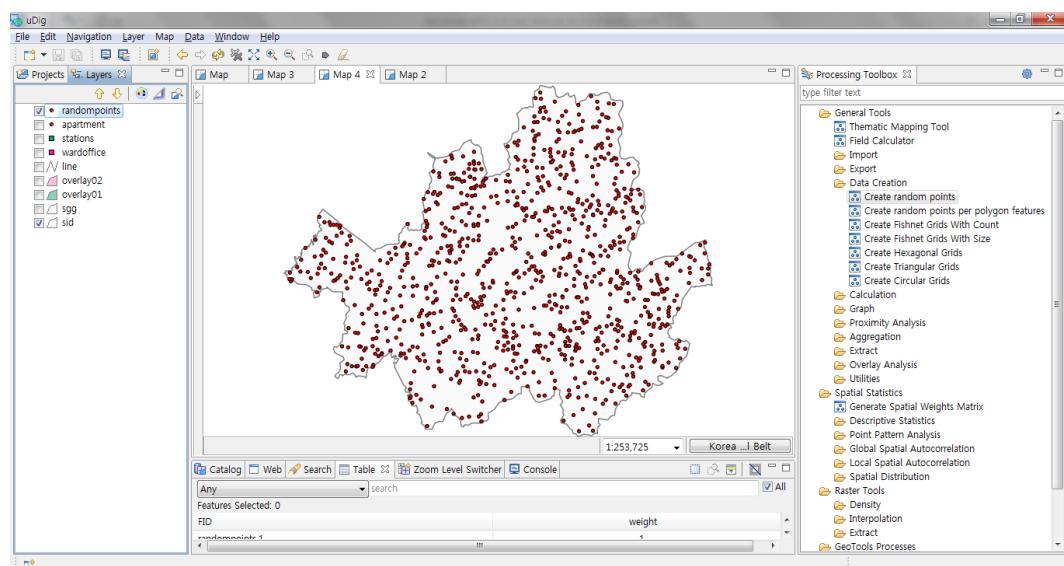
```

<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
  <wps:Body>
    <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
      xmlns:foss="http://www.opengeospatial.net/foss">
      <wfs:Query typeName="foss:sid"/>
    </wfs:GetFeature>
  </wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>pointCount</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>1000</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows an example of creating 1000 random points based on the administrative boundary (Si-Do).



4.2.1.9. Random Points per Features

Creates random point of every feature using polygon feature layers (**polygonFeatures**) and expression formula (**expression**).

■ **Syntax**

RandomPointsPerFeatures (SimpleFeatureCollection polygonFeatures, Expression expression): SimpleFeatureCollection

■ **Parameters**

■ **Data Inputs**

Identifier	Description	Type	Required
polygonFeatures	The features which contains the features into which the random points will be placed.	Complex	✓
expression	Field or Expression representing Number of Points.	Literal	-

■ **Process Outputs**

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ **Constraints**

- **Expression** parameter can be input in numbers, fields or formulas (arithmetic operation association between spatial and attribute fields)

■ **Request Examples**

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RandomPointsPerFeatures</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
```

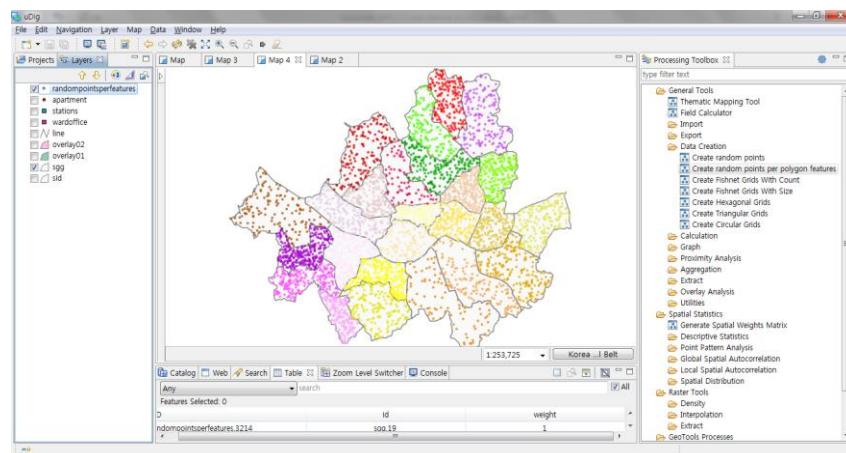
```

<ows:Identifier>polygonFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
    <wps:Body>
        <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
            xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:ssg"/>
        </wfs:GetFeature>
    </wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
    <ows:Identifier>expression</ows:Identifier>
    <wps>Data>
        <wps:LiteralData>[pop_den] / 100</wps:LiteralData>
    </wps>Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
        <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows an example of creating random points using attribute information of population density in the administrative boundary (Si-Gun-Gu).



4.2.2. Calculation

These processes are for calculating new values using Geometry or attribute value of fields.

4.2.2.1. Calculate XY Coordinate

Adds X (**xField**) and Y (**yField**) field and calculates values of coordinate systems set by users for each feature (**inputFeatures**).

■ Syntax

CalculateXYCoordinate (SimpleFeatureCollection inputFeatures, String xField, String yField, Boolean inside, CoordinateReferenceSystem targetCRS): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input features to be calculated.	Complex	✓
xField	X coordinate field that will be calculated.	Literal	-
yField	Y coordinate field that will be calculated.	Literal	-
inside	Centroid(False, Default), Inside(True)	Literal	-
targetCRS	The target coordinate reference system to use for reprojection. Ex)epsg:4326	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- The **inputFeatures** can be points, lines, and polygons.
- If the **inputFeatures** is polygon and the **inside** is set to True, center points must be included inside the polygon.
- If the **targetCSR** is Null, returns to coordinate value of original data, otherwise to coordinate conversion value.

■ Request Examples

This is the result of returning to GML3 by loading foss:stores layer of GeoServer through WFS and calculating values of EPSG:4326(WGS84 latitude and longitude) coordinate system in xcoord, ycoord field.

If the **inputFeatures** is polygon due to the **inside** value is set to True, adjusted X, Y value of the center point will be returned in order to place within the polygon.

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:CalculateXYCoordinate</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
            xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:stores" />
            </wfs:GetFeature>
          </wps:Body>
        </wps:Reference>
      </wps:Input>
      <wps:Input>
        <ows:Identifier>xField</ows:Identifier>
        <wps:Data>
          <wps:LiteralData>xcoord</wps:LiteralData>
        </wps:Data>
      </wps:Input>
      <wps:Input>
        <ows:Identifier>yField</ows:Identifier>
        <wps:Data>
          <wps:LiteralData>ycoord</wps:LiteralData>
        </wps:Data>
      </wps:Input>
      <wps:Input>
```

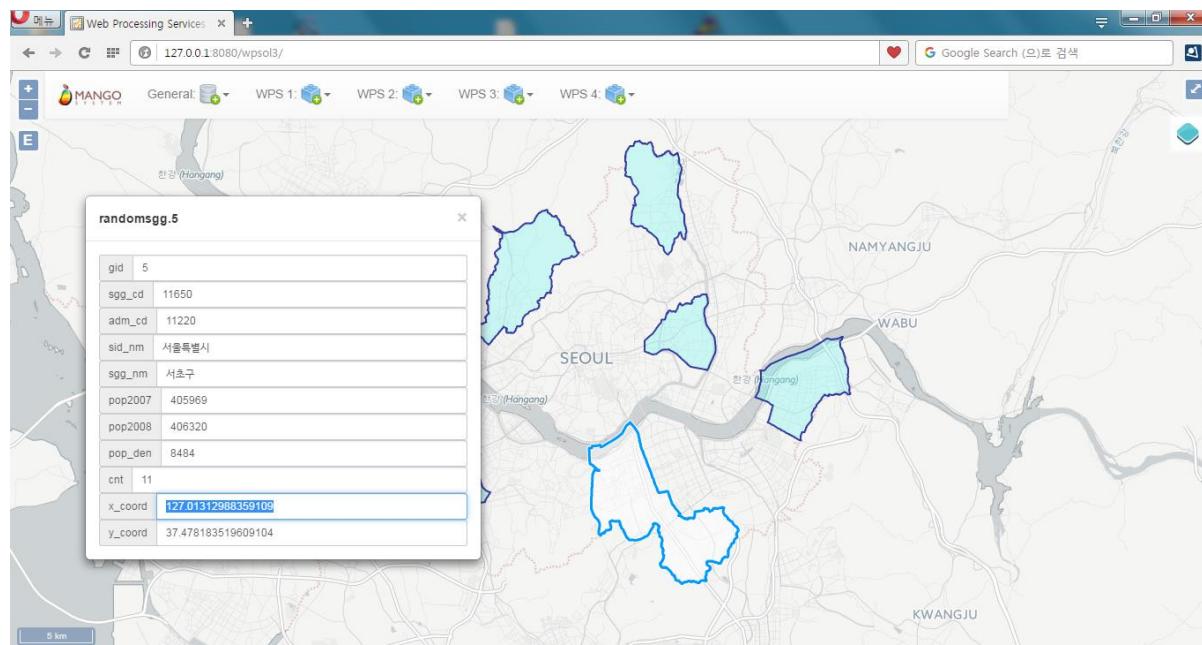
```

<ows:Identifier>inside</ows:Identifier>
<wps:Data>
  <wps:LiteralData>True</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>targetCRS</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>EPSG:4326</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows an example of calculating xcoord, ycoord values to EPSG:4326(WGS84 latitude and longitude) coordinate value based on the center value of the polygon layer.



4.2.2.2. Calculate Area

Calculates area (**areaField**) and perimeter (**perimeterField**) of polygon feature layers (**inputFeatures**).

■ Syntax

```
CalculateArea (SimpleFeatureCollection inputFeatures, String areaField, String perimeterField): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input polygon features to be calculated.	Complex	✓
areaField	The area field that will be calculated. geom_area is a default.	Literal	-
perimeterField	The perimeter field that will be calculated.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **InputFeatures** must be polygons.
- Unit of area and perimeter calculation is consistent with that of coordinate system of **inputFeatures**.

■ Request Examples

This is the result of returning to GML format after loading foss:randomsgg layer of WFS and calculating area and perimeter.

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
```

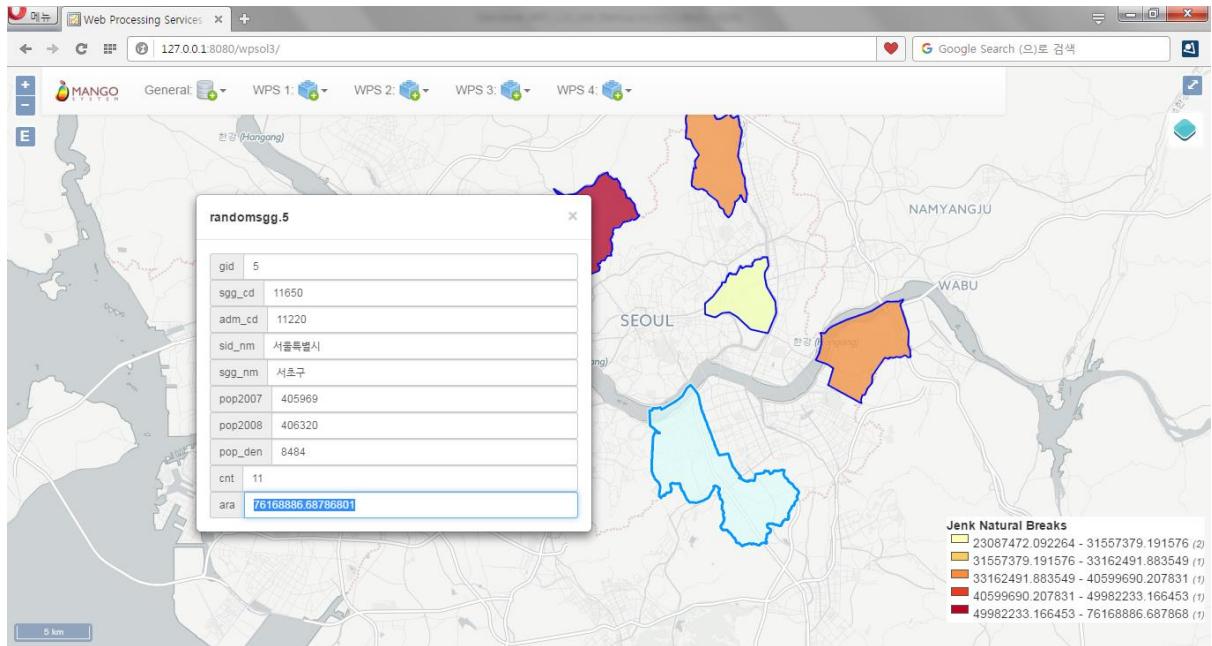
```

xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:CalculateArea</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:randomsgg" />
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>areaField</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>area</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>perimeterField</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>perimeter</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows result of calculating polygon areas in **areaFeild**.



4.2.2.3. Calculate Length

Adds fields to polygon or line feature layers and calculates the length (**lengthField**).

■ Syntax

CalculateLength (SimpleFeatureCollection inputFeatures, String lengthField):

SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input line or polygon features to be calculated.	Complex	✓
lengthField	The length field that will be calculated. geom_len is a default.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Parameter of **inputFeatures** must be polygon or line types.
- Unit of perimeter calculation value follows that of the coordinate system.

■ Request Examples

This is the result of returning in GML formate after loading foss:line layer of WFS and calculating len (length of Geometry).

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:CalculateLength</ows:Identifier>
```

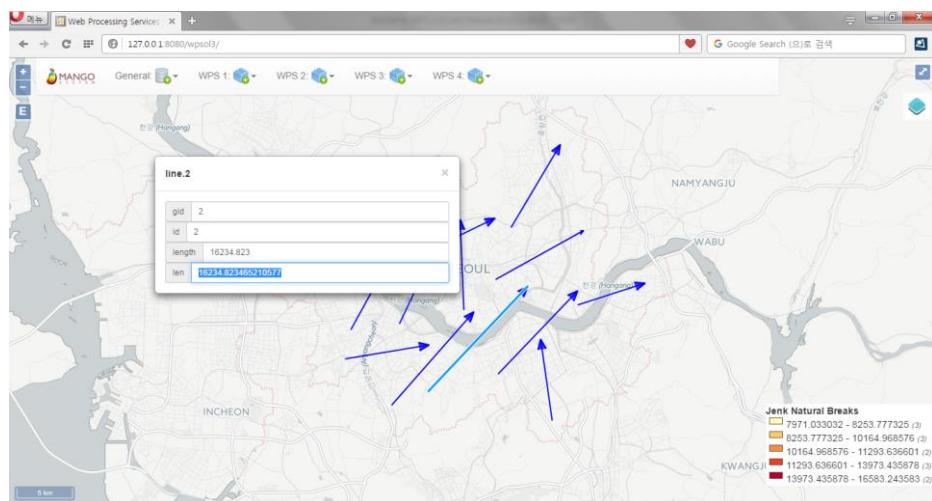
```

<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
          xmlns:foss="http://www.opengeospatial.net/foss">
          <wfs:Query typeName="foss:line" />
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>lengthField</ows:Identifier>
    <wps:Data>
      <wps:LiteralData>len</wps:LiteralData>
    </wps:Data>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of calculating line length in len field.



4.2.2.4. Calculate Field

Calculates new field values or changes Geometry Type by using user-set expression formula (**expression**).

■ Syntax

CalculateField (SimpleFeatureCollection inputFeatures, Expression expression, String fieldName): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The Input features to be calculated.	Complex	✓
expression	The simple calculation expression used to create a value that will populate the selected rows. ex) [population] / ([geom_area] / 1000000)	Literal	-
fieldName	The field that will be updated with the new calculation. Evaluated is a default.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If returned value through **expression** is Geometry, **fieldName** is ignored and Geometry value of returned data is applied.

■ Request Examples

This is the result of returning in GLM formate after calculating field value of pop_den by loading foss:randomsgg layer of WFS and calculating population density using pop2008 (population) and area (area([geom])).

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
```

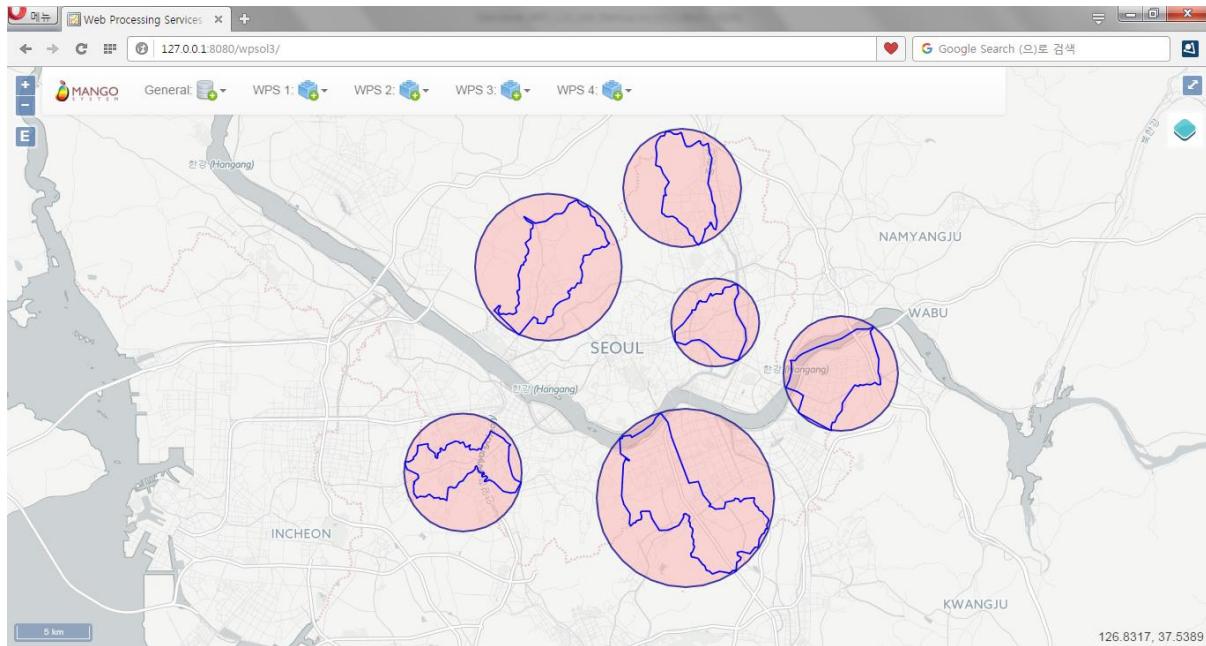
```

xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:CalculateField</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3">
            xmlns:foss="http://www.opengeospatial.net/foss">
              <wfs:Query typeName="foss:randomsgg" />
            </wfs:GetFeature>
          </wps:Body>
        </wps:Reference>
      </wps:Input>
      <wps:Input>
        <ows:Identifier>expression</ows:Identifier>
        <wps>Data>
          <wps:LiteralData>round([pop2008] / (area( [geom] ) / 1000000))</wps:LiteralData>
        </wps>Data>
      </wps:Input>
      <wps:Input>
        <ows:Identifier>fieldName</ows:Identifier>
        <wps>Data>
          <wps:LiteralData>pop_den</wps:LiteralData>
        </wps>Data>
      </wps:Input>
    </wps:DataInputs>
    <wps:ResponseForm>
      <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
        <ows:Identifier>result</ows:Identifier>
      </wps:RawDataOutput>
    </wps:ResponseForm>
  </wps:Execute>

```

■ Response

The following figure shows an example of processing mincircle ([geom]). Not only such attribute value calculation using field arithmetic operation is possible, process of Geometry change is also possible.



■ Advanced Expression

The followings are the diverse applications of **expression** parameter. **[geom]** is Geometry field name of PostGIS, Shapefile etc.

Identifier	Expression	Return Value
General Formula	round([pop2008] / (area([geom]) / 1000000))	Numeric
Area	area([geom])	Numeric
Perimeter/Length	geomLength([geom])	Numeric
X of center point	getX(centroid([geom]))	Numeric
Polygons to lines	boundary([geom])	Geometry
Buffer	bufferWithSegments([geom], 250, 16)	Polygon
Center points of gravity	centroid([geom])	Point
Points contained within polygons	interiorPoint([geom])	Point
Start points of polygons or lines	startPoint([geom])	Point
End points of polygons or lines	endPoint([geom])	Point
Convex Hull Convex Hull of polygons, lines and multipoint	convexHull([geom])	Polygon
Minimum circle around polygons, lines and multipoint	mincircle([geom])	Polygon
Minimum radius line of the area containing polygons, lines and multipoint	minimumdiameter([geom])	Line
Minimum radius region containing	minrectangle([geom])	Polygon

polygons, lines and multipoint

Least octagonal area containing octagonalenvelope([geom]) Polygon
polygons, lines and multipoint

Mover by x offset, y offset offset([geom], 4000, 3000) Geometry

4.2.2.5. Calculate Count

Calculates feature numbers using feature layers (**inputFeatures**) and filters (**filter**).

■ Syntax

CountFeatures (SimpleFeatureCollection inputFeatures, Filter filter): Integer

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features to be calculated.	Complex	✓
filter	The filter to apply	Complex	-

■ Process Outputs

Identifier	Description	Type	Required
result	The number of features.	Literal	✓

■ Constraints

- Calculates feature numbers using layers and filters and then returns them.

■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:CountFeatures</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
            xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:emd" />
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm mimeType="text/plain" />
</wps:Execute>
```

```

</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>filter</ows:Identifier>
<wps:Data>
<wps:ComplexData mimeType="text/plain; subtype=cql"><![CDATA[sgg_nm =
'강남구']]></wps:ComplexData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput>
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

Feature numbers are returned in Integer values.

4.2.2.6. Sum Polygon Areas

Calculates the sum of area of all polygon features using polygon feature layers (**inputFeatures**) and filters (**filter**).

■ Syntax

SumAreas (SimpleFeatureCollection inputFeatures, Filter filter): Double

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Polygon features.	Complex	✓
filter	The filter to apply	Complex	-

■ Process Outputs

Identifier	Description	Type	Required
result	The area of features.	Literal	✓

■ Constraints

- Returns to the sum of Geometry area of features using polygon layers and filters.

■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:SumAreas</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
            xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:emd" />
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>filter</ows:Identifier>
      <wps:Data>
        <wps:ComplexData mimeType="text/plain; subtype=cql"><![CDATA[sgg_nm =
'강남구']]></wps:ComplexData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput>
```

```
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

Areas of polygon features are returned in Double values.

4.2.2.7. Extract Values to Points

Calculates cell's digital values (**valueField**) of raster layers in attribute fields of points by overlapping point features (**pointFeatures**) and raster layers.

■ **Syntax**

```
ExtractValuesToPoints (SimpleFeatureCollection pointFeatures, String valueField,  
GridCoverage2D valueCoverage, ExtractionType valueType): SimpleFeatureCollection
```

■ **Parameters**

■ **Data Inputs**

Identifier	Description	Type	Required
pointFeatures	The input point features defining the locations.	Complex	✓
valueField	The value field to be calculated.	Literal	-
valueCoverage	The gridcoverage whose values will be extracted.	Complex	✓
valueType	Extraction type: Default, SlopeAsDegree, SlopeAsPercentrise, Aspect.	Literal	-

■ **Process Outputs**

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ **Constraints**

- If the **valueField** is set to Null, uses the name of evaluated field
- If **valueType** parameter is set to Null, returns in original cell value of **GridCoverage**.
- If **valueCoverage** is DEM, **valueType** can use SlopeAsDegree, SlopeAsPercentrise, Aspect options.

■ **Request Examples**

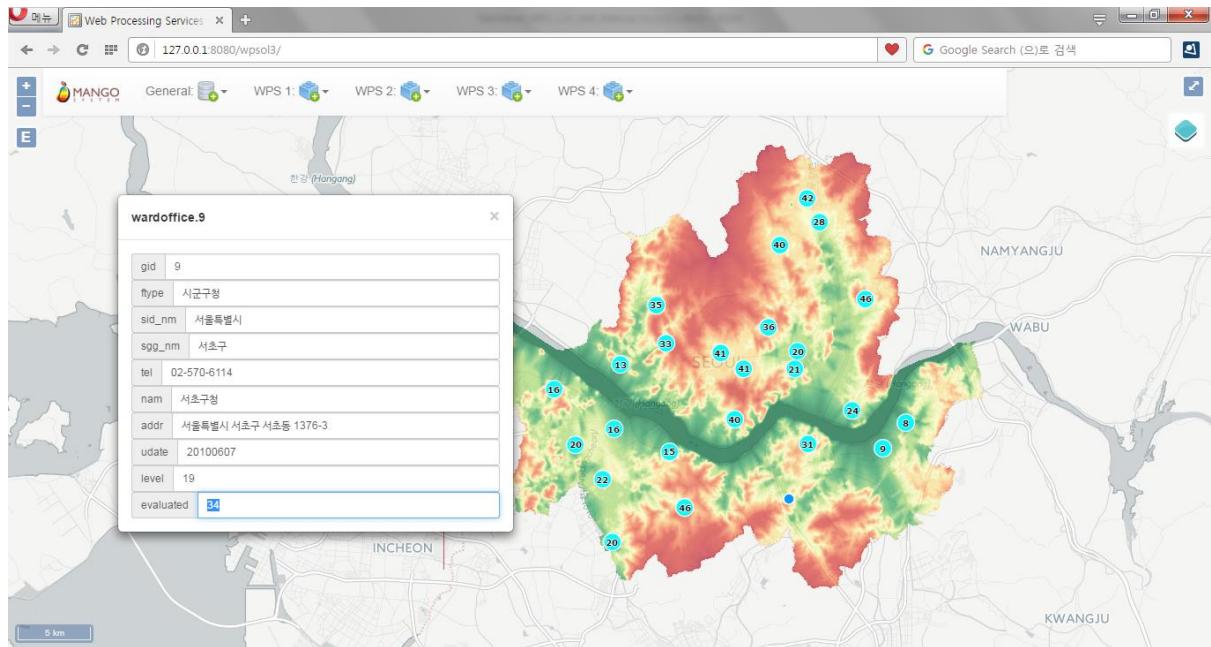
```
<?xml version="1.0" encoding="utf-8"?>  
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"  
xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
```

```
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:ExtractValuesToPoints</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>pointFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:wardoffice" />
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>valueField</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>evaluated</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>valueCoverage</ows:Identifier>
      <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
        <wps:Body>
          <wcs:GetCoverage service="WCS" version="1.1.1">
            <ows:Identifier>foss:seoul_dem30</ows:Identifier>
            <wcs:DomainSubset>
              <gml:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#5181">
                <ows:LowerCorner>179171.39881047895 436569.3290600816</ows:LowerCorner>
                <ows:UpperCorner>216221.0981287582 466869.08315843146</ows:UpperCorner>
              </gml:BoundingBox>
            </wcs:DomainSubset>
            <wcs:Output format="image/tiff" />
          </wcs:GetCoverage>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
      <ows:Identifier>result</ows:Identifier>
```

```
</wps:RawDataOutput>  
</wps:ResponseForm>  
</wps:Execute>
```

■ Response

The following figure shows the result off calculating DEM elevation values of point data of municipal offices.



4.2.3. Extract

These processes are for selecting or cutting features using filter or geometry.

4.2.3.1. Select Features (Query & Retype)

Selecting features by setting catalogs of fields and spatial or attributable filters (**filter**).

■ Syntax

SelectFeatures (SimpleFeatureCollection inputFeatures, Filter filter, String attributes):

SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features to be queried.	Complex	✓
filter	The filter to apply.	Complex	-
attributes	The comma separated fields list to include in output.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Only feature meeting specified criteria can be selected using filter.
- You can select some attribute information or change the field order by setting a comma-separated field.

■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
```

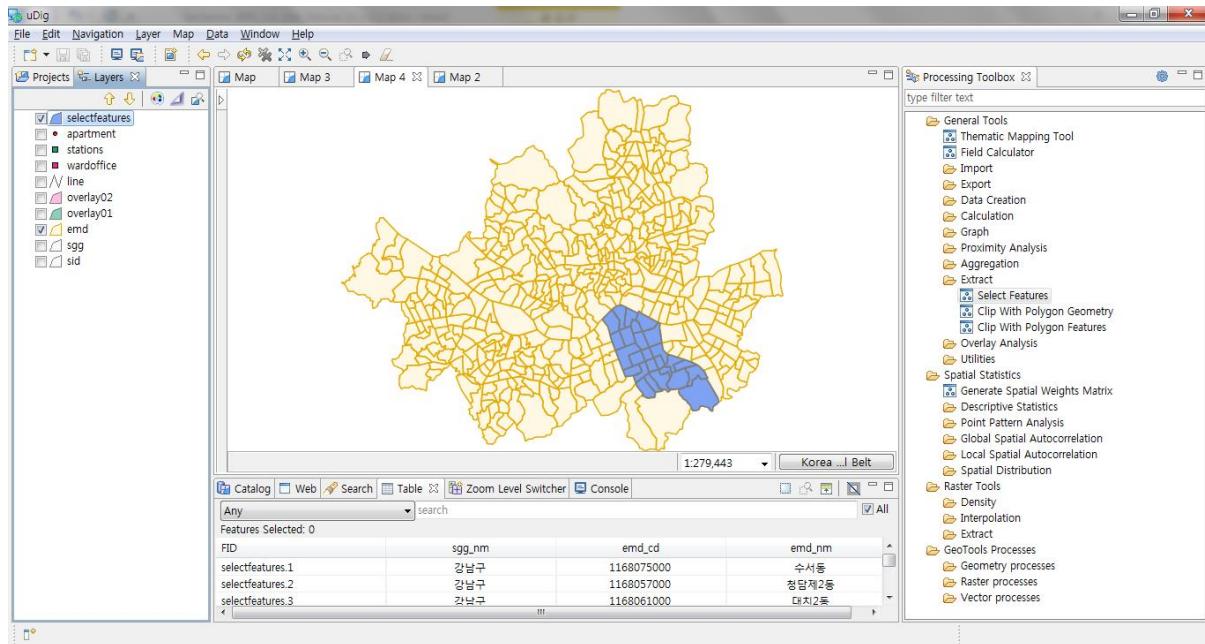
```

xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:SelectFeatures</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:emd" />
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>filter</ows:Identifier>
<wps:Data>
<wps:ComplexData mimeType="text/plain; subtype=cql"><![CDATA[sgg_nm =
'강남구']]></wps:ComplexData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>attributes</ows:Identifier>
<wps:Data>
<wps:LiteralData>gid, geom, sgg_nm, emd_cd, emd_nm</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of selecting Gangnam-Gu in Eup-Myeon-Dong administrative boundary data and extracting geom, sgg_nm, emd_cd, emd_nm fields.



4.2.3.2. Clip with Geometry

Sets geometry (**clipGeometry**) of polygons to be cut and then cuts the feature layers.

■ Syntax

ClipWithGeometry (SimpleFeatureCollection inputFeatures, Geometry clipGeometry):

SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features to be clipped.	Complex	✓
clipGeometry	The polygon geometry used to clip the input features.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **ClipGeometry** must be Polygon or Multipolygon.

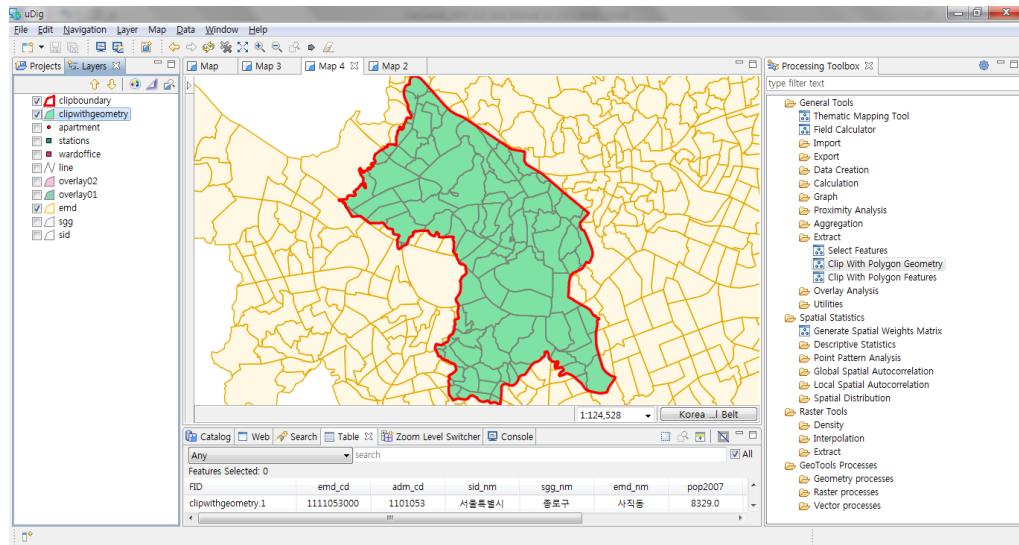
■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:ClipWithGeometry</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
```

```
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
  <wps:Body>
    <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
      xmlns:foss="http://www.opengeospatial.net/foss">
      <wfs:Query typeName="foss:emd" />
    </wfs:GetFeature>
  </wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>clipGeometry</ows:Identifier>
  <wps:Data>
    <wps:ComplexData mimeType="application/wkt"><![CDATA[POLYGON ((14123807 4505796,
      14139276 4505796, 14139276 4524167.5, 14123807 4524167.5, 14123807
      4505796))]]></wps:ComplexData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of clipping the specified region (Geometry) in the administrative boundary (Eup-Myeon-Dong).



4.2.3.3. Clip with Features

Sets polygon feature layers to be clipped (**inputFeature**) and then clips the feature layers.

■ Syntax

ClipWithFeatures (SimpleFeatureCollection inputFeatures, SimpleFeatureCollection clipFeatures): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features to be clipped.	Complex	✓
clipFeatures	The features used to clip the input features.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **ClipGeometry** must be Polygon or Multipolygon.

■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:ClipWithFeatures</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3">
```

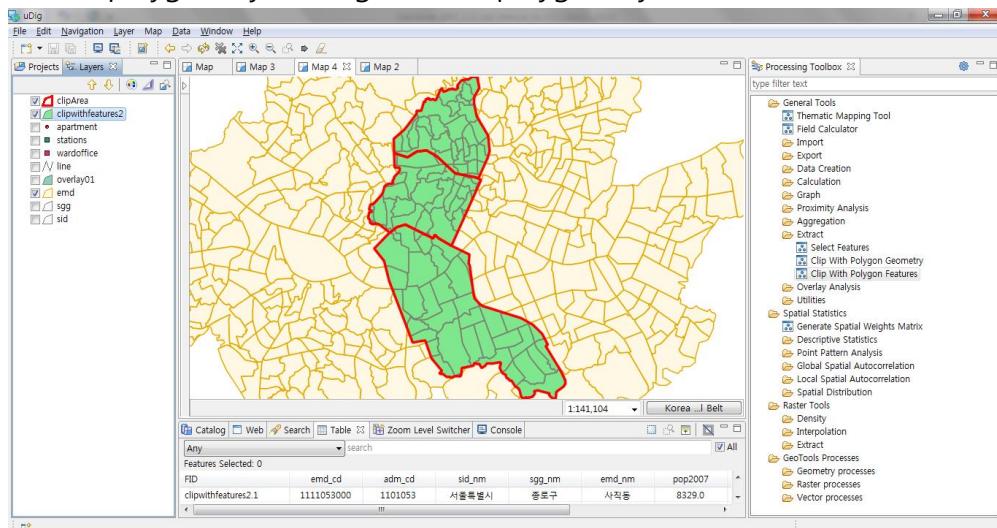
```

xmlns:foss="http://www.opengeospatial.net/foss"
    <wfs:Query typeName="foss:emd" />
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
    <ows:Identifier>clipFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
            <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss"
            <wfs:Query typeName="foss:overlay02" />
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
        <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows an example of clipping Eup-Meyon-Dong administrative district polygon layer using random polygon layer.



4.2.3.4. Merge Features

A few layers with the same feature type (**features**) merge as one layer.

■ Syntax

MergeFeatures (List<SimpleFeatureCollection> features): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
features	Input feature collections to be merge.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **Features** parameter must be the same feature type.
- Since the **features** parameter is collection type, there is more than 1 feature layer available and the request below is referred.
- Schema of output layer is different with schema of the first layer of **features** parameter.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:MergeFeatures</ows:Identifier>
```

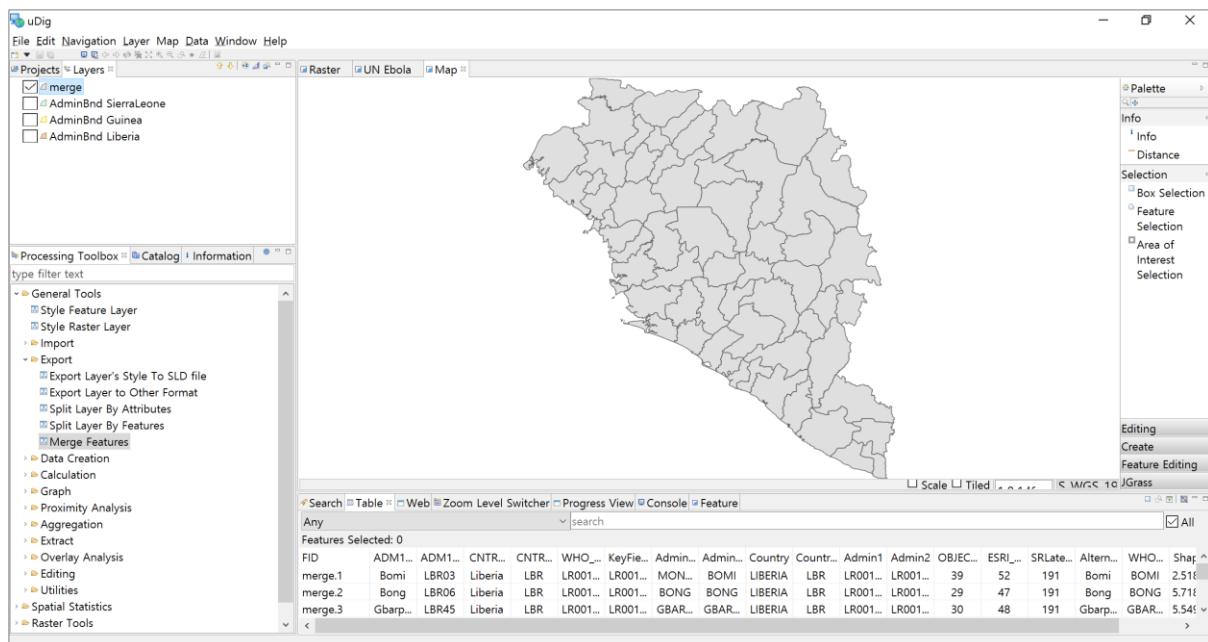
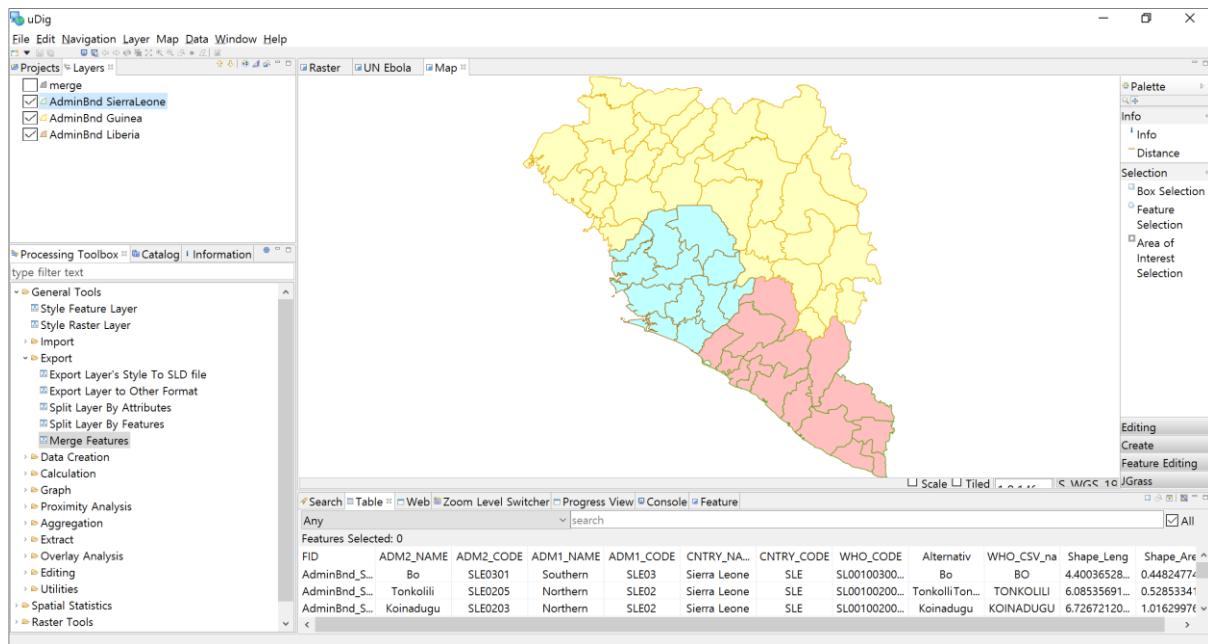
```

<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>features</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
        xmlns:foss="http://www.opengeospatial.net/foss">
          <wfs:Query typeName="foss:ssga"/>
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>features</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
        xmlns:foss="http://www.opengeospatial.net/foss">
          <wfs:Query typeName="foss:ssgb"/>
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

This is the result of mering 3 polygon layers into 1 layer.



4.2.4. Overlay

These processes are for overlay analysis between two layers such as Union, Intersect, Symmetrical Difference, Difference, Identity, Update etc.

4.2.4.1. Union

Conducts Union Overlay analysis between two input feature layers (**inputFeatures**).

Output layer retains all attribute values of two features.

■ Syntax

Union (SimpleFeatureCollection inputFeatures, SimpleFeatureCollection overlayFeatures):

SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features.	Complex	✓
overlayFeatures	Overlay features.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Input feature layer can be points, lines and polygons, and feature type of output layer is the same with **inputFeatures** layer
- Output layers contain all attribute values of **inputFeatures**, and **overlayFeatures**.

■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

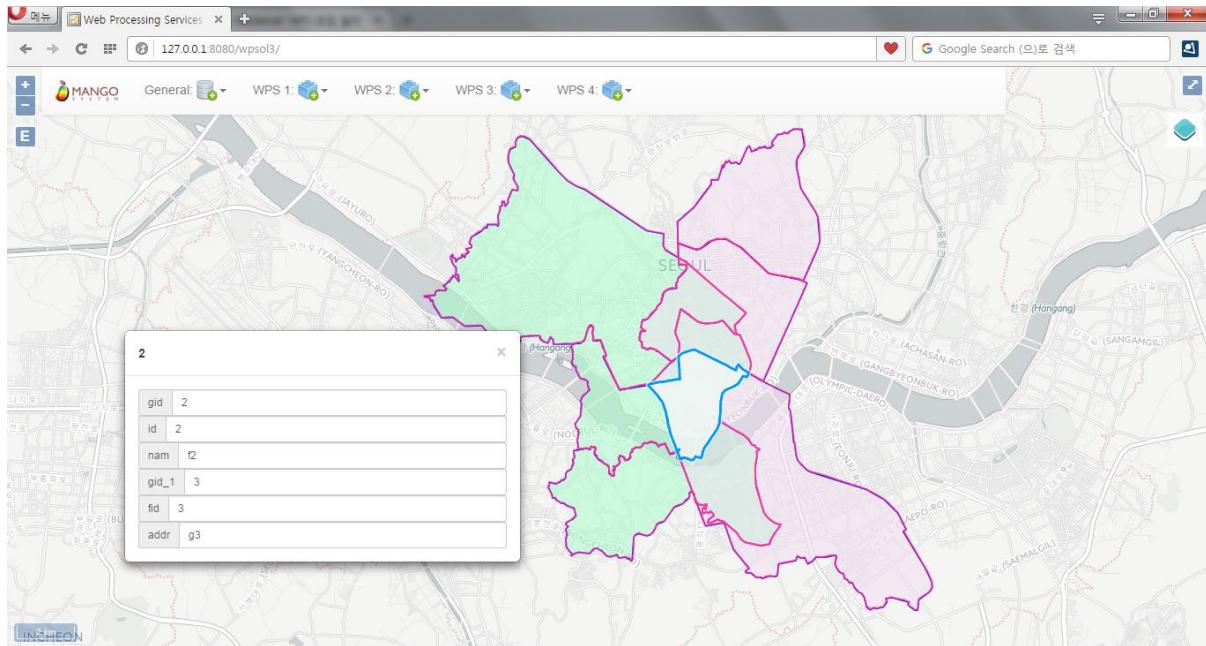
```

xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:Union</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3" />
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>overlayFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3" />
        </wps:Body>
      </wps:Reference>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of Union Overlay analysis between two polygon layers. Attribute values of two layers are all contained.



4.2.4.2. Intersect

Conducts Intersect Overlay analysis between two input feature layers (**inputFeatures**).

Output layer retains all attribute values of two features.

■ Syntax

```
Intersect (SimpleFeatureCollection inputFeatures, SimpleFeatureCollection  
overlayFeatures): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features.	Complex	✓
overlayFeatures	Overlay features.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Input feature layers can be points, lines and polygons, and feature type of output layer is the same with **inputFeatures** layer.
- Output layers contains all attribute value of **inputFeatures**, and **overlayFeatures**.

■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>  
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"  
xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"  
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"  
xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"  
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
```

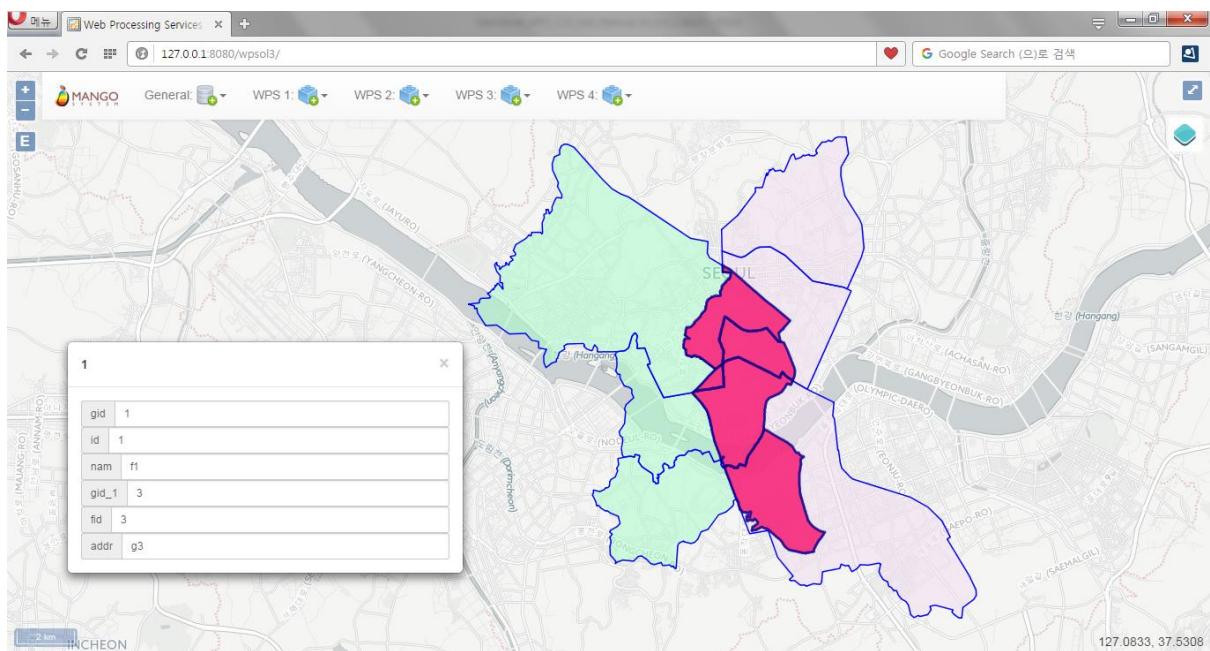
```

<ows:Identifier>statistics:Intersect</ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
        xmlns:foss="http://www.opengeospatial.net/foss">
          <wfs:Query typeName="foss:overlay01" />
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>overlayFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
        xmlns:foss="http://www.opengeospatial.net/foss">
          <wfs:Query typeName="foss:overlay02" />
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of Intersect Overlay analysis between two polygon layers. Attribute values of two layers are all contained.



4.2.4.3. Difference

Conducts Difference Overlay analysis between two polygon feature layers (**inputFeatures**).

■ Syntax

Difference (SimpleFeatureCollection inputFeatures, SimpleFeatureCollection differenceFeatures): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features.	Complex	✓
differenceFeatures	Difference features.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Feature type of input layer is the same with **inputFeatures** layer.

■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:Difference</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
```

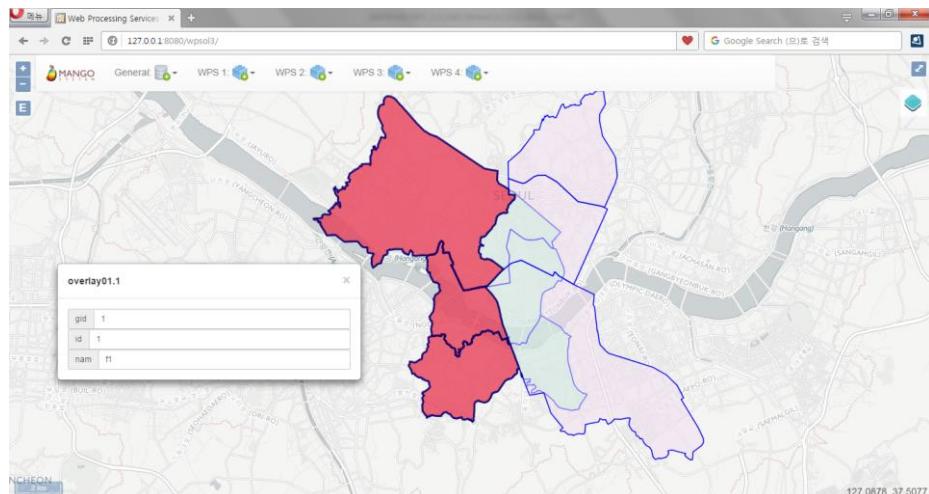
```

<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
    <wfs:Query typeName="foss:overlay01" />
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
    <ows:Identifier>differenceFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
            <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
            xmlns:foss="http://www.opengeospatial.net/foss">
                <wfs:Query typeName="foss:overlay02" />
            </wfs:GetFeature>
        </wps:Body>
    </wps:Reference>
</wps:Input>
</wps:Datalnputs>
<wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
        <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of Difference Overlay analysis between two polygon layers.



4.2.4.4. Symmetrical Difference

Conducts Symmetrical Difference Overlay analysis between two polygon feature layers (**inputFeatures**). Output layer retains all attribute values of two features.

■ Syntax

SymDifference (SimpleFeatureCollection inputFeatures, SimpleFeatureCollection differenceFeatures): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features.	Complex	✓
differenceFeatures	Difference features.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Input layers must be polygon feature types, and the feature type of output layer is the same with **inputFeatures** layer.
- Output layers contains all attribute value of **inputFeatures**, and **overlayFeatures**.

■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:SymDifference</ows:Identifier>
  <wps:DataInputs>
```

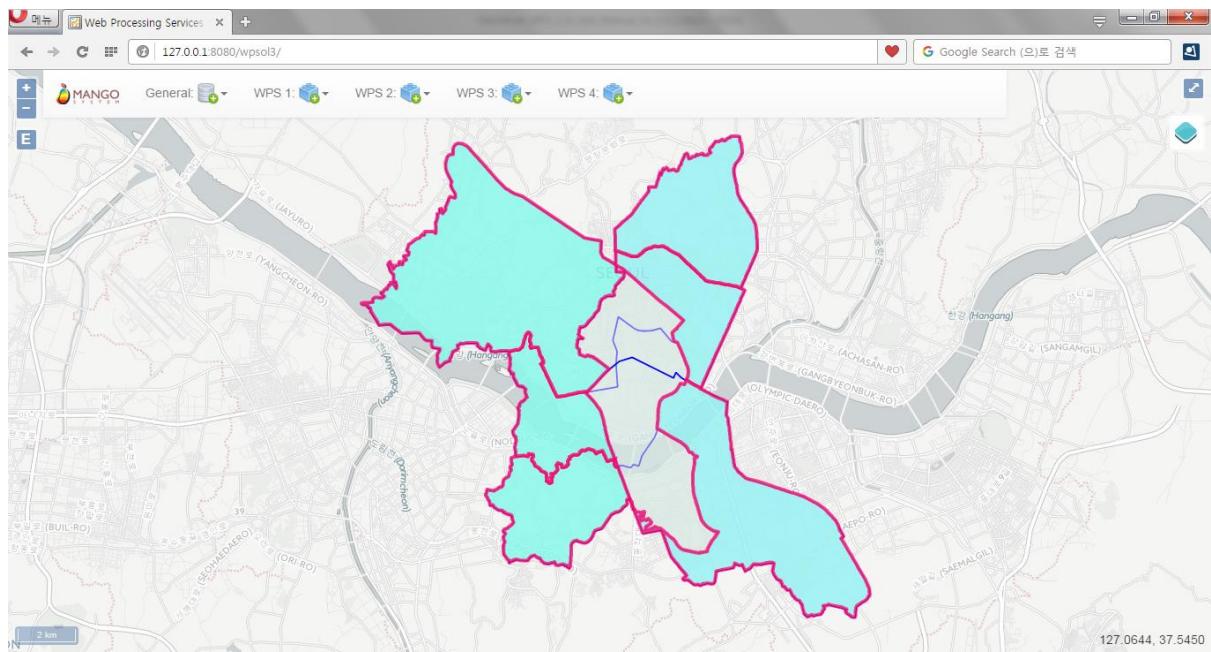
```

<wps:Input>
  <ows:Identifier>inputFeatures</ows:Identifier>
  <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
    <wps:Body>
      <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
      xmlns:foss="http://www.opengeospatial.net/foss">
        <wfs:Query typeName="foss:overlay01" />
      </wfs:GetFeature>
    </wps:Body>
  </wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>differenceFeatures</ows:Identifier>
  <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
    <wps:Body>
      <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
      xmlns:foss="http://www.opengeospatial.net/foss">
        <wfs:Query typeName="foss:overlay02" />
      </wfs:GetFeature>
    </wps:Body>
  </wps:Reference>
</wps:Input>
</wps:Datalnputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of Symmetrical Difference Overlay analysis between two polygon layers. Attribute values of two layers are all contained.



4.2.4.5. Identity

Conducts Difference Overlay analysis between two polygon feature layers (**inputFeatures**).

■ Syntax

Identity (SimpleFeatureCollection inputFeatures, SimpleFeatureCollection identityFeatures):
SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features.	Complex	✓
identityFeatures	Identity features.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **InputFeatures**, and **identityFeatures** must be Polygon or MultiPolygon feature types.
- Output layers contain all attribute values of **inputFeatures**, and **overlayFeatures**.

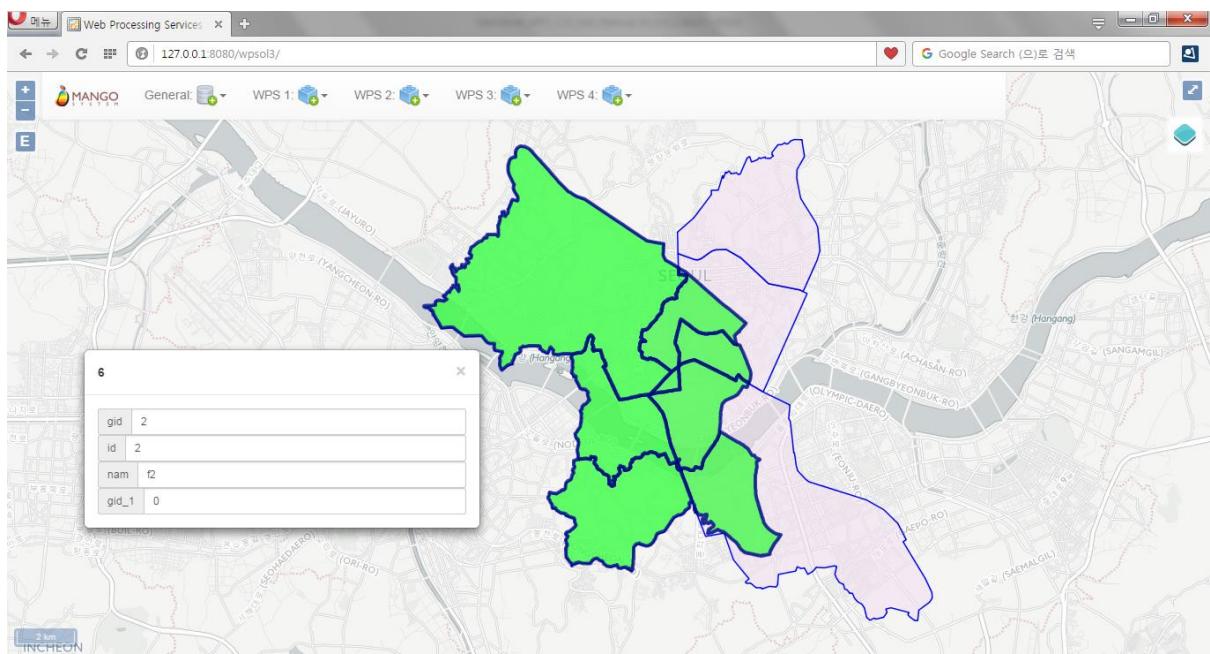
■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:Identity</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
```

```
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
  <wps:Body>
    <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
      xmlns:foss="http://www.opengeospatial.net/foss">
      <wfs:Query typeName="foss:overlay01" />
    </wfs:GetFeature>
  </wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>identityFeatures</ows:Identifier>
  <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
    <wps:Body>
      <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
        xmlns:foss="http://www.opengeospatial.net/foss">
        <wfs:Query typeName="foss:overlay02" />
      </wfs:GetFeature>
    </wps:Body>
  </wps:Reference>
</wps:Input>
</wps:Datalnputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of Identity Overlay analysis between two polygon layers.



4.2.4.6. Update

Conducts Update Overlay analysis between two polygon feature layers (**inputFeatures**).

■ Syntax

Update (SimpleFeatureCollection inputFeatures, SimpleFeatureCollection updateFeatures):

SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features.	Complex	✓
updateFeatures	Update features.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **InputFeatures**, and **identityFeatures** must be Polygon or MultiPolygon feature types.
- Output layers contain all attribute values of **inputFeatures**, and **overlayFeatures**.

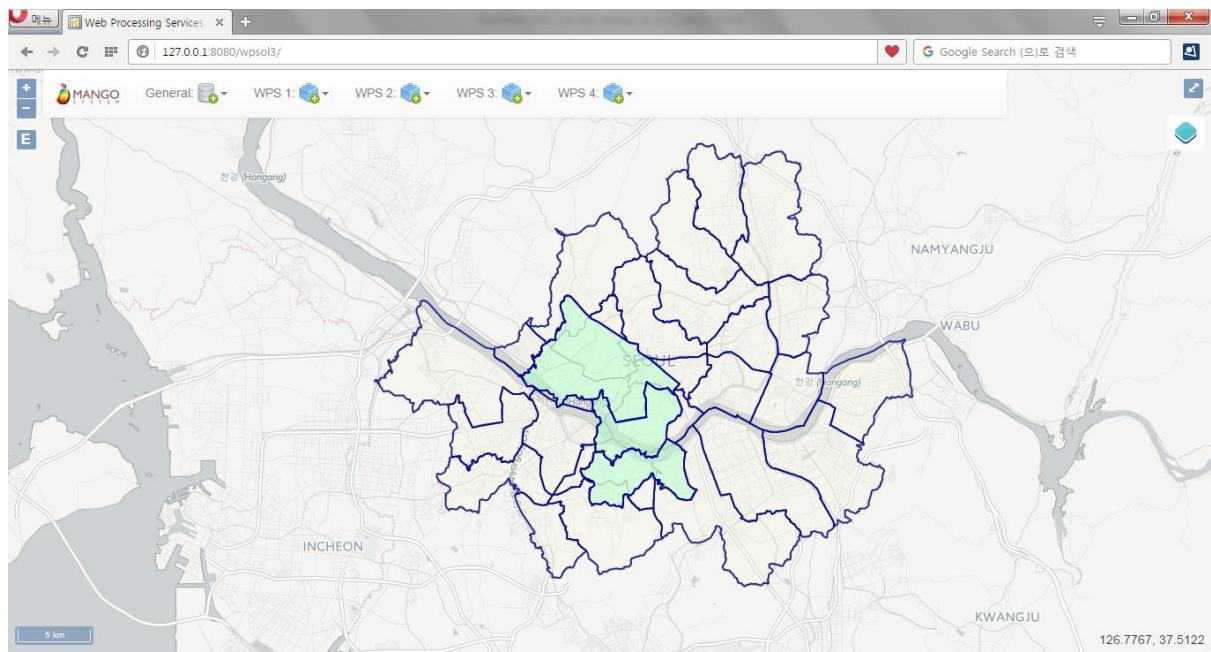
■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:Update</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
```

```
<wps:Body>
  <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
  xmlns:foss="http://www.opengeospatial.net/foss">
    <wfs:Query typeName="foss:ssg" />
  </wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>updateFeatures</ows:Identifier>
  <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
    <wps:Body>
      <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
      xmlns:foss="http://www.opengeospatial.net/foss">
        <wfs:Query typeName="foss:overlay01" />
      </wfs:GetFeature>
    </wps:Body>
  </wps:Reference>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of Identity Overlay analysis between two polygon layers. Overlap region of Input layer and Update layer is replaced by the features of Update layer.



4.2.5. Proximity

These processes are for distance calculation and analysis such as buffer.

4.2.5.1. Buffer Features using Expression

Conducts Buffer analysis using user-defined buffer distance (**distance**), buffer field or buffer expression formula.

■ Syntax

```
BufferFeatures (SimpleFeatureCollection inputFeatures, Expression distance, int quadrantSegments): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input features to be buffered.	Complex	✓
distance	The distance expression used to create distance. Ex) 1000 or [field] or [field] * 0.5 etc...	Literal	✓
quadrantSegments	The number of line segments used to represent a quadrant of a circle. Default is 8.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **InputFeatures** can be points, lines and polygons.
- **Distance expression** can be numbers, and functions¹ returned by number.

■ Request Examples

¹ http://docs.geoserver.org/stable/en/user/filter/function_reference.html 참조

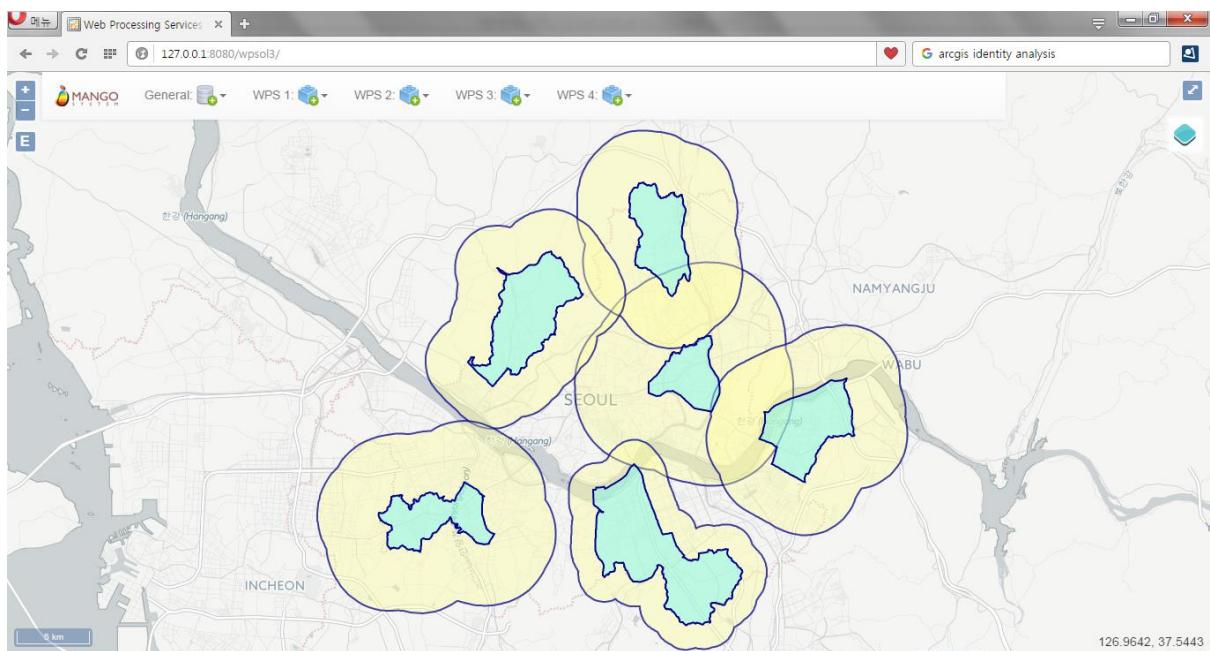
```

<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:BufferFeatures</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
            xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:randomsgg" />
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>distance</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>[pop_den] / 2.0) * 0.5</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of conducting buffer by using attribute values of polygon layers and displaying the buffer distance as Expression ([pop_den] / 2.0) * 0.5).



4.2.5.2. Single Sided Buffer

Conducts One-direction Buffer analysis using user-defined buffer distance (**distance**), buffer field or buffer expression formula.

■ Syntax

SingleSidedBuffer (SimpleFeatureCollection inputFeatures, Expression distance, Integer quadrantSegments): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input features to be buffered.	Complex	✓
distance	The distance expression used to create distance. Ex) 1000 or [field] or [field] * 0.5 etc...	Literal	✓
quadrantSegments	The number of line segments used to represent a quadrant of a circle. Default is 8.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **InputFeatures** must be lines.
- **Distance expression** can be numbers, and functions² returned by number.
- Buffer will be created on the left of line progression direction if the **distance** parameter value is positive, or right if the distance parameter value is negative.

■ Request Examples

² http://docs.geoserver.org/stable/en/user/filter/function_reference.html 참조

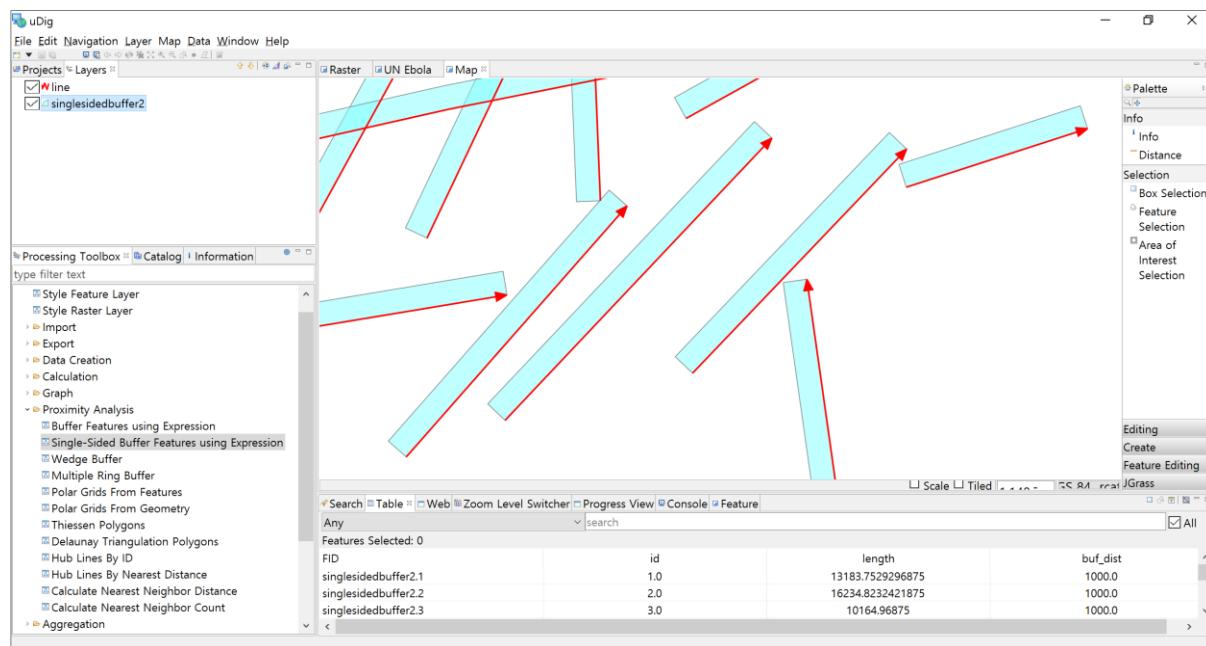
```

<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:SingleSidedBuffer</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:line"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>distance</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>250</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

■ Response

If the buffer distance value is positive as following, creates buffer on the left of the line progression direction



4.2.5.3. Multiple Ring Buffer

Conducts Buffer analysis based on comma-separated buffer distance (**distances**).

■ **Syntax**

MultipleRingBuffer (SimpleFeatureCollection inputFeatures, String distances, Boolean outsideOnly, Boolean dissolve): SimpleFeatureCollection

■ **Parameters**

■ **Data Inputs**

Identifier	Description	Type	Required
inputFeatures	The input features to be buffered.	Complex	✓
distances	The comma separated list of buffer distances. Ex) 250,500,750,1000	Literal	✓
outsideOnly	The area inside of the input polygon features will excluded from the resulting buffer. Default is True.	Literal	-
dissolve	Determines if buffers will be dissolved to resemble rings around the input features. Default is False.	Literal	-

■ **Process Outputs**

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ **Constraints**

- **InputFeatures** can be points, lines and polygons.
- Unit of distance is the same with that of **inputFeatures** coordinate system.
- If **Dissolve** parameter is true, attribute value of **inputFeatures** is ignored and only distance value is retained. If Dissolve parameter is false, attribute value of inputFeatures is retained.

■ **Request Examples**

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
```

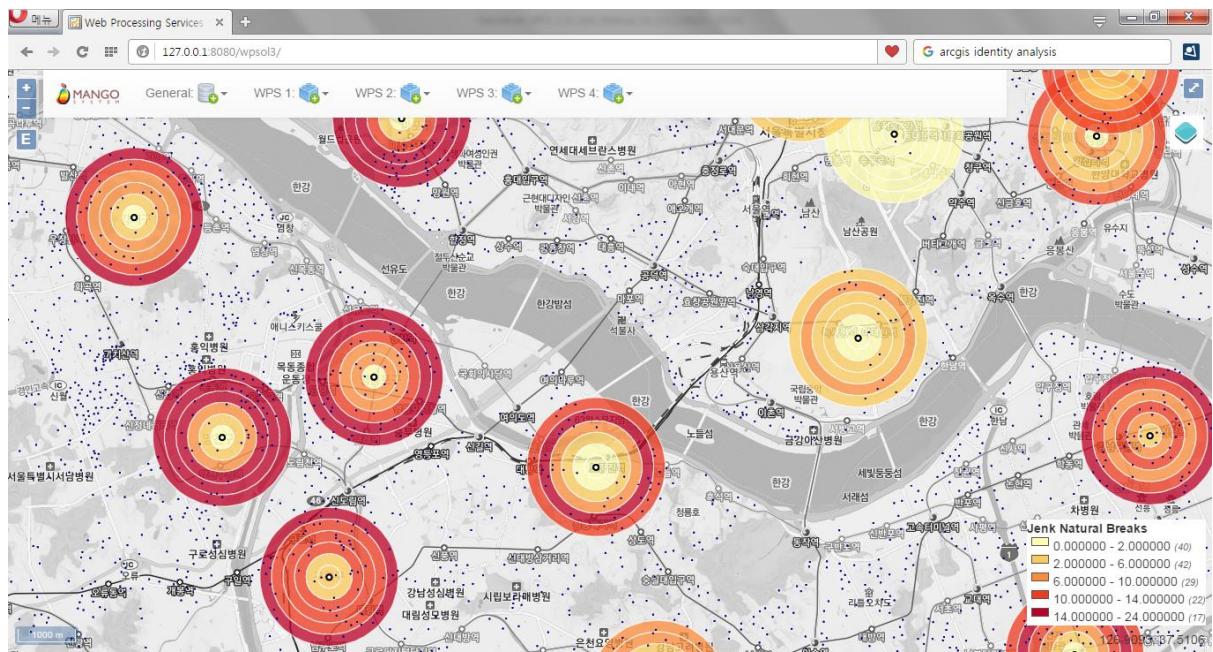
```

xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:MultipleRingBuffer</ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3">
<wfs:Query typeName="foss:wardoffice"/>
</wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>distances</ows:Identifier>
    <wps>Data>
      <wps:LiteralData>250,500,750,1000,1250,1500</wps:LiteralData>
    </wps>Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>outsideOnly</ows:Identifier>
    <wps>Data>
      <wps:LiteralData>True</wps:LiteralData>
    </wps>Data>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of processing buffer analysis with a radius of 250,500,750,1000,1250,1500 meters from municipal office point.



4.2.5.4. Wedge Buffer

Conducts Wedge Buffer analysis using point layer (**pointFeatures**) and attributes such as **azimuth**, **wedgeAngle** and radius (**innerRadius**, **outerRadius**).

■ Syntax

WedgeBuffer (SimpleFeatureCollection pointFeatures, Expression azimuth, Expression wedgeAngle, Expression innerRadius, Expression outerRadius): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
pointFeatures	The point features.	Complex	✓
azimuth	The azimuth (compass direction) expression. Ex) 45 or [field] or [field] * 0.5 etc...	Literal	✓
wedgeAngle	The wedge angle expression. Ex) 45 or [field] or [field] * 0.5 etc...	Literal	✓
innerRadius	The inner radius expression. The default is 0. Ex) 25 or [field] or [field] * 0.5 etc...	Literal	-
outerRadius	The outer radius expression. Ex) 100 or [field] or [field] * 0.5 etc...	Literal	✓

■ Process Outputs

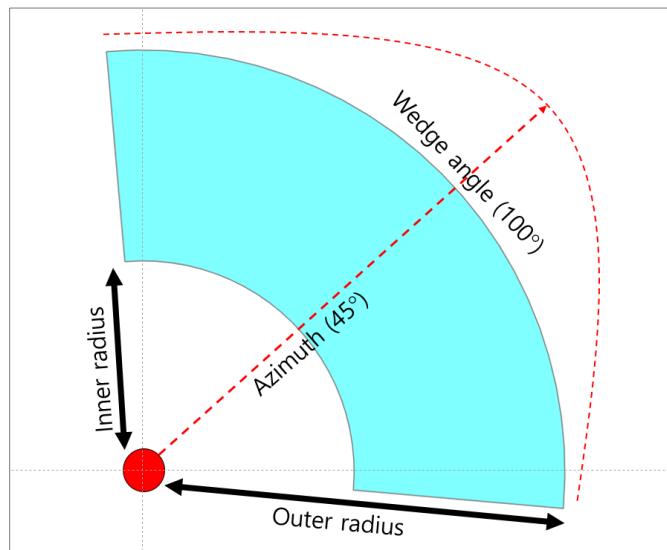
Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **PointFeatures** must be point types.
- **Azimuth**, **wedgeAngle**, **innerRadius**, **outerRadius** **expressions** can be numbers, and functions³ returned by number.

³ http://docs.geoserver.org/stable/en/user/filter/function_reference.html 참조

- Larger value between **innerRadius**, **outerRadius** parameter value is used as **outerRadius** value. At least one of the two values should be greater than 0.
- Parameter value will be explained next.



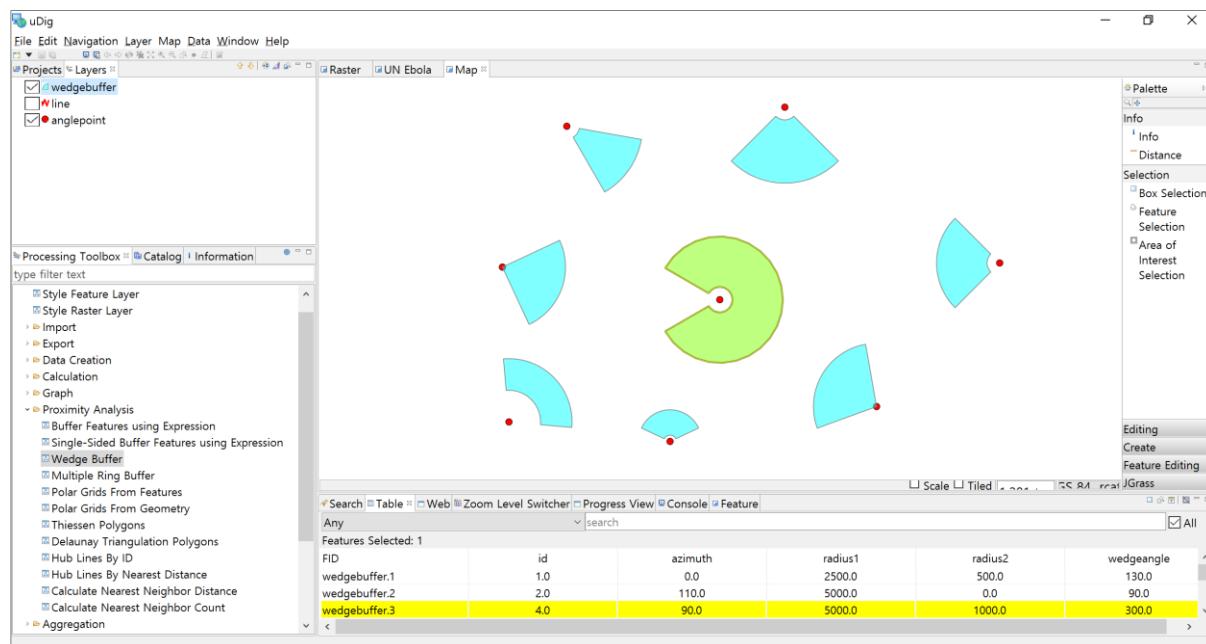
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:WedgeBuffer</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>pointFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:anglepoint"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>azimuth</ows:Identifier>
    </wps:Input>
  </wps:DataInputs>
  <wps:Process>
    <wps:Identifier>statistics:WedgeBuffer</wps:Identifier>
    <wps:InputData identifier="pointFeatures">
      <wps:Reference>
        <wps:Value>http://geoserver/wfs</wps:Value>
      </wps:Reference>
    </wps:InputData>
    <wps:InputData identifier="azimuth">
      <wps:Value>45</wps:Value>
    </wps:InputData>
    <wps:InputData identifier="innerRadius">
      <wps:Value>0</wps:Value>
    </wps:InputData>
    <wps:InputData identifier="outerRadius">
      <wps:Value>100</wps:Value>
    </wps:InputData>
  </wps:Process>
  <wps:Output>
    <ows:Identifier>result</ows:Identifier>
  </wps:Output>
</wps:Execute>
```

```
<wps:Data>
  <wps:LiteralData>azimuth</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>wedgeAngle</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>wedgeangle</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>innerRadius</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>radius1</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>outerRadius</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>radius2</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of processing Wedge buffer using attributes of point layer.



4.2.5.5. Near (Nearest Distance & Attributes)

Calculates distances and attribute values of the nearest near features (**nearFeatures**) of input feature layer (**inputFeatures**).

■ **Syntax**

```
Near (SimpleFeatureCollection inputFeatures, SimpleFeatureCollection nearFeatures,  
String nearIdField, Double maximumDistance): SimpleFeatureCollection
```

■ **Parameters**

■ **Data Inputs**

Identifier	Description	Type	Required
inputFeatures	Input Features.	Complex	✓
nearFeatures	Near Features.	Complex	✓
nearIdField	Near ID field.	Literal	-
maximumDistance	Maximum distance.	Literal	-

■ **Process Outputs**

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ **Constraints**

- Unless setting **nearIdField**, uses inside ID of feature.
- **InputFeatures**, **nearFeatures** can be points, lines and polygons, and the nearest distance between two Geometry is calculated.
- **MaximumDistance** is set and Null value is input when there is no feature within the set distance.

■ **Request Examples**

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"  
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"  
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
```

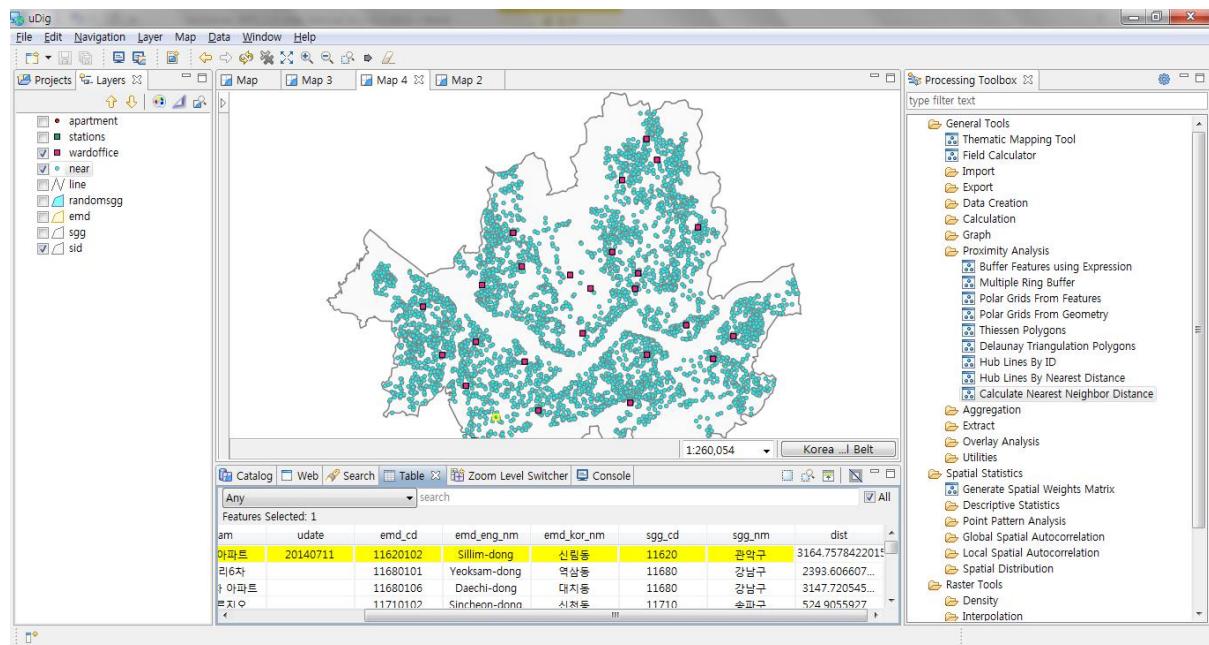
```

xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:Near</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:apartment"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>nearFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:wardoffice"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    </wps:DataInputs>
    <wps:ResponseForm>
      <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
        <ows:Identifier>result</ows:Identifier>
      </wps:RawDataOutput>
    </wps:ResponseForm>
  </wps:Execute>

```

■ Response

The following figure shows the result of processing Near analysis between apartments and municipal offices. It is calculated using the nearest distance.



4.2.5.6. Nearest Neighbor Count

Calculates number of near features (**nearFeatures**) inside the radius based on the input feature layers (**inputFeatures**).

■ Syntax

```
NearestNeighborCount (SimpleFeatureCollection inputFeatures, String countField,  
SimpleFeatureCollection nearFeatures, Double searchRadius): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input Features.	Complex	✓
countField	Count field. The default is count.	Literal	-
nearFeatures	Near Features.	Complex	✓
searchRadius	Search radius. Search radius must be greater than 0.	Literal	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **InputFeatures**, **nearFeatures** can be points, lines and polygons, and the nearest distance between two Geometry is calculated.
- Unit of **searchRadius** parameter is the same with distance unit of **inputFeatures**, which must be greater than 0.

■ Request Examples

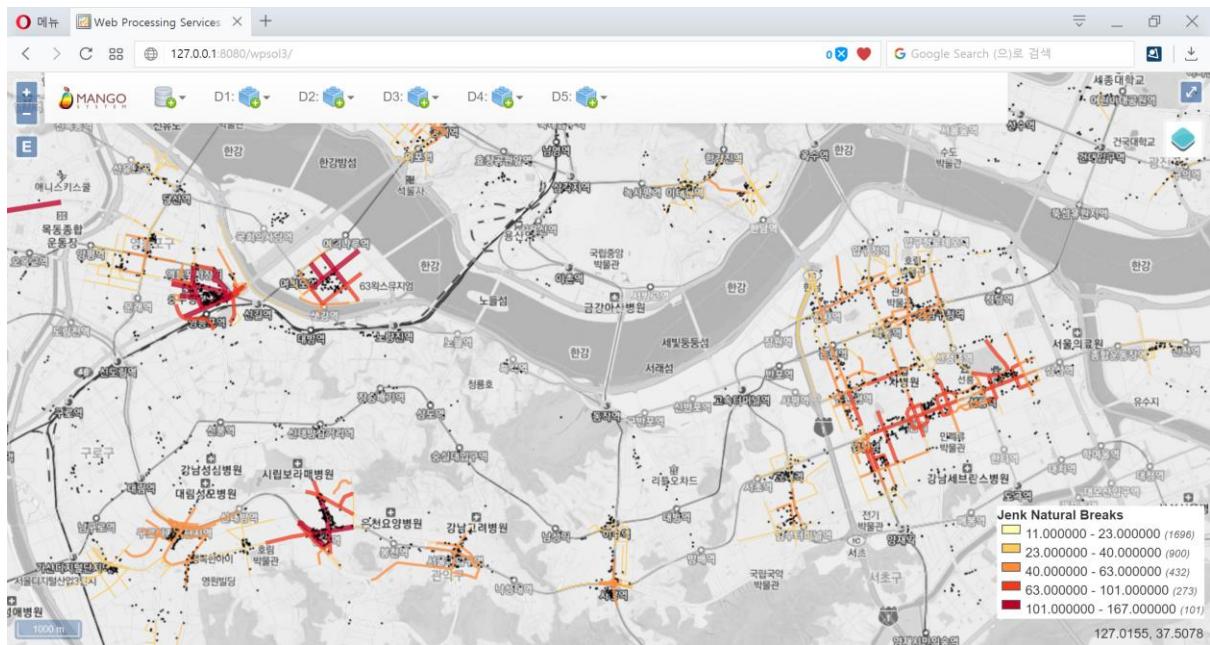
```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"  
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"  
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"  
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"  
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0"
```

```
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:NearestNeighborCount</ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
          xmlns:foss="http://www.opengeospatial.net/foss"
            <wfs:Query typeName="foss:road"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>countField</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>pub_cnt</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>nearFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
            xmlns:foss="http://www.opengeospatial.net/foss"
              <wfs:Query typeName="foss:pubs"/>
              </wfs:GetFeature>
            </wps:Body>
          </wps:Reference>
        </wps:Input>
        <wps:Input>
          <ows:Identifier>searchRadius</ows:Identifier>
          <wps:Data>
            <wps:LiteralData>500</wps:LiteralData>
          </wps:Data>
        </wps:Input>
      </wps:DataInputs>
      <wps:ResponseForm>
        <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
          <ows:Identifier>result</ows:Identifier>
        </wps:RawDataOutput>
      </wps:ResponseForm>
```

```
</wps:Execute>
```

■ Response

The following figure shows the result of graduated symbol representing the number of points within a 500-meter radius from the main roads.



4.2.5.7. Polar Grids from Geometry

Creates polar grid based on the reference Geometry and a comma-separated radius (**radius**).

■ Syntax

PolarGridsFromGeometry (Geometry origin, String radius, RadialType radialType, Integer sides): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
origin	The center features of polar grids.	Complex	✓
radius	The list of radius(unit:data unit): Ex) 200, 300, 400, 500.	Literal	✓
radialType	Radial Type: Polar (Default), Base.	Literal	-
sides	The number of sides. Default is 8	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Values of angle and radius of each cell in input layers are calculated.
- If default value of **sides** parameter (8) is used, adds azimuth field, and calculates direction value of NE, N, NW, W, SW, S, SE, E.

■ Request Examples

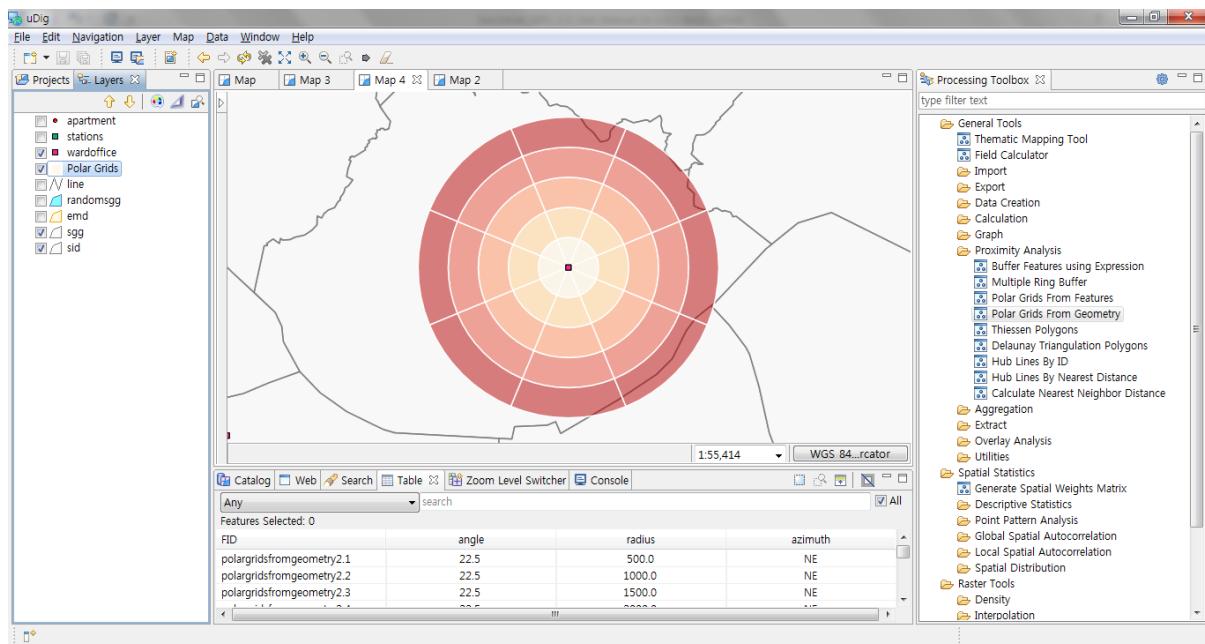
```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0"
```

```
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:PolarGridsFromGeometry</ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>origin</ows:Identifier>
    <wps:Data>
      <wps:ComplexData mimeType="application/wkt"><![CDATA[POINT (14136522.58319524
4513573.676204068)]]></wps:ComplexData>
    </wps:Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>radius</ows:Identifier>
    <wps:Data>
      <wps:LiteralData>500, 1000, 1500, 2000, 2500</wps:LiteralData>
    </wps:Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>radialType</ows:Identifier>
    <wps:Data>
      <wps:LiteralData>Polar</wps:LiteralData>
    </wps:Data>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of creating Polar Grid with intervals of 500, 1000, 1500, 2000, 2500 meters with point (POINT (14136522.58319524 4513573.676204068)) as

the center.



4.2.5.8. Polar Grids from Features

Creates Polar grid based on the several Geometry of reference feature layers and a comma-separated radius (**radius**).

■ Syntax

```
PolarGridsFromFeatures (SimpleFeatureCollection origin, String radius, RadialType  
radialType, Integer sides): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
origin	The center features of polar grids.	Complex	✓
radius	The list of radius(unit:data unit): Ex) 200, 300, 400, 500.	Literal	✓
radialType	Radial Type: Polar (Default), Base.	Literal	-
sides	The number of sides.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Values of angle and radius of each cell in input layers are calculated.
- If default value of **sides** parameter (8) is used, adds azimuth field, and calculates direction value of NE, N, NW, W, SW, S, SE, E.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"  
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"  
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"  
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"  
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0"
```

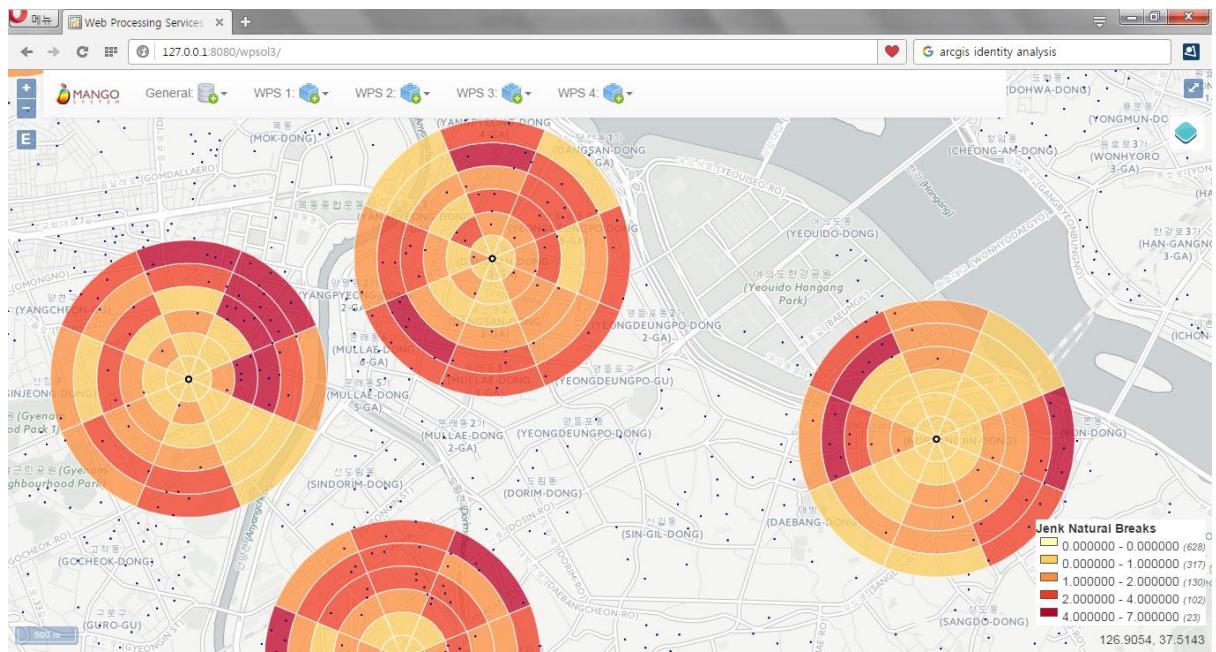
```

http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:PolarGridsFromFeatures</ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>origin</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlNs:foss="http://www.opengeospatial.net/foss">
          <wfs:Query typeName="foss:wardoffice"/>
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>radius</ows:Identifier>
    <wps>Data>
      <wps:LiteralData>500, 1000, 1500, 2000</wps:LiteralData>
    </wps>Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>radialType</ows:Identifier>
    <wps>Data>
      <wps:LiteralData>Polar</wps:LiteralData>
    </wps>Data>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of creating 8-directional Polar Grids with radius of 250,500,750,1000,1250,1500 meters from municipal offices, then calculating the number of apartments in every cell, and mapping out.



4.2.6. Aggregation

These processes are for overlapping one or more data and calculating new values.

4.2.6.1. Point Statistics

Calculates the numbers or descriptive statistics of point layers (**pointFeatures**) intersecting with the polygon layers (**inputFeatures**).

■ Syntax

PointStatistics (SimpleFeatureCollection inputFeatures, SimpleFeatureCollection pointFeatures, String countField, String statisticsFields): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The polygon features to be calculated.	Complex	✓
pointFeatures	The point features to be calculated.	Complex	✓
countField	The count field. count is a default	Literal	-
statisticsFields	Centroid(False, Default), Inside(True)	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **InputFeatures** must be polygon types, and **pointFeatures** must be point types.
- **CountField** stores the number of points contained within the polygons. The default value is count.
- **StatisticsFields** are input as [Function name, Field name] structure as follows, and the available functions are as follows. For example, Sum.pop, Mean.pop

Input	Return field name
First: String field, Dissolve object Feature's first value	FST_Field name

Last: String field, Dissolve object Feature's last value	LST_ Field name
Sum: Numeric field, Dissolve object Feature's sum value	SUM_ Field name
Mean: Numeric field, Dissolve object Feature's mean value	AVG_ Field name
Min: Numeric field, Dissolve object Feature's minimum value	MIN_ Field name
Max: Numeric field, Dissolve object Feature's maximum value	MAX_ Field name
Std: Numeric field, Dissolve object Feature's standard deviation value	STD_ Field name
Var: Numeric field, Dissolve object Feature's variance	VAR_ Field name
Range: Numeric field, Dissolve object Feature's range	RNG_ Field name
Count: Dissolve object Feature's number	CNT_ Field name

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:PointStatistics</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>polygonFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:ssg"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>pointFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:gasstation"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
  </wps:DataInputs>
</wps:Execute>
```

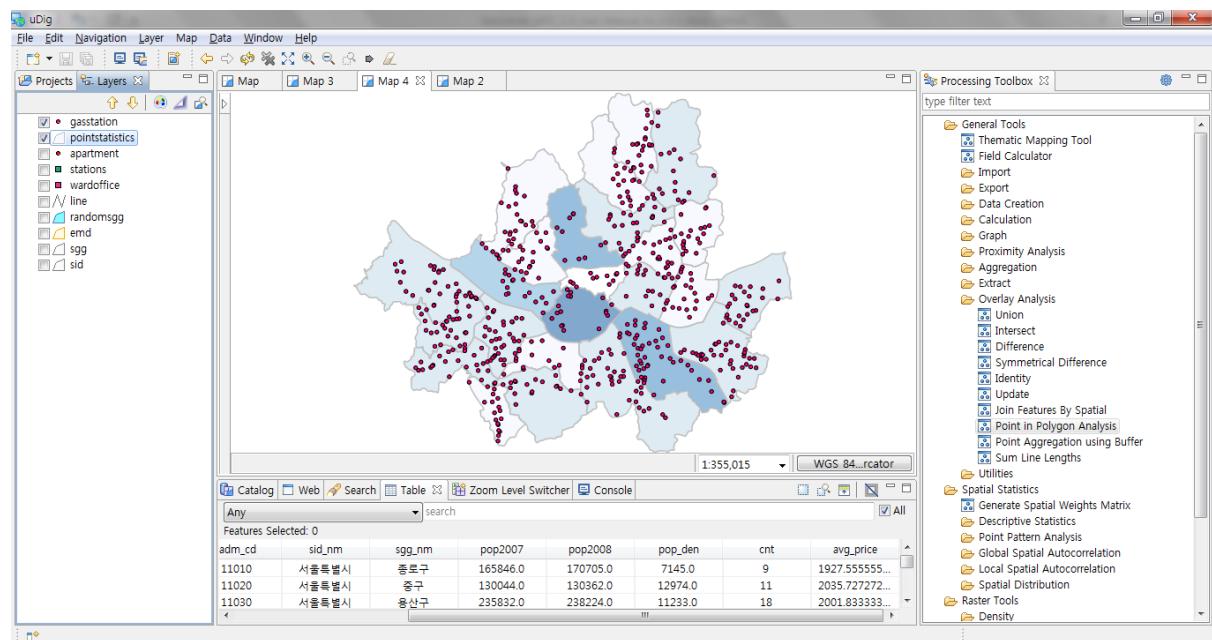
```

</wps:Input>
<wps:Input>
  <ows:Identifier>countField</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>cnt</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>statisticsFields</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>Mean.price</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of calculating and mapping the average price of gas stations in the administrative boundary (Si-Gun-Gu).



4.2.6.2. Aggregate (Union) Polygons

Dissolves the input polygon features (**polygonFeatures**) and creates one polygon feature layer.

■ Syntax

UnionPolygon (SimpleFeatureCollection polygonFeatures, Boolean preserveHole):
SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
polygonFeatures	The polygon features to be processed.	Complex	✓
preserveHole	Preserve or remove hole (interior ring).	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If **preserveHole** parameter is set to false, all interior rings are returned to polygons eliminated.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:UnionPolygon</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>polygonFeatures</ows:Identifier>
```

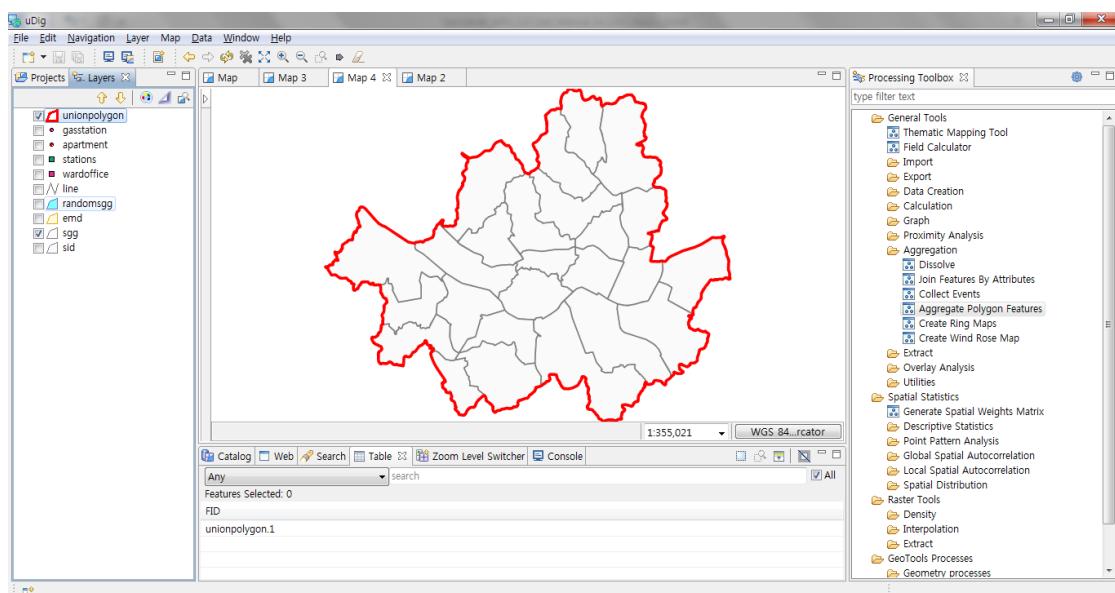
```

<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
  <wps:Body>
    <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
      xmlns:foss="http://www.opengeospatial.net/foss">
      <wfs:Query typeName="foss:ssg"/>
    </wfs:GetFeature>
  </wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>preserveHole</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>False</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

If there is any **InteriorRing(Hole)** due to the **preserveHole** parameter is false, eliminates interior rings(Hole).



4.2.6.3. Collect Events

If points have the same locations, or they are contained within a specified radius, merges them into one feature.

■ Syntax

CollectEvents (SimpleFeatureCollection inputFeatures, String countField, Double tolerance):
SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features representing event or incident data.	Complex	✓
countField	The field to be calculated coincident points count. icount (Default).	Literal	-
tolerance	The tolerance distance for considering two points equal.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Unless setting **countField**, uses default value of **countField**.
- If the **tolerance** value is 0, considers exactly matching points. If the **tolerance** value is greater than 0, the features within the **tolerance distance** are regarded as the same.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
```

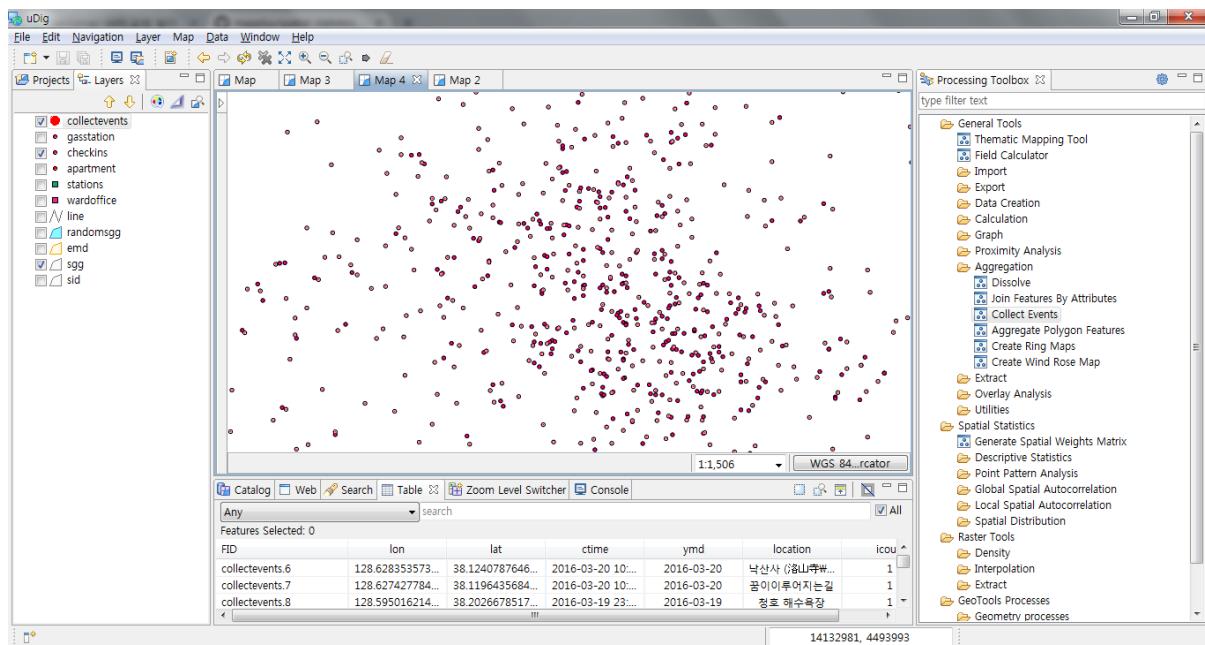
```

xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:CollectEvents </ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
      xmlns:foss="http://www.opengeospatial.net/foss">
          <wfs:Query typeName="foss:checkins"/>
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>countField</ows:Identifier>
    <wps:Data>
      <wps:LiteralData>icount</wps:LiteralData>
    </wps:Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>tolerance</ows:Identifier>
    <wps:Data>
      <wps:LiteralData>5</wps:LiteralData>
    </wps:Data>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of joining all the points within a set distance into one feature.



4.2.6.4. Spatial Join

Performs spatial join of two features based on spatial relationships. All attribute values of two feature layers are contained.

■ Syntax

SpatialJoin (SimpleFeatureCollection inputFeatures, SimpleFeatureCollection joinFeatures, SpatialJoinType joinType, Double searchRadius): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features.	Complex	✓
joinFeatures	Join features.	Complex	✓
joinType	Join Type. KeepAllRecord, OnlyMatchingRecord	Literal	-
searchRadius	Search Radius.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If **joinType** parameter value is set to **KeepAllRecord**, returns all features of **inputFeatures** that have not been spatial joined.
- If provides **searchRadius** parameter, performs join with features contained within **searchRadius**.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
```

```

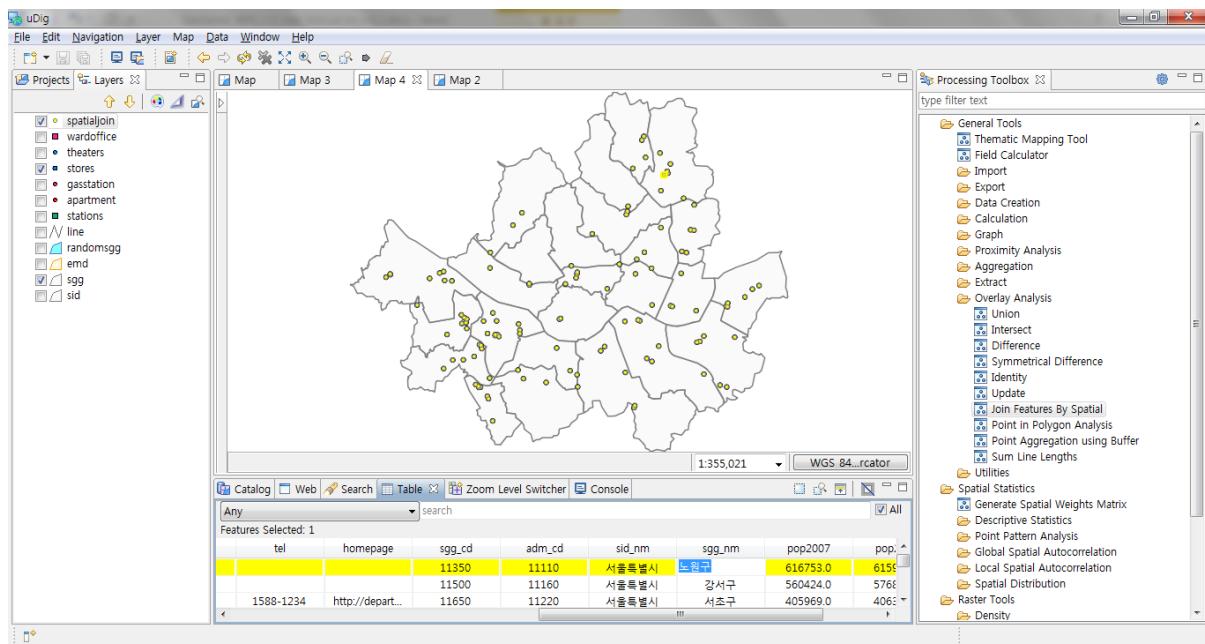
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:SpatialJoin</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:stores"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>joinFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:ssg"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>joinType</ows:Identifier>
<wps:Data>
<wps:LiteralData>KeepAllRecord</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of Spatial Join of information of the administrative boundary (Si-Gun-Gu) layer containing point layer of larger markets. For point or line

layers, you can set the search radius to get the properties of nearby features.



4.2.6.5. Attribute Join

Performs join using two feature layers (**inputFeatures**) and join fields. All attribute values of two feature layers are contained.

■ Syntax

```
AttributeJoin (SimpleFeatureCollection inputFeatures, String primaryKey,  
SimpleFeatureCollection joinFeatures, String foreignKey, Join.Type joinType):  
SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features.	Complex	✓
primaryKey	Primary key field.	Literal	✓
joinFeatures	Join features.	Complex	✓
foreignKey	Foreign key field.	Literal	✓
joinType	Join type. INNER, OUTER	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If **joinType** is set to INNER, returns joined features of **inputFeatures**.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"  
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"  
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"  
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"  
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0  
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">  
    <ows:Identifier>statistics:AttributeJoin</ows:Identifier>
```

```
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
        xmlns:foss="http://www.opengeospatial.net/foss">
          <wfs:Query typeName="foss:stores"/>
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>primaryKey</ows:Identifier>
    <wps:Data>
      <wps:LiteralData>sgg_cd</wps:LiteralData>
    </wps:Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>joinFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
        xmlns:foss="http://www.opengeospatial.net/foss">
          <wfs:Query typeName="foss:ssg"/>
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>foreignKey</ows:Identifier>
    <wps:Data>
      <wps:LiteralData>sgg_cd</wps:LiteralData>
    </wps:Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>joinType</ows:Identifier>
    <wps:Data>
      <wps:LiteralData>OUTER</wps:LiteralData>
    </wps:Data>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
```

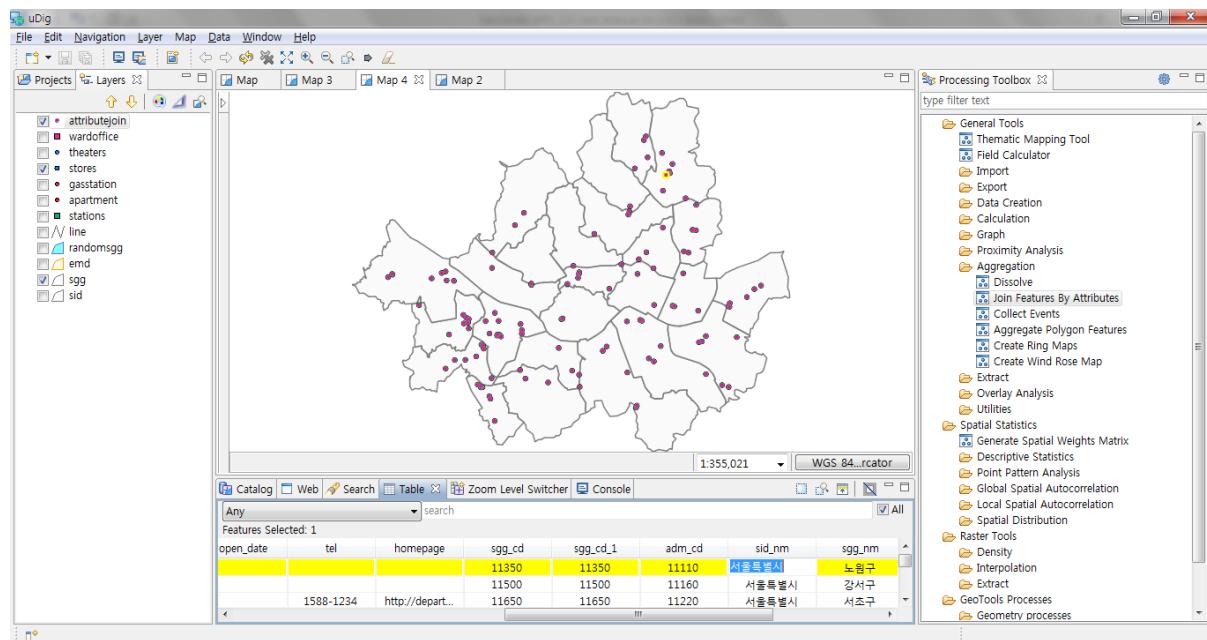
```

<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
  <ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of joining large markets with join field of code of the administrative district (Si-Gun-Gu).



4.2.6.6. Buffer Point Statistics

Inputs radius and calculates the numbers or statistics (Sum, Max, Min, Mean etc.) of attribute information of points within the radius.

■ Syntax

BufferPointStatistics (SimpleFeatureCollection inputFeatures, Double distance, SimpleFeatureCollection pointFeatures, String countField, String statisticsFields):
SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features.	Complex	✓
distance	Search distance.	Literal	✓
pointFeatures	Point features.	Complex	✓
countField	Count field. Default is count.	Literal	-
statisticsFields	Statistics Fields: Function.PropertyName(First, Last, Sum, Mean, Min, Max, Std, Count)	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **CountField** stores the number of points contained in the polygons. The default value is count.
- **StatisticsFields** are input as [Function name, Field name] structure as follows, and the available functions are as follows. For example, Sum.pop, Mean.pop

Input	Return field
First: String field, Dissolve object Feature's first value	FST_Field Name
Last: String field, Dissolve object Feature's last value	LST_Field Name

Sum: Numeric field, Dissolve object Feature's sum	SUM_Field Name
Mean: Numeric field, Dissolve object Feature's mean value	AVG_Field Name
Min: Numeric field, Dissolve object Feature's minimum value	MIN_Field Name
Max: Numeric field, Dissolve object Feature's maximum value	MAX_Field Name
Std: Numeric field, Dissolve object Feature's standard deviation	STD_Field Name
Var: Numeric field, Dissolve object Feature's variance	VAR_Field Name
Range: Numeric field, Dissolve object Feature's range	RNG_Field Name
Count: Dissolve object Feature's number	CNT_Field Name

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:BufferPointStatistics</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:wardoffice"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>distance</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>2000</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>pointFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
```

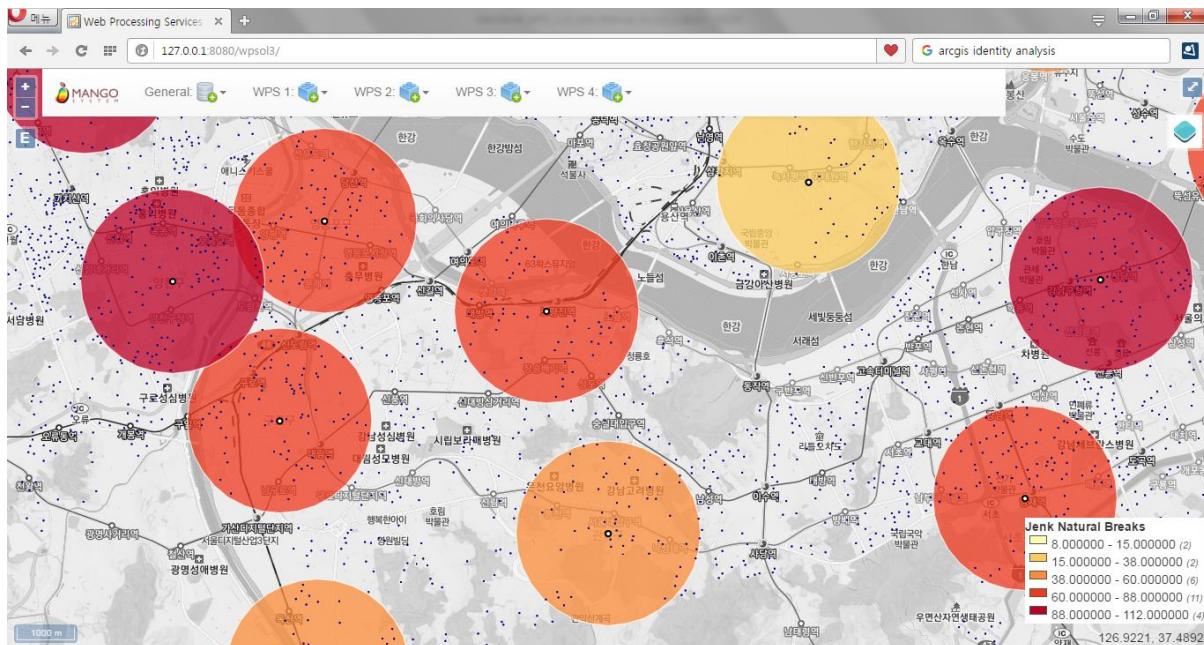
```

<wfs:Query typeName="foss:apartment"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>countField</ows:Identifier>
<wps:Data>
<wps:LiteralData>cnt</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of calculating and mapping the number of apartments within 2000 meter of municipal office.



4.2.6.7. Sum Line Lengths

Clips the line feature layers intersecting with the polygon features (**polygons**) and calculates numbers of intersecting line features and the sum of length.

■ Syntax

SumLineLength (SimpleFeatureCollection polygons, String lengthField, String countField, SimpleFeatureCollection lines): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
polygons	The polygon features that will be calculated.	Complex	✓
lengthField	The length field that will be calculated. sum_len is default.	Literal	✓
countField	The count field that will be calculated. line_cnt is default.	Literal	-
lines	The line features that will be calculated.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **LengthField** stores sum of cut length of lines intersecting with polygon features. The default value is sum_len.
- **CountField** stores the numbers of lines contained within the polygons. The default value is line_cnt.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
```

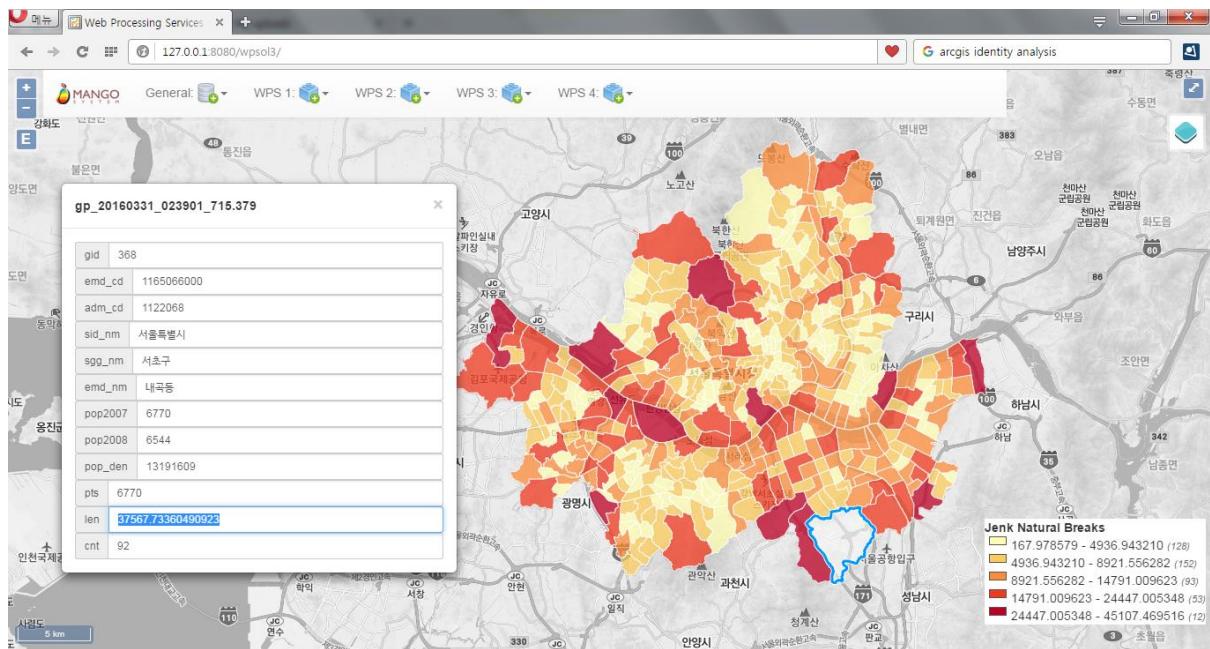
```

xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:SumLineLength</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>polygons</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:emd"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>lengthField</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>len</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>lines</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:road"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of calculating and mapping of the extension of main roads in the administrative boundary (Eup-Myeon-Dong).



4.2.6.8. Spatial Clump Map

Creates Spatial Clump Map using point features (**inputFeatures**) and radius expressions (**radius**).

■ Syntax

SpatialClumpMap (SimpleFeatureCollection inputFeatures, Expression radius, Integer quadrantSegments): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features to be buffered.	Complex	✓
radius	The radius expression used to create distance. Ex) 1000 or [field] or [field] * 0.5 etc...	Literal	✓
quadrantSegments	The number of line segments used to represent a quadrant of a circle. Default is 8.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- The default value of **quadrantSegments** parameter is 8.

■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:SpatialClumpMap</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
```

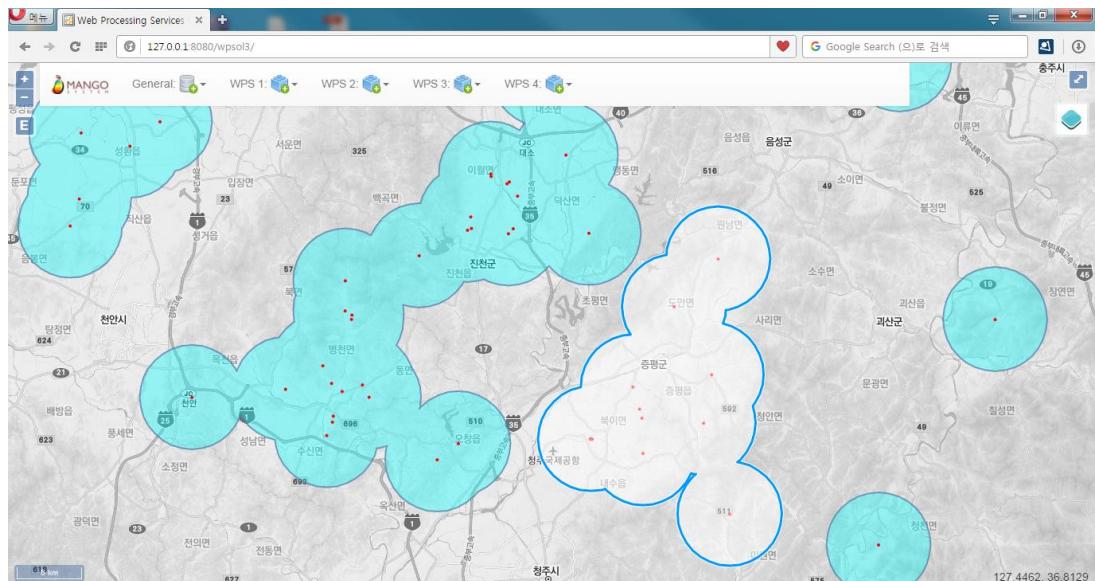
```

<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
    <wps:Body>
        <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
            xmlns:foss="http://www.opengeospatial.net/foss"
                <wfs:Query typeName="foss:outbreak" />
            </wfs:GetFeature>
        </wps:Body>
    </wps:Reference>
</wps:Input>
<wps:Input>
    <ows:Identifier>radius</ows:Identifier>
    <wps:Data>
        <wps:LiteralData>5000</wps:LiteralData>
    </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
        <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of Spatial Clump Map by setting the radius as 5Km.



4.2.6.9. Hexagonal Binning

Creates Hexagon grid using point features (**features**), extents, and radius and then creates polygon features counting for the numbers of points in each cell.

■ Syntax

HexagonalBinning (SimpleFeatureCollection **features**, Expression **weight**, ReferencedEnvelope **bbox**, Double **size**, Boolean **validGrid**): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
features	Input point features to be aggregated.	Complex	✓
weight	The numeric field or expression used to weight values. Ex) [field] or [field] * 0.5 etc...	Literal	-
bbox	The extent of the grids.	Complex	-
size	Size of the grids.	Literal	✓
validGrid	Returns only valid grid. Default is True.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If **bbox** is set to Null, **extent** of feature data is used.
- If gives **weight expression**, accumulates weight values.
- **Size** must be greater than 0.
- Default value of **validGrid** parameter is true, and only the grid of which number is more than 0 is returned.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?>
```

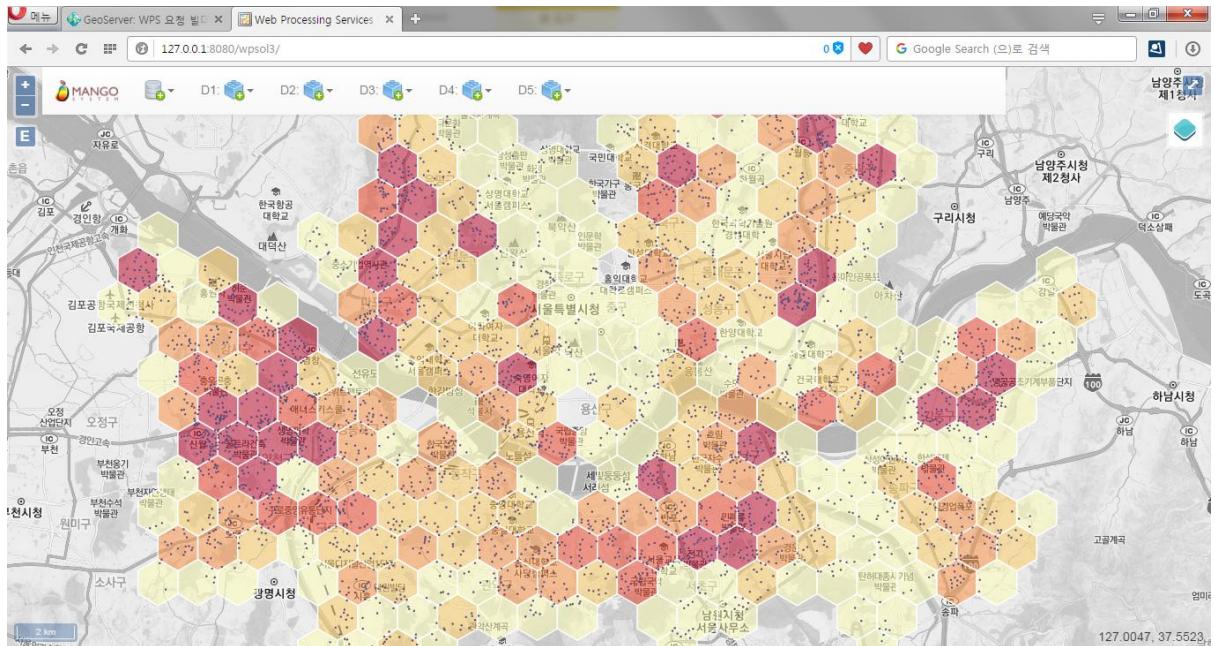
```

<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:HexagonalBinning</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>features</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
            xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:apartment"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>size</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>1500</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of creating and visualizing of hexagons with a size of 1000 meters created from apartment point data of Seoul.



4.2.6.10. *Rectangular Binning*

Creates rectangle grids using point features (**features**), extent (**bbox**), and radius (**width**, **height**) and then creates polygon features counting for the number of points in each cell.

■ Syntax

RectangularBinning (SimpleFeatureCollection features, Expression weight, ReferencedEnvelope bbox, Double width, Double height, Boolean validGrid):

SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
features	Input point features to be aggregated.	Complex	✓
weight	The numeric field or expression used to weight values. Ex) [field] or [field] * 0.5 etc...	Literal	-
bbox	The extent of the grids.	Complex	-
width	Width of the grids.	Literal	✓
height	Height of the grids.	Literal	✓
validGrid	Returns only valid grid. Default is True.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

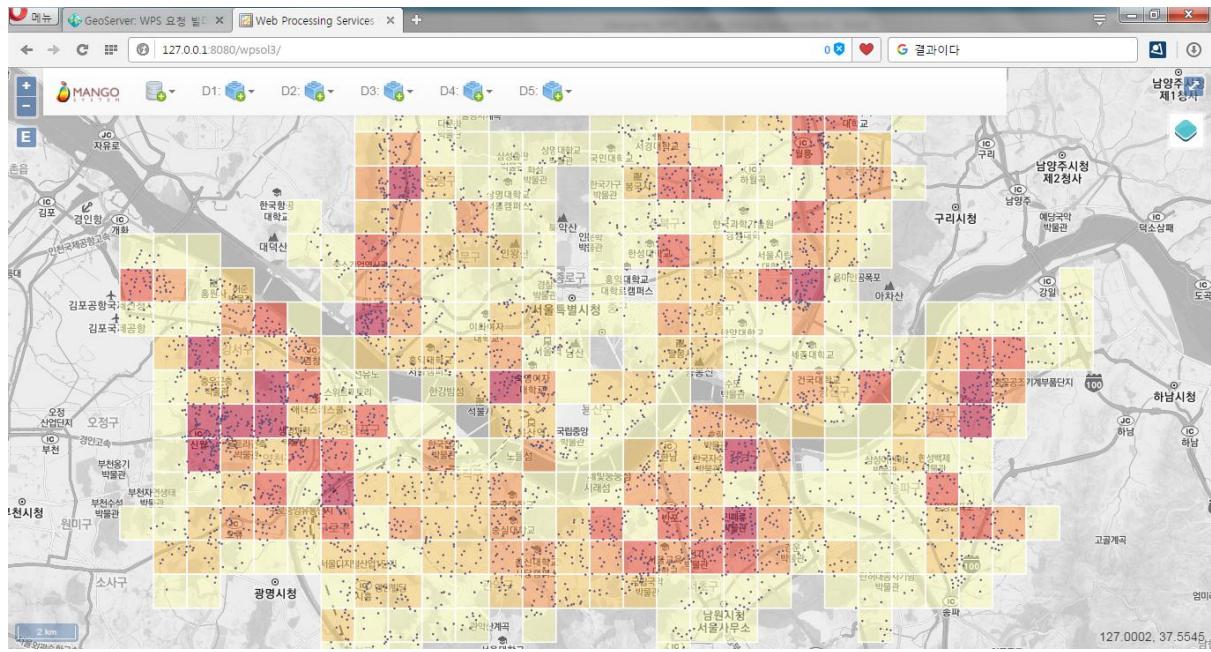
- If **bbox** is set to Null, uses **extent** of feature data.
- If gives **weight expression**, accumulates weight values.
- **Width, height** must be greater than 0.
- Default value of **validGrid** parameter is true, and only the grids of which number is more than 0 are returned.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RectangularBinning</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>features</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
            <wfs:Query typeName="foss:apartment"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>width</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>1500</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>height</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>1500</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of creating and visualizing of rectangles with a side length of 1500 meters created from apartment point data of Seoul.



4.2.6.11. Circular Binning

Creates circle grids using point features (**features**), extent (**bbox**), radius (**radius**) and then creates polygon features counting for the number of points in each cell.

■ Syntax

CircularBinning (SimpleFeatureCollection features, Expression weight, ReferencedEnvelope bbox, Double radius, Boolean validGrid): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
features	Input point features to be aggregated.	Complex	✓
weight	The numeric field or expression used to weight values. Ex) [field] or [field] * 0.5 etc...	Literal	-
bbox	The extent of the grids.	Complex	-
radius	Radiuss of the grids.	Literal	✓
validGrid	Returns only valid grid. Default is True.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If **bbox** is set to Null, uses **extent** of feature data.
- If gives **weight expression**, accumulates weight values.
- **Radius** must be greater than 0.
- Default value of **validGrid** parameter is true, and only the grids of which numbers are more than 0 are returned.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?>
```

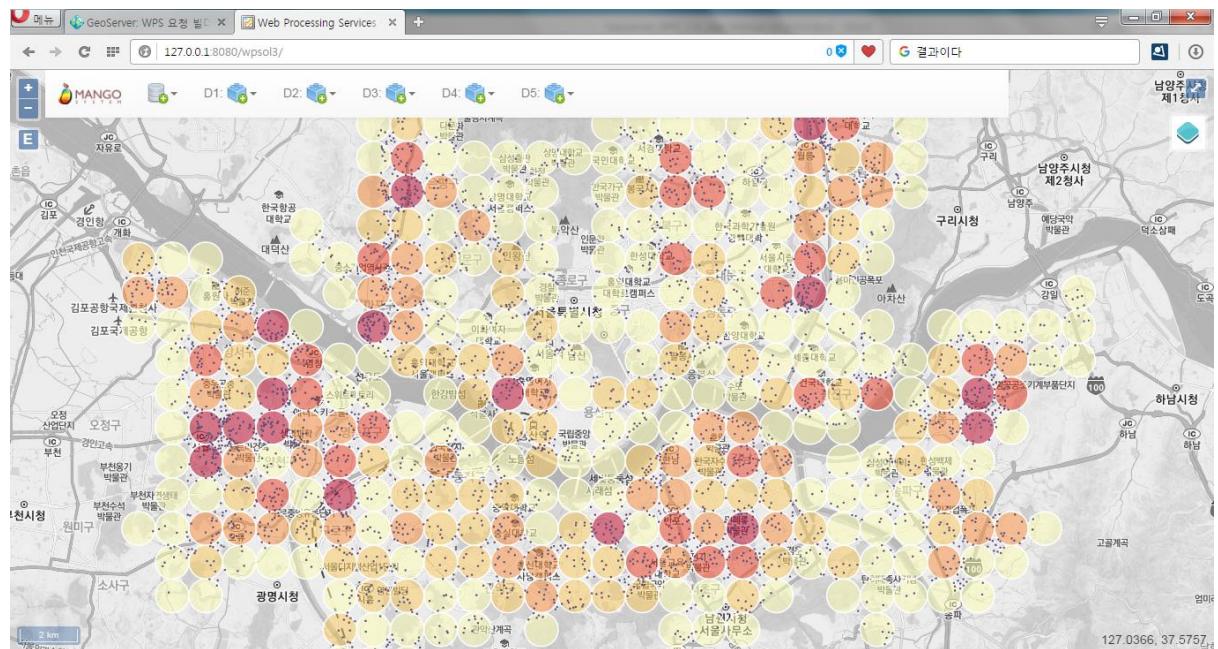
```

<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:CircularBinning</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>features</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
            xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:apartment"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>radius</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>750</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of creating and visualizing of circles with a radius of 750 meters created from apartment point data of Seoul.



4.2.7. Generalization

These processes are for generalization such as Dissolve, Simplification etc.

4.2.7.1. Dissolve

Performs Dissolve analysis using attribute fields of feature layers (**inputFeatures**) and aggregate functions.

■ Syntax

```
Dissolve (SimpleFeatureCollection inputFeatures, String dissolveField, String statisticsFields, Boolean useMultiPart): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features.	Complex	✓
dissolveField	The field on which to dissolve features.	Literal	✓
statisticsFields	The fields and statistics with which to summarize attributes. Statistics fields(Function.PropertyName): First, Last, Sum, Mean, Min, Max, Std, Count.	Literal	-
useMultiPart	Specifies whether multipart features are allowed in the output features.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If **useMultiPart** is set to False, converts and returns dissolved features in Single Part.
- **StatisticsFields** are input as [Function name, Field name] structure as follows, and the available functions are as follows. For example, Sum.pop, Mean.pop

Input

Return feild

First: String field, Dissolve object Feature's first value	FST_Field Name
Last: String field, Dissolve object Feature's last value	LST_Field Name
Sum: Numeric field, Dissolve object Feature's sum	SUM_Field Name
Mean: Numeric field, Dissolve object Feature's mean value	AVG_Field Name
Min: Numeric field, Dissolve object Feature's minimum value	MIN_Field Name
Max: Numeric field, Dissolve object Feature's maximum value	MAX_Field Name
Std: Numeric field, Dissolve object Feature's standard deviation	STD_Field Name
Var: Numeric field, Dissolve object Feature's variance	VAR_Field Name
Range: Numeric field, Dissolve object Feature's range	RNG_Field Name
Count: Dissolve object Feature's number	CNT_Field Name

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:Dissolve</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:emd"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>dissolveField</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>sgg_nm</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>statisticsFields</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>fst.sid_nm,sum.pop2007,sum.pop2008,sum.pts</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
</wps:Execute>
```

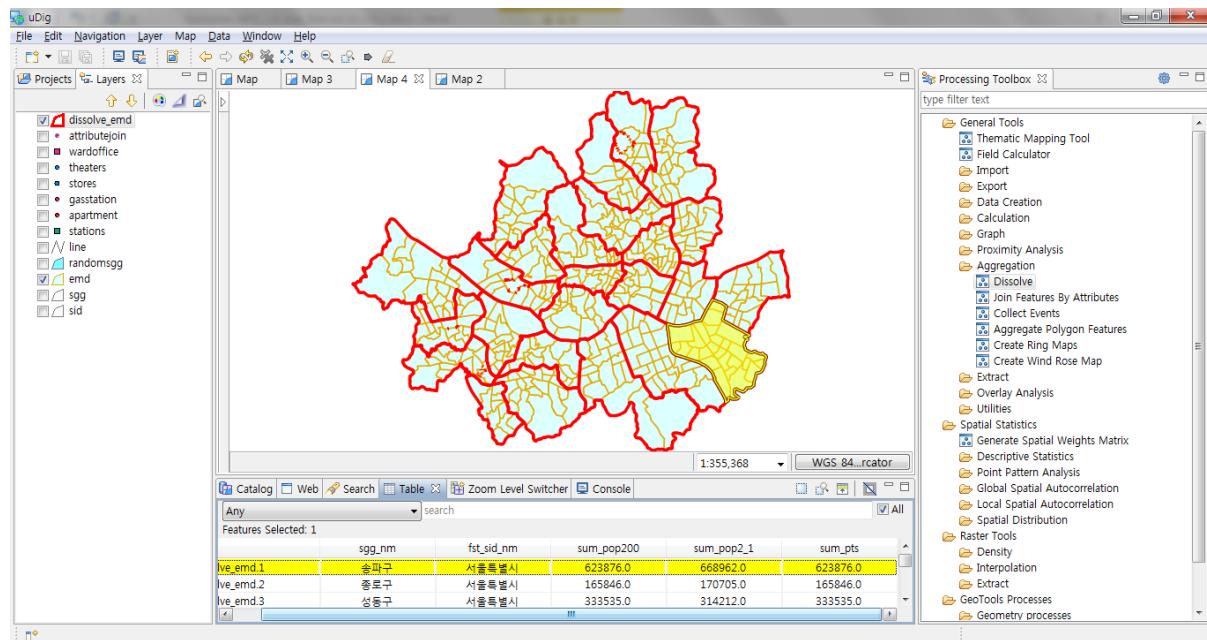
```

</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of setting Si-Gun-Gu name (sgg_nm) fields of the administrative boundary (Eup-Myeon-Dong) and aggregated fields of fst.sid_nm, sum.pop2007, sum.pop2008, sum.pts and then conducting Dissolve.



4.2.7.2. Remove Polygon Holes

Eliminates all holes (Interior Rings) of polygon feature layers (**inputFeatures**) or holes smaller than the set size (**minimumArea**).

■ Syntax

RemoveHoles (SimpleFeatureCollection inputFeatures, Expression minimumArea):
SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The polygon features to be removed.	Complex	✓
minimumArea	Remove holes smaller than this area expression. Ex) 10.0 or filter expression.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **MinimumArea** combines numbers or field values and expressions of returning in numbers are available.
- If **minimumArea** parameter value is set to 0, eliminates all holes (Interior Ring) of polygons, and if the value is greater than 0, eliminates Hole with area smaller than the set values.

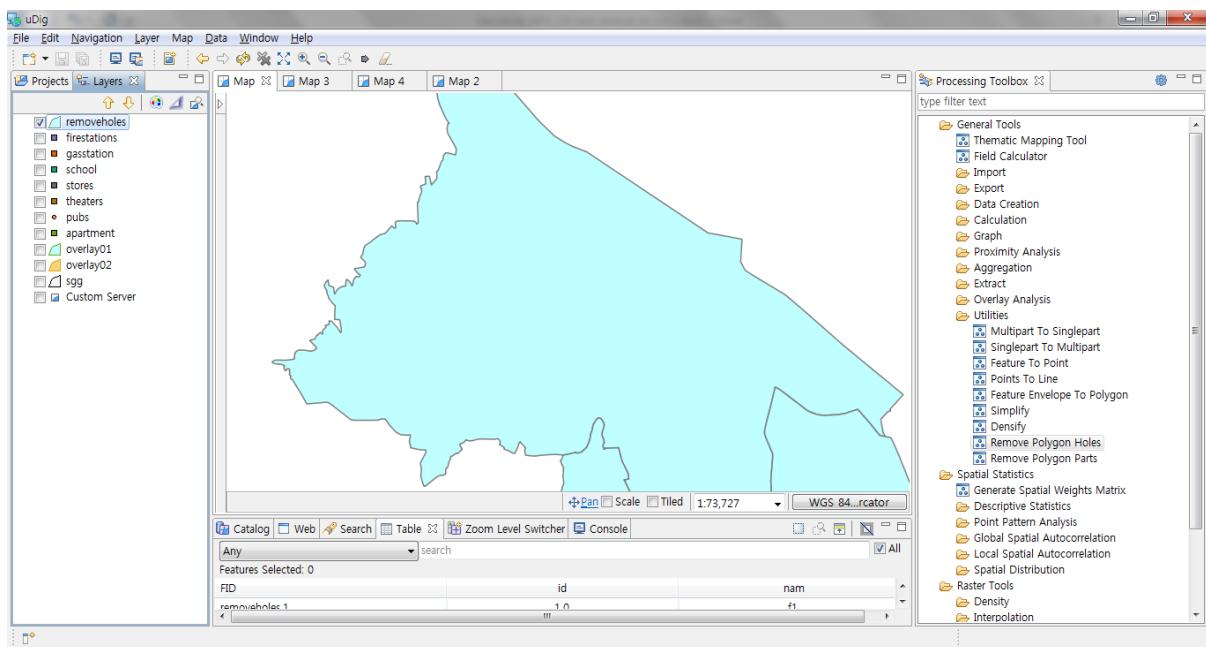
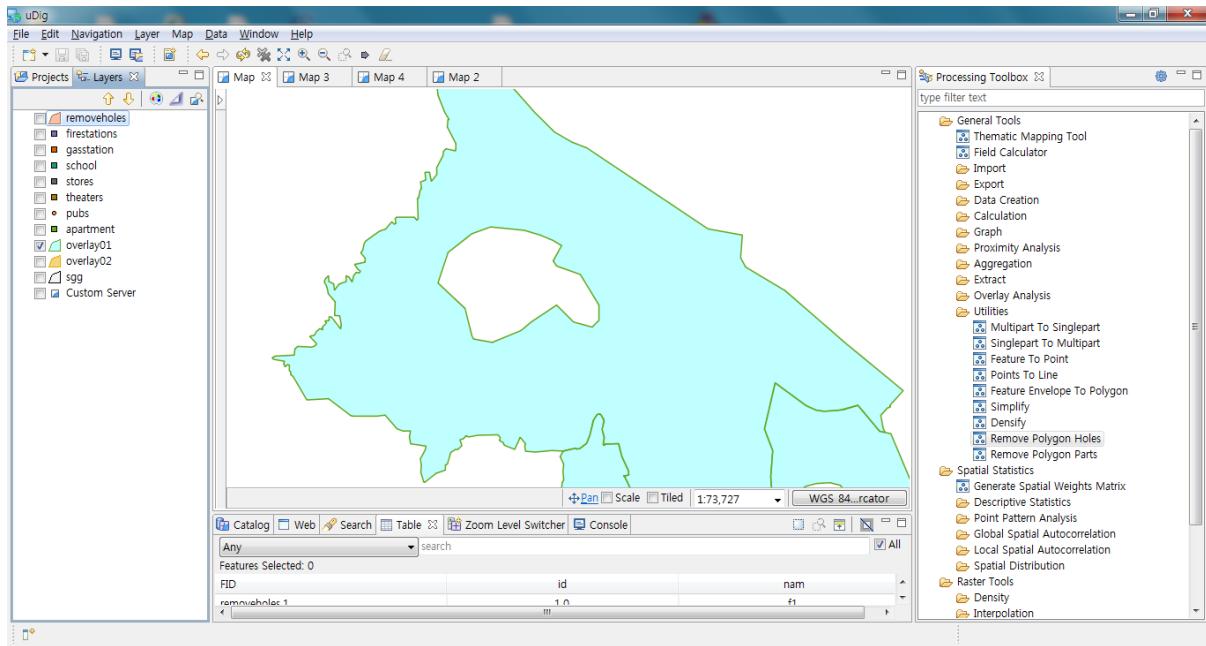
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
```

```
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:RemoveHoles</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:overlay01"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>minimumArea</ows:Identifier>
<wps:Data>
<wps:LiteralData>1000</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows an example of eliminating holes from polygons with Interior Ring (Hole).



4.2.7.3. Remove Polygon Part

Leaves only the parts smaller than the set size or the parts with the largest area with removing all others in the polygon feature layers consisting of Multipart Geometry.

■ Syntax

RemoveParts (SimpleFeatureCollection inputFeatures, Expression minimumArea):
SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The polygon features to be removed.	Complex	✓
minimumArea	Remove polygon parts smaller than this area expression. ex) 10.0 or filter expression.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **MinimumArea** combines numbers or field values and expressions of returning in numbers are available.
- If **minimumArea** parameter value is set to 0, eliminates all holes (Interior Ring) of polygons, and if the value is greater than 0, eliminates holes with area smaller than the set value.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
```

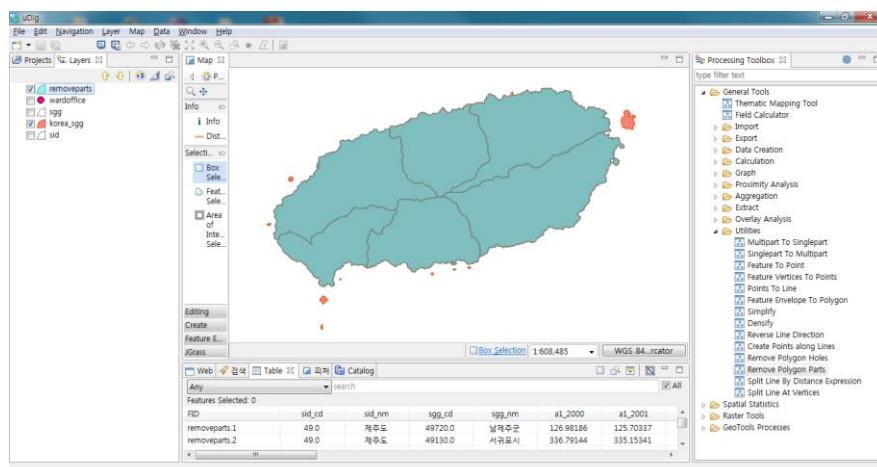
```

xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:RemoveParts</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:korea_sgg"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows an example of eliminating polygons less than the specified area in **MultiPolygon**. All of the islands with small areas are eliminated, as shown in figure.



4.2.7.4. Simplify

Simplifies polygon or line features using Douglas-Peucker simplifying algorithm.

■ Syntax

Simplify (SimpleFeatureCollection inputFeatures, Expression tolerance, Boolean preserveTopology): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input line or polygon features to be simplified.	Complex	✓
tolerance	Distance tolerance to simplify ex) 10.0 or filter expression.	Literal	✓
preserveTopology	If True, ensures that simplified features are topologically valid. Default is True	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **Tolerance** parameter can use numbers and functions, and Douglas-Peucker algorithm is used.
- If the **preserveTopology** parameter is set to True, maintains minimum topology rule regardless of the **Tolerance** value.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0"
```

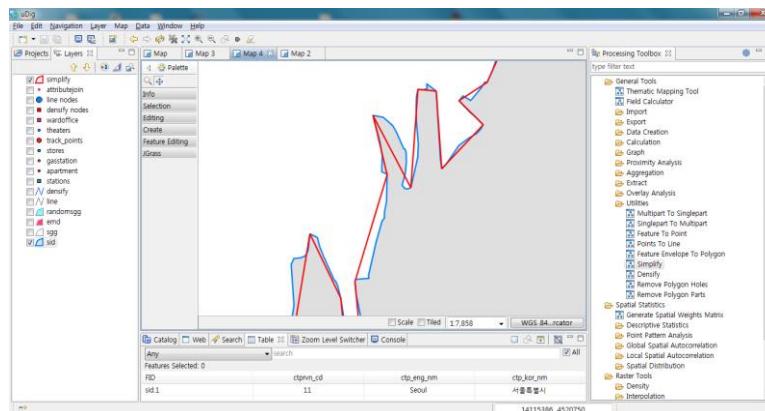
```

http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:Simplify</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:road"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>tolerance</ows:Identifier>
<wps:Data>
<wps:LiteralData>5</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The blue line is the original, and the red line is the result of Simplifying.



4.2.7.5. Densify

Adds vertices with set tolerance (**tolerance**) interval in every line segment of polygon or line features.

■ Syntax

Densify (SimpleFeatureCollection inputFeatures, Expression tolerance):

SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input line or polygon features to be calculated.	Complex	✓
tolerance	Distance tolerance to densify ex) 10.0 or filter expression.	Literal	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **Tolerance** parameter can use numbers and functions.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:Densify</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
```

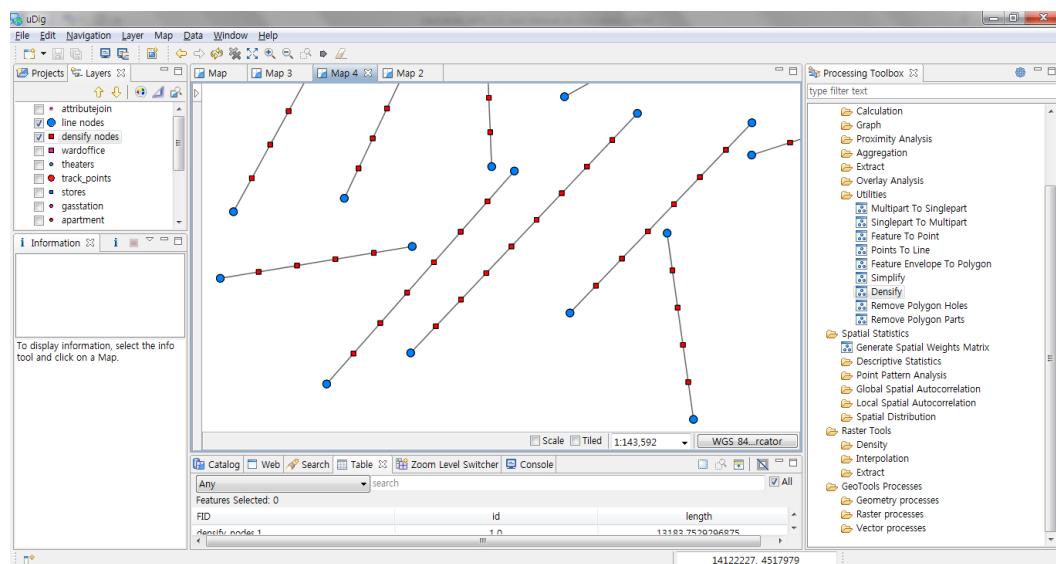
```

<wps:Body>
  <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
  xmlns:foss="http://www.opengeospatial.net/foss">
    <wfs:Query typeName="foss:line"/>
  </wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>tolerance</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>250</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of conducting Densify with a 250-meter interval in the original lines composed with start points and end points. The blue ones are the original vertices, and the red ones are the added vertices.



4.2.7.6. Eliminate

Removes the Sliver polygon based on the shared area or length of the neighbor polygons.

■ Syntax

Eliminate (SimpleFeatureCollection inputFeatures, EliminateOption option, Filter exception): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The layer whose polygons will be merged into neighboring polygons.	Complex	✓
option	The options specify which method will be used for eliminating features.	Literal	-
exception	The exception filter used to identify features that will not be altered.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **InputFeatures** layers must be polygon types.
- The **option** parameter incorporates the Sliver polygons with the largest shared length (Length), the largest area (SmallArea), or the smallest neighbor (SmallArea) based on the basis of eliminating Sliver polygons..
- **Exception** parameter sets features excluded from processing as filters.

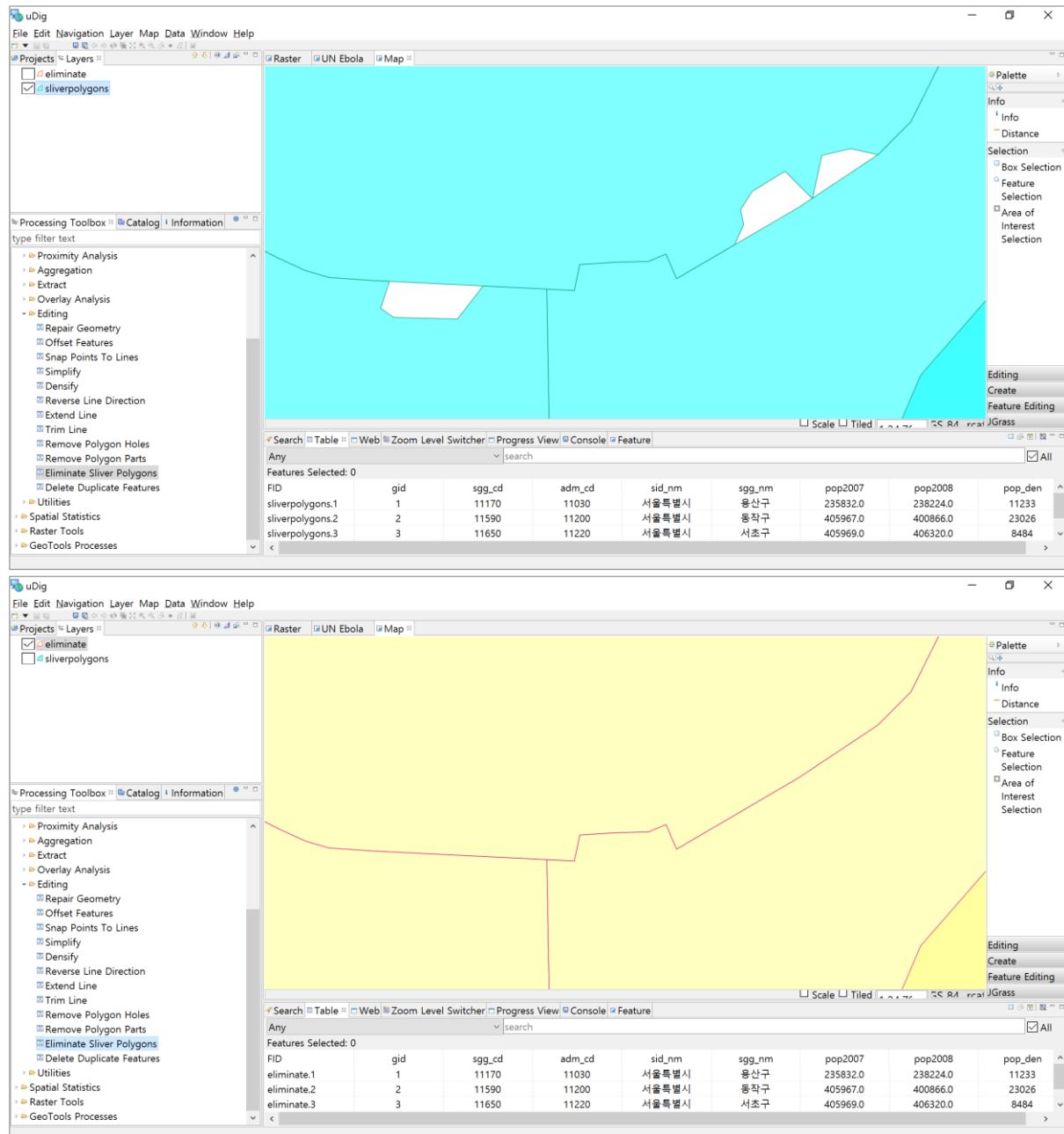
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
```

```
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:Eliminate</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:seoul_series"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>option</ows:Identifier>
<wps>Data>
<wps:LiteralData>Length</wps:LiteralData>
</wps>Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the original feature and result of eliminating Sliver polygon based on the shared length in the origin layer containing Sliver polygon.



4.2.8. Editing

These processes are for simplification such as Dissolve, Simplification etc.

4.2.8.1. Reverse Line Direction

- Changes the vertex order of line feature layers (**lineFeatures**).

■ Syntax

FlipLine (SimpleFeatureCollection lineFeatures): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
lineFeatures	The input line features.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **LineFeature** parameter must be line layer.

■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:FlipLine</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>lineFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
```

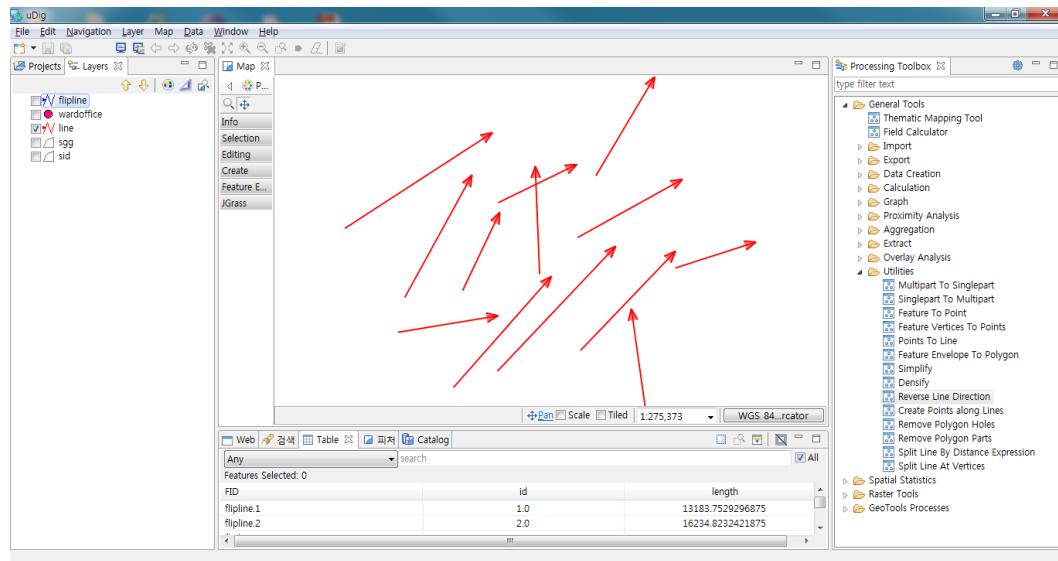
```

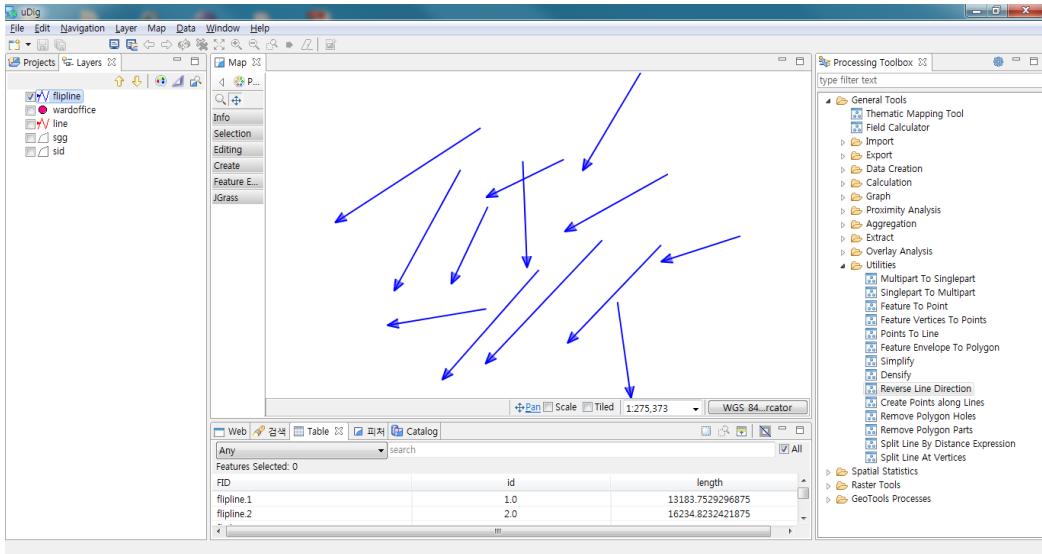
<wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:foss="http://www.opengeospatial.net/foss">
  <wfs:Query typeName="foss:line" />
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the original (red) line data and converted result (blue) data.





4.2.8.2. Offset Features

Moves all features of feature layers (**inputFeatures**) by x, y offsets (**offsetX**, **offsetY**).

■ Syntax

OffsetFeatures (SimpleFeatureCollection **inputFeatures**, Double **offsetX**, Double **offsetY**):

SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features that can be multipoint, line, and polygon.	Complex	✓
offsetX	X offset.	Literal	-
offsetY	Y offset.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

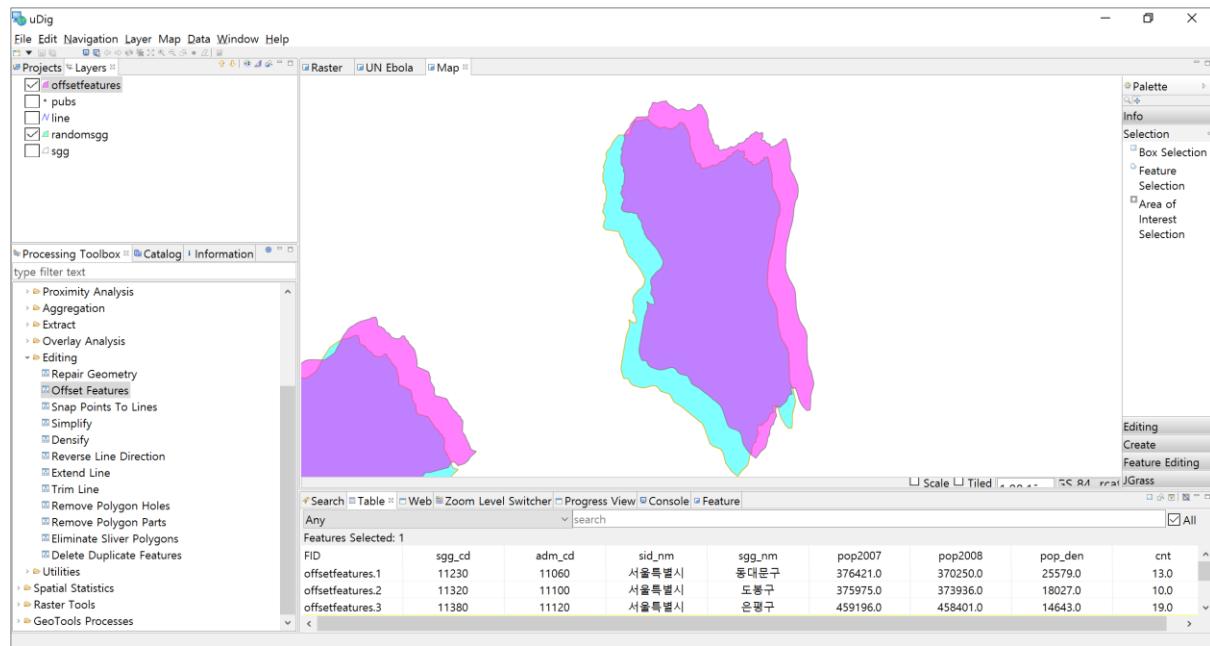
- Based on the original locations, moves to right if offsetX is positive, left if negative, up if offsetY value is positive, and down if negative.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:OffsetFeatures</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
        xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:randomsgg"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier> offsetX</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>500</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier> offsetY</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>500</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of moving the original layer by 500 meters on the X axis and 500 meters on the Y axis.



4.2.8.3. Snap Points To Lines

Moves to the nearest line or polygon boundaries (**lineFeatures**) based on the feature snap distance of the point layers (**pointFeatures**).

■ Syntax

SnapPointsToLines (SimpleFeatureCollection pointFeatures, SimpleFeatureCollection lineFeatures, Double tolerance): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
pointFeatures	Point features to be snapped.	Complex	✓
lineFeatures	Line features that can be Line or polygon boundary.	Complex	✓
tolerance	Snap tolerance. If tolerance is 0, nearest line feature will be used.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **LineFeatures** layers must be line or polygon types.
- If **tolerance** parameter is 0, moves to the nearest line features, and uses the distance unit of **pointFeatures**.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
```

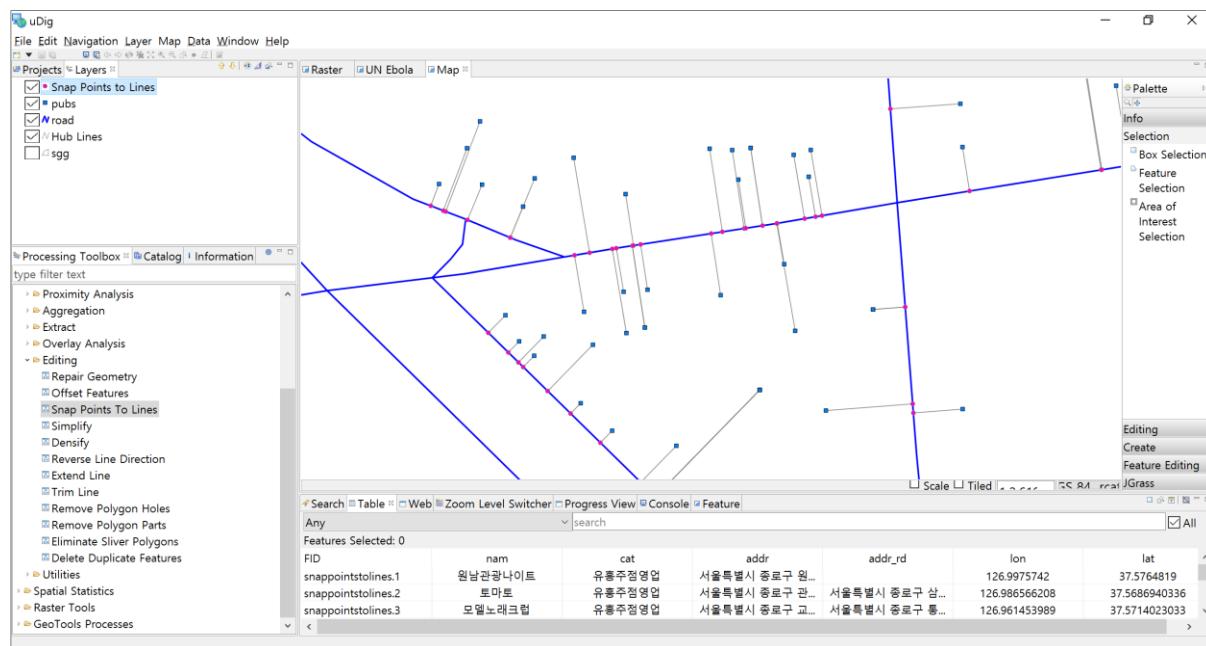
```

<ows:Identifier>statistics:SnapPointsToLines</ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>pointFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
        xmlns:foss="http://www.opengeospatial.net/foss">
          <wfs:Query typeName="foss:pubs"/>
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>lineFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
        xmlns:foss="http://www.opengeospatial.net/foss">
          <wfs:Query typeName="foss:road"/>
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of snapping points near roads to the nearest roads (line).



4.2.8.4. Extend Line

Extends the features of the line layers (**lineFeatures**) to the intersection of the first intersecting line within a specified distance (**length**).

■ Syntax

ExtendLine (SimpleFeatureCollection lineFeatures, Double length, Boolean extendTo):
SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
lineFeatures	The line features to be extended.	Complex	✓
length	The maximum distance a line segment can be extended to an intersecting feature.	Literal	✓
extendTo	Controls whether line segments can be extended to other extended line segments within the specified extend length.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **LineFeatures** layer must be line type.
- **Length** parameter is the maximum distance that a line segment can extend to, and uses the distance units of **lineFeatures**.
- If the **extendTo** parameter is set to True, extends all segments by length.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
```

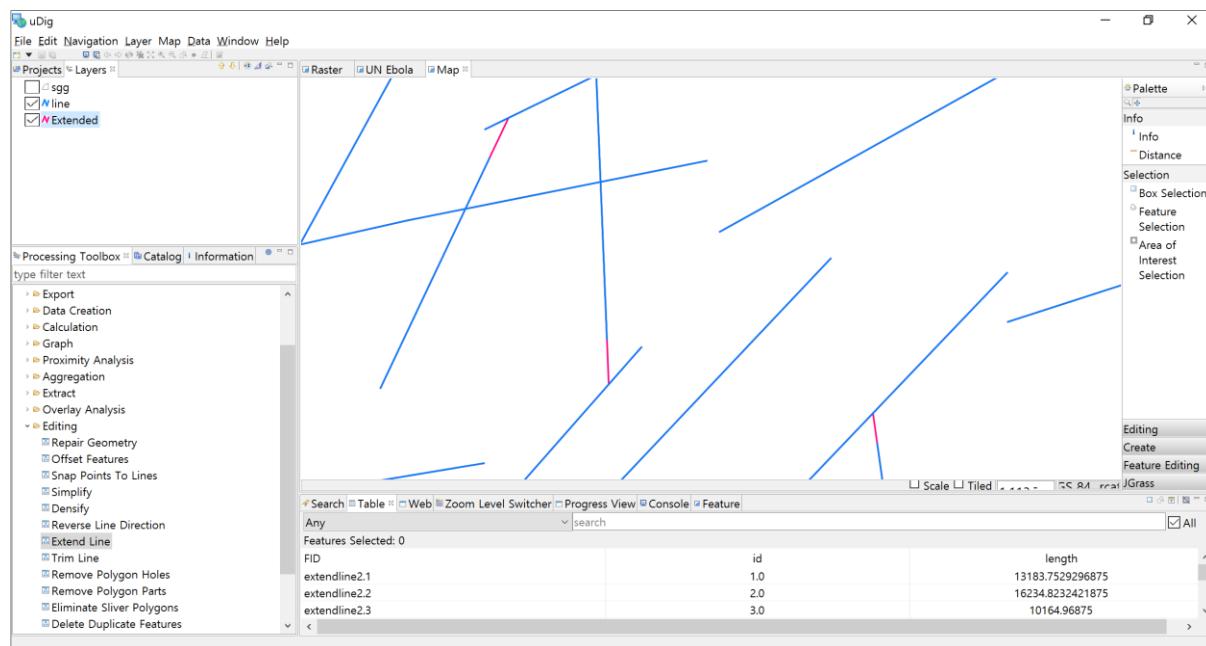
```

xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:ExtendLine</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>lineFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
            xmlns:foss="http://www.opengeospatial.net/foss">
              <wfs:Query typeName="foss:line"/>
            </wfs:GetFeature>
          </wps:Body>
        </wps:Reference>
      </wps:Input>
      <wps:Input>
        <ows:Identifier>length</ows:Identifier>
        <wps>Data>
          <wps:LiteralData>2000</wps:LiteralData>
        </wps>Data>
      </wps:Input>
      <wps:Input>
        <ows:Identifier>extendTo</ows:Identifier>
        <wps>Data>
          <wps:LiteralData>True</wps:LiteralData>
        </wps>Data>
      </wps:Input>
    </wps:DataInputs>
    <wps:ResponseForm>
      <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
        <ows:Identifier>result</ows:Identifier>
      </wps:RawDataOutput>
    </wps:ResponseForm>
  </wps:Execute>

```

■ Response

The following figures shows the result of extending line by specified distance.



4.2.8.5. Trim Line

Trims the features in the line layers (**lineFeatures**) that is shorter than a specified length (**dangleLength**) and do not intersect with other lines at start/end points.

■ Syntax

```
TrimLine (SimpleFeatureCollection lineFeatures, Double dangleLength, Boolean  
deleteShort): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
lineFeatures	The line features to be trimmed.	Complex	✓
dangleLength	Line segments that are shorter than the specified Dangle Length and do not touch another line at both endpoints (dangles) will be trimmed.	Literal	✓
deleteShort	Controls whether line segments which are less than the dangle length and are free-standing will be deleted. Default is True.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **LineFeatures** layer must be line type.
- If **deleteShort** parameter is set to True and the length of an independent line not intersecting with other lines at start/end points is less than the length of **dangleLength**, eliminates this line.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"  
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
```

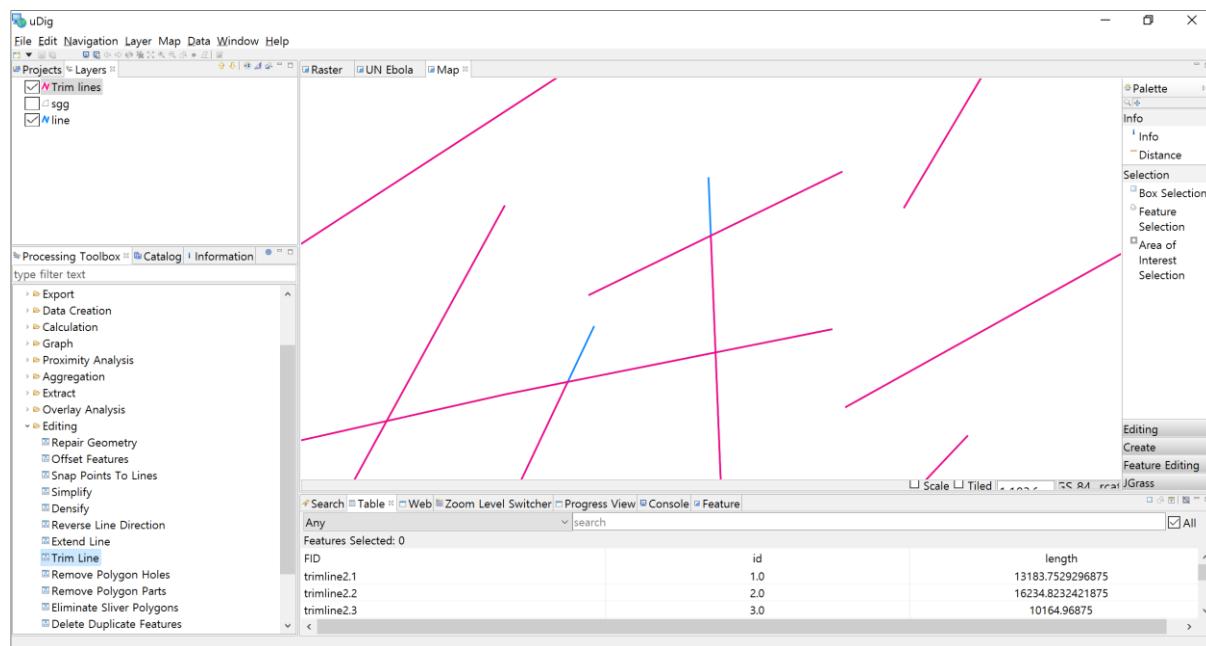
```

xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:TrimLine</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>lineFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
            <wfs:Query typeName="foss:line"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>dangleLength</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>1700</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>deleteShort</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>True</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of trimming segment smaller than specified length.



4.2.8.6. Delete Duplicated Features

Leaves only one feature with the same Geometry in the feature layers (**inputFeatures**) and deletes others.

■ Syntax

DeleteDuplicates (inputFeatures SimpleFeatureCollection): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input features to be processed.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **InputFeatures** can be points, lines and polygons. Only features with the same Geometry are seen as duplicated.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:DeleteDuplicates</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
```

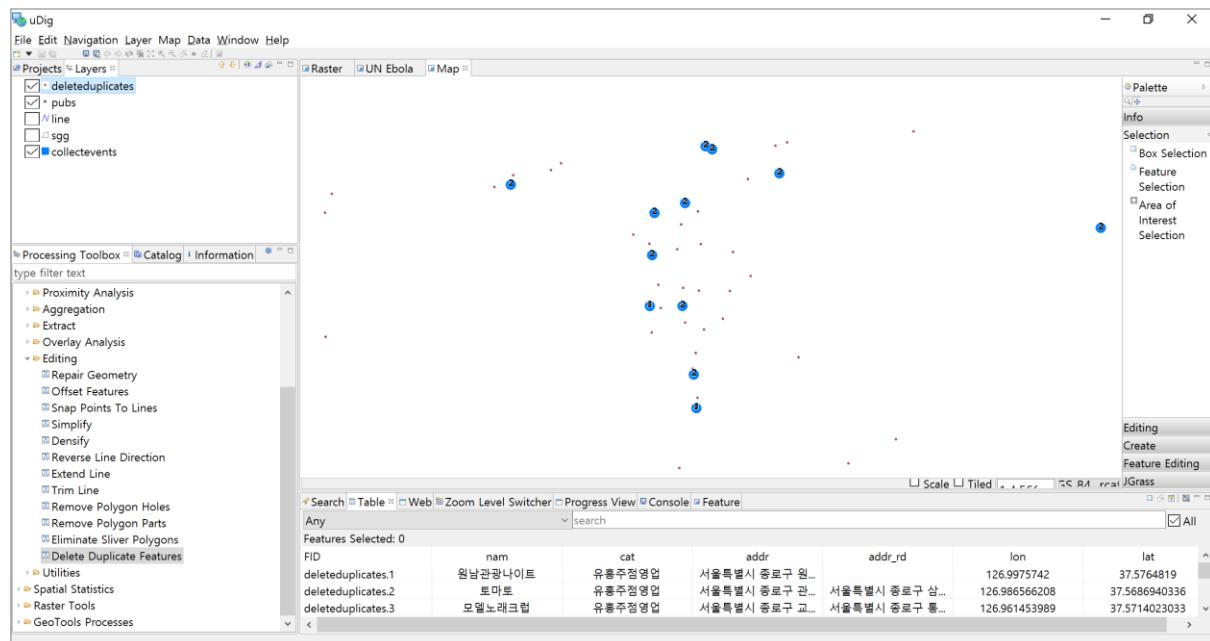
```

xmlns:foss="http://www.opengeospatial.net/foss"
    <wfs:Query typeName="foss:pubs"/>
    </wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
        <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

Among duplicated points, only one feature is stored, as shown as follows.



4.2.9. Feature Tools

These processes are for conversions such as Geometry type conversion, and format conversion.

4.2.9.1. Feature to Point

Converts feature layers (**inputFeatures**) into point feature layers such as center points and so on.

■ Syntax

```
FeatureToPoint (SimpleFeatureCollection inputFeatures, Boolean inside, Boolean singlePart): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input features that can be multipoint, line, polygon.	Complex	✓
inside	Centroid(False), Inside(True, Default)	Literal	-
singlePart	Centroid of each part. Default is False	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **InputFeatures** can be points, lines and polygons.
- If the **inputFeatures** is polygon and **inside** is set to True, the center point is contained within the polygon.
- If **singlePart** is set to True and geometry is MultiPart, converts geometry of all parts into center points.

■ Request Examples

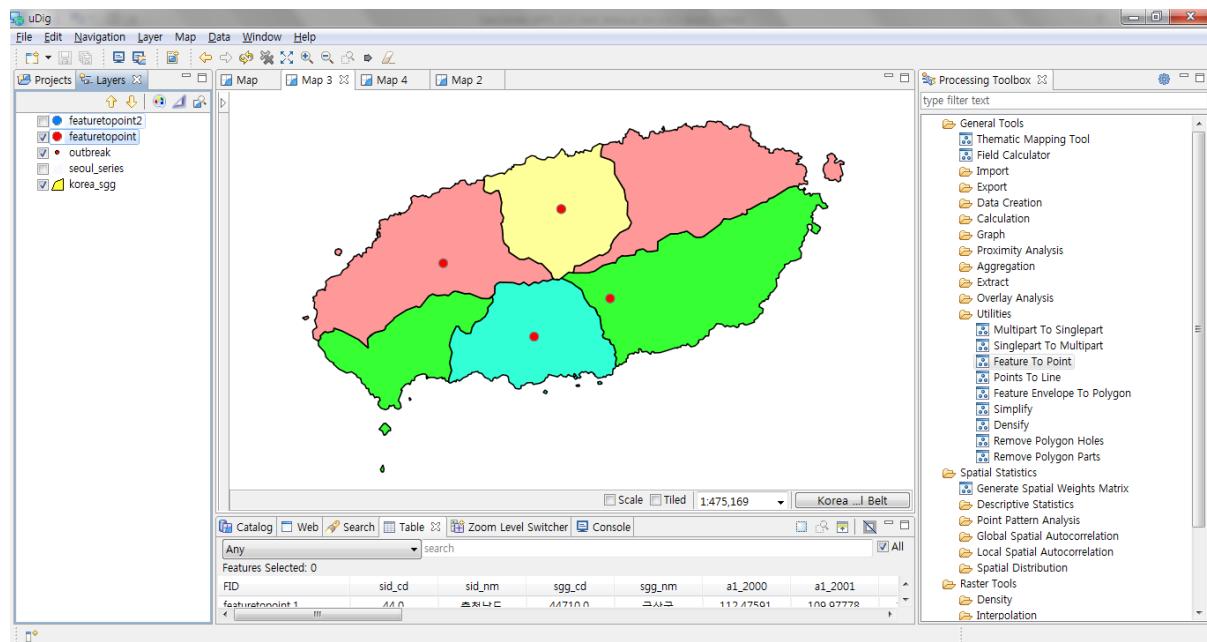
```

<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:FeatureToPoint</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:korea_sgg"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>inside</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>True</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>singlePart</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>False</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

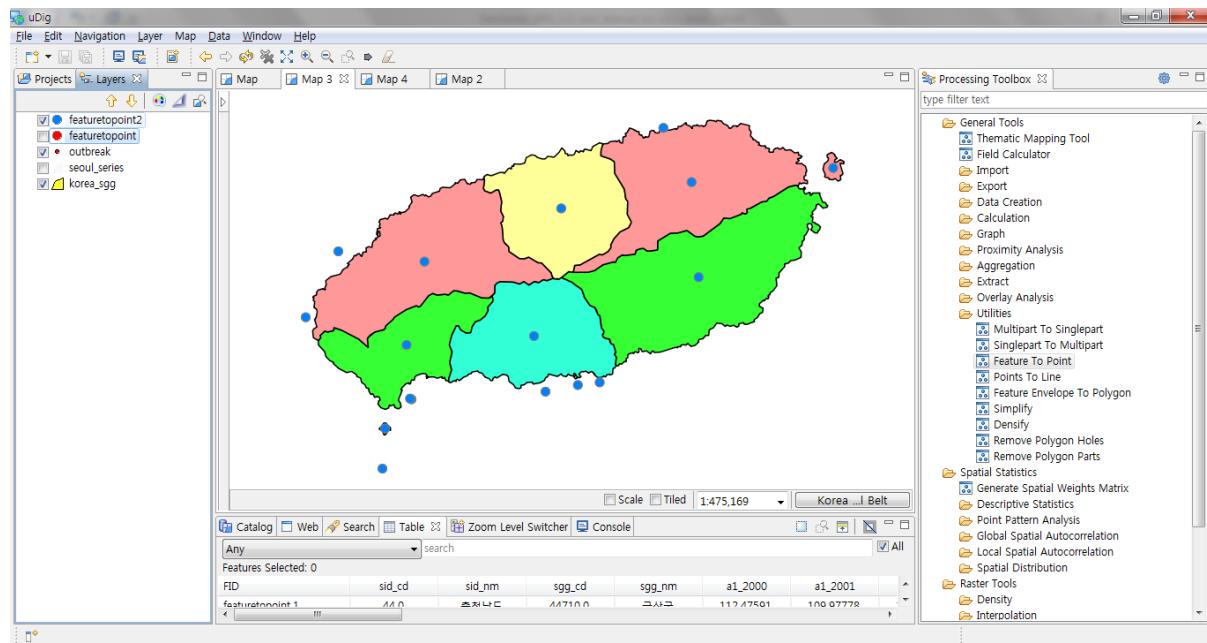
```

■ Response

The following figure shows the situation that the **inside** parameter is True and **singlePart** parameter is True. If it is MultiPolygon, it is converted into one center point.



This is the situation that **inside** parameter is set to True and **singlePart** parameter is set to False. If it is MultiPolygon, it is converted into center points by the number of polygons.



4.2.9.2. Multipart to Singlepart

Converts feature layers (**inputFeatures**) that configured with MultiPart into that with SinglePart.

■ Syntax

MultipartToSinglepart (SimpleFeatureCollection inputFeatures): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features that can be any feature type.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **InputFeatures** parameter must be Multipart (MultiPoint, MultiLineString, MultiPolygon) feature types.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:MultipartToSinglepart</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
```

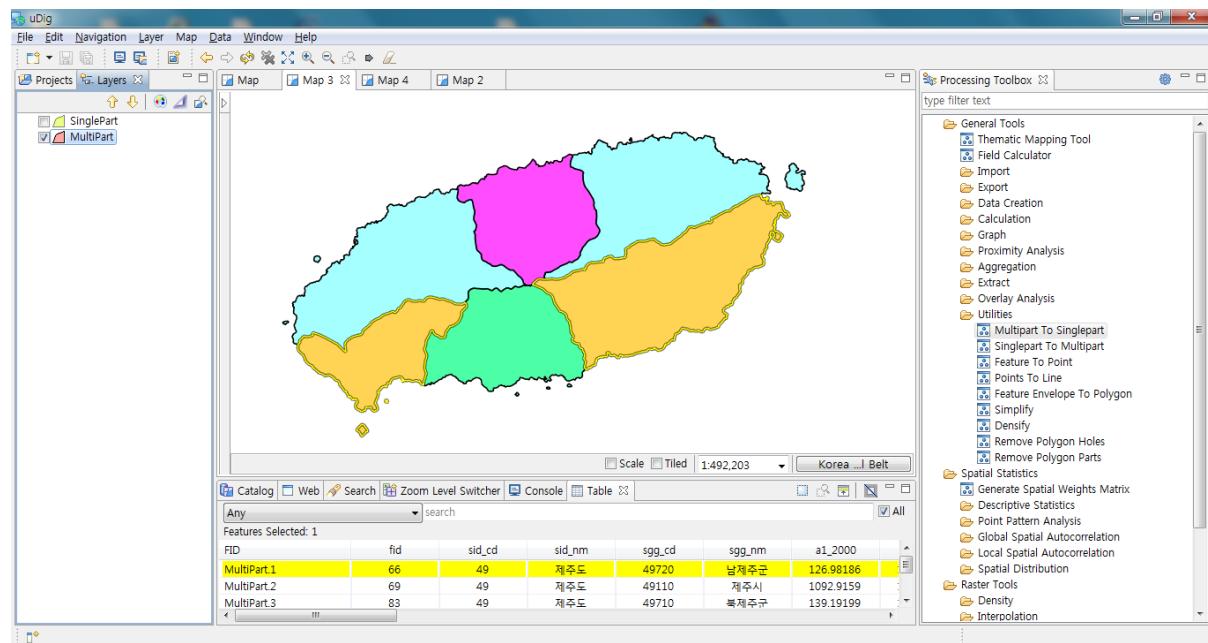
```

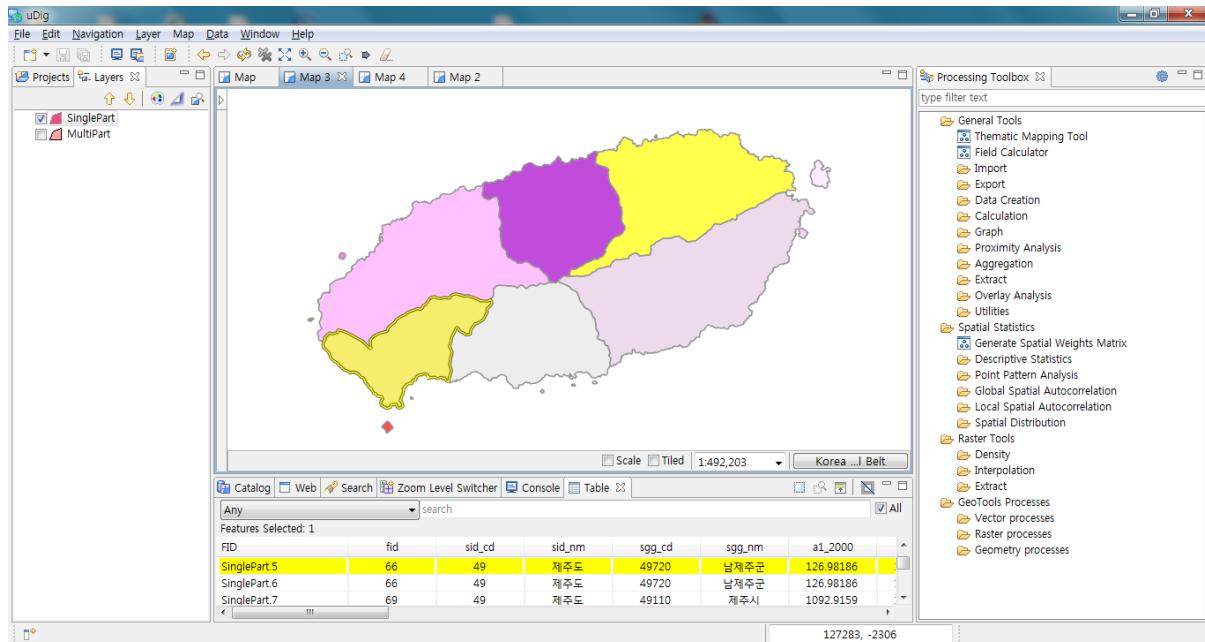
xmlns:foss="http://www.opengeospatial.net/foss"
    <wfs:Query typeName="foss:korea_sgg"/>
    </wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
        <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of converting MultiPolygon configured with more than 2 polygons into single polygon.





4.2.9.3. Singlepart to Multipart

Converts feature layers (**inputFeatures**) that configured with SinglePart into that with MultiPart based on attribute values.

■ Syntax

SinglepartToMultipart (SimpleFeatureCollection inputFeatures, String caseField, Boolean dissolve): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features that can be point, line, polygon.	Complex	✓
caseField	The field on which to aggregate features.	Literal	✓
dissolve	If true, neighborhood features are dissolved. Default is False	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If **dissolve** parameter is set to True, returns in geometry that dissolves the adjacent polygons or lines.

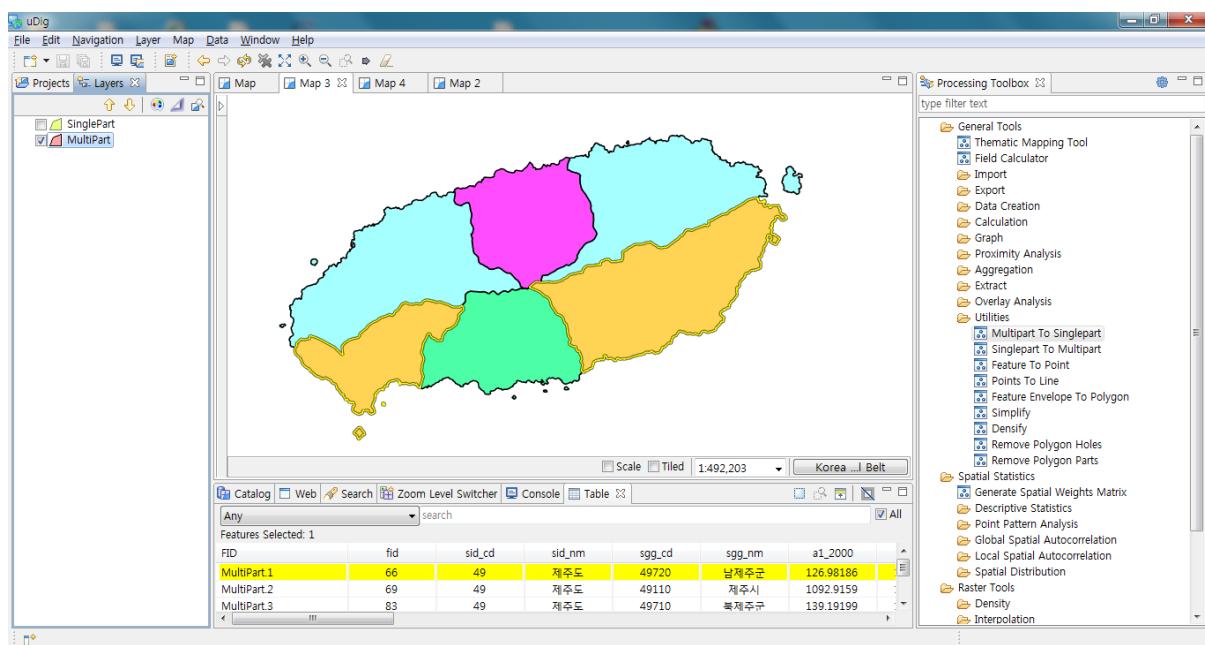
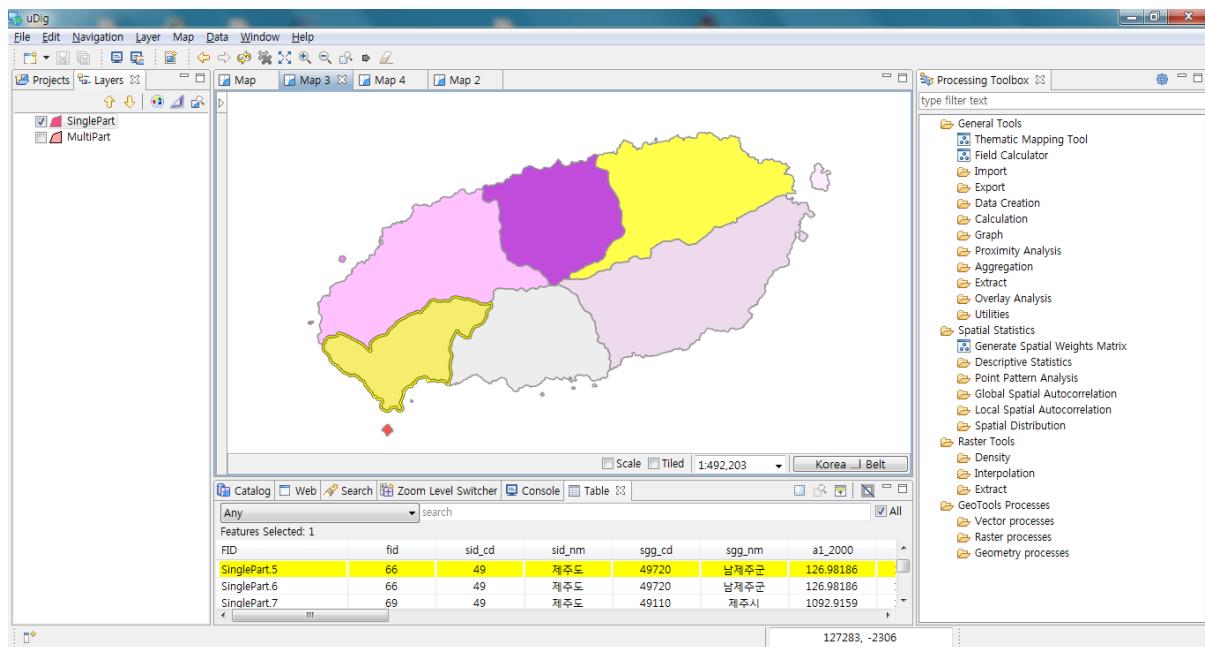
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:SinglepartToMultipart</ows:Identifier>
  <wps:DataInputs>
```

```
<wps:Input>
  <ows:Identifier>inputFeatures</ows:Identifier>
  <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
    <wps:Body>
      <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
      xmlns:foss="http://www.opengeospatial.net/foss">
        <wfs:Query typeName="foss:emd"/>
      </wfs:GetFeature>
    </wps:Body>
  </wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>caseField</ows:Identifier>
  <wps>Data>
    <wps:LiteralData>sgg_nm</wps:LiteralData>
  </wps>Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows an example of converting layer configured with single polygon back into MultiPolygon based on the name of the administrative doundary (Si-Gun-Gu).



4.2.9.4. Feature Envelope to Polygon

Converts the minimum boundary area for each feature in the feature layer to a polygon feature layer.

■ Syntax

FeatureEnvelopeToPolygon (SimpleFeatureCollection inputFeatures, Boolean singleEnvelope): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features that can be multipoint, line, polygon.	Complex	✓
singleEnvelope	Specifies whether to use one envelope for each entire multipart feature or one envelope per part of a multipart feature. Default is True	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If **singleEnvelope** is set to False and the geometry of each feature is MultiLineString or MultiPolygon, converts to Single Part and returns Envelope polygon for each geometry.

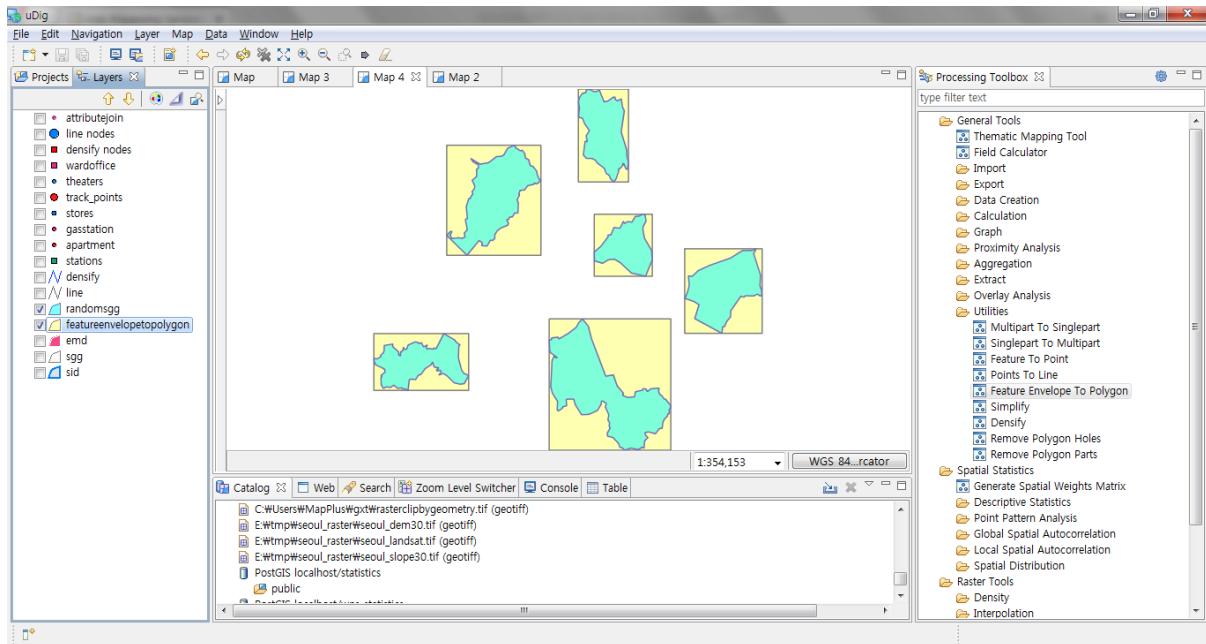
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:FeatureEnvelopeToPolygon</ows:Identifier>
```

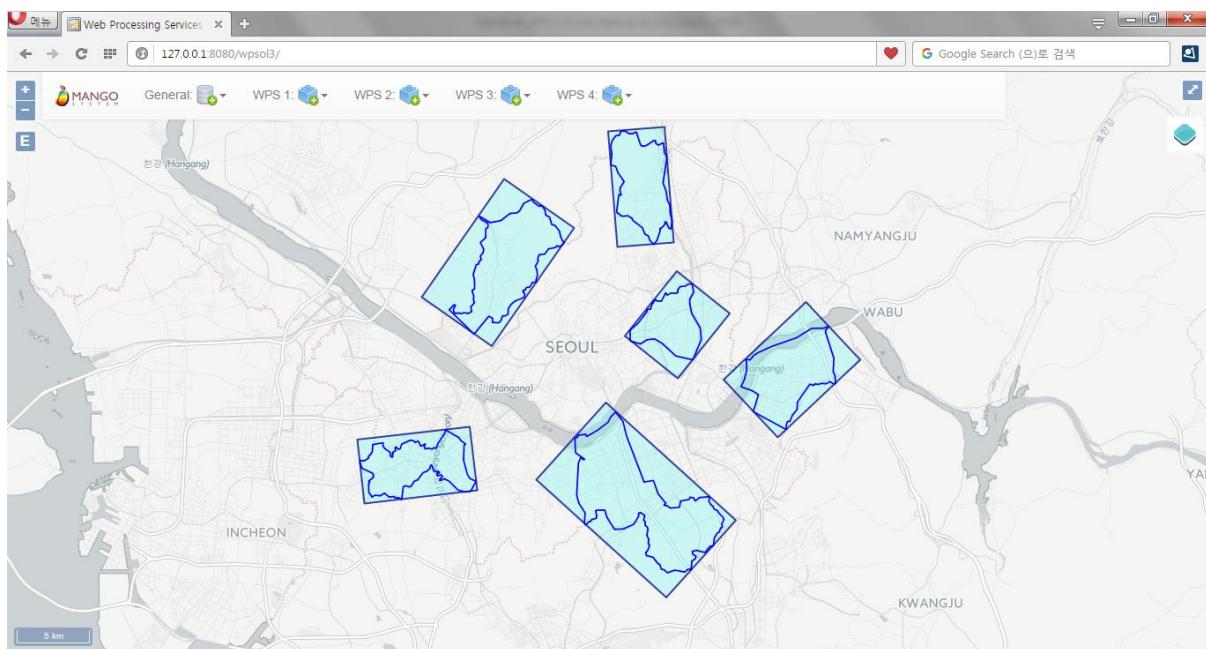
```
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlNs:foss="http://www.opengeospatial.net/foss">
          <wfs:Query typeName="foss:korea_sgg"/>
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>singleEnvelope</ows:Identifier>
    <wps:Data>
      <wps:LiteralData>False</wps:LiteralData>
    </wps:Data>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of converting Envelopes of polygon feature geometry to polygons.



Minimum bounding Envelope surrounding the polygon can be get using the [[Calculate Field](#)] function.



4.2.9.5. Points to Line

Sets line field (**lineField**) and aligned field value in the point feature layer (**inputField**) and converts into line or polygon layers.

■ Syntax

PointsToLine (SimpleFeatureCollection inputFeatures, String lineField, String sortField, Boolean closeLine): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The point features to be converted into lines.	Complex	✓
lineField	Each feature in the output will be based on unique values in the Line Field.	Literal	-
sortField	By default, points used to create each output line feature will be used in the order they are found. If a different order is desired, specify a Sort Field.	Literal	-
closeLine	Specifies whether output line features should be closed. Default is False.	Literal	

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If sets **lineField** parameter, lines are created separately according to the unique value of **lineField**.
- If sets **sortField** parameter, lines are created using points aligned to **sortField**.
- If the **closeLine** parameter is set to True, creates polygons by linking the start point with the end point.

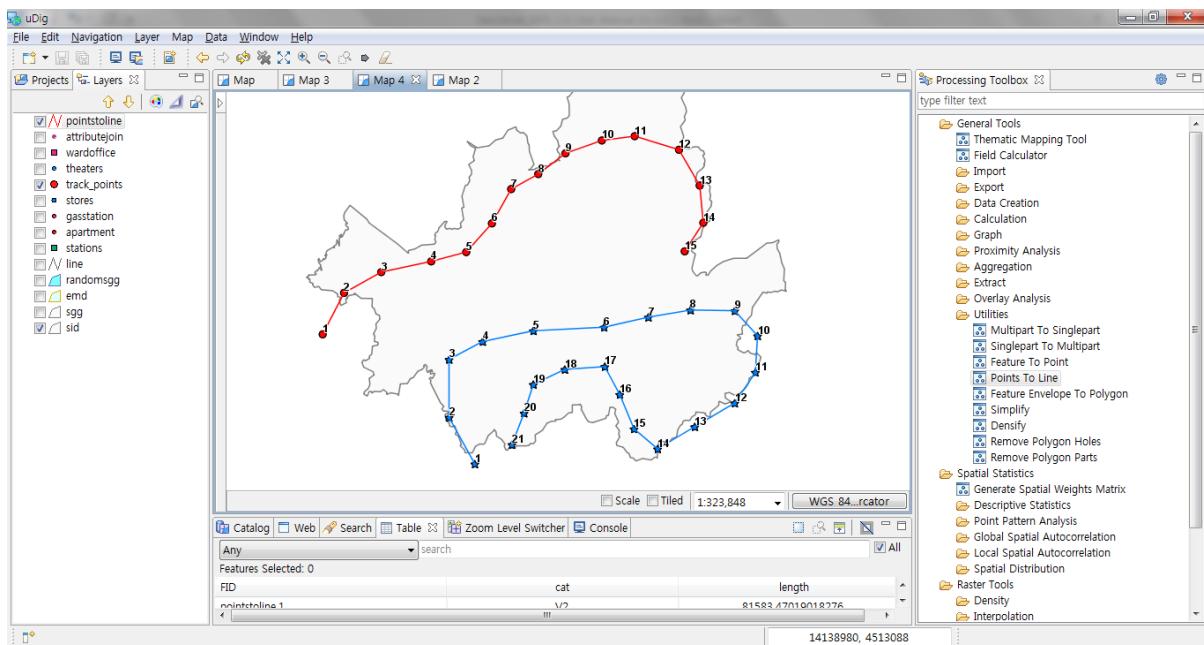
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:PointsToLine</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:track_points"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>lineField</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>cat</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>sortField</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>id</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>closeLine</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>False</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
```

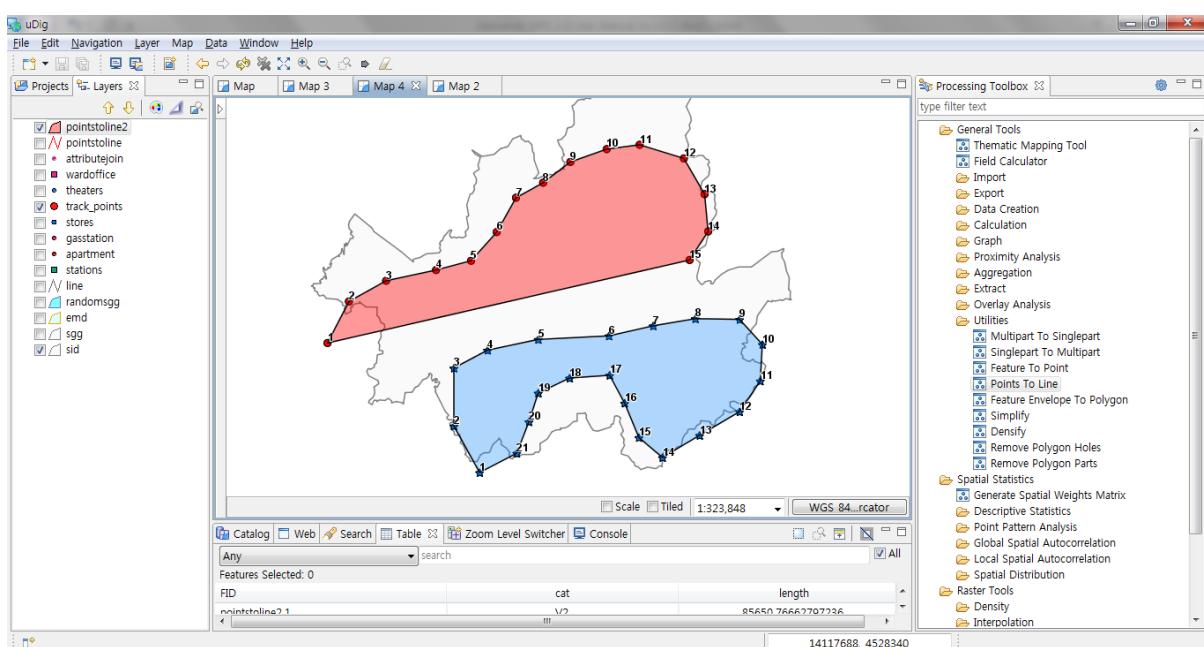
```
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of converting all points including the category and serial number to lines. The lines are created according to the serial number sequence and consist of two categories.



Polygon is created when **closeLine** is set to True in the example above.



4.2.9.6. Ring Maps

Creates a Ring Map by setting the property fields (**fields**) or the numbers of rings separated by a comma of the feature layers (**inputFeatures**).

■ Syntax

RingMap (SimpleFeatureCollection inputFeatures, String fields, String targetField, Integer ringGap): [SimpleFeatureCollection, SimpleFeatureCollection]

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features that can be point, line, and polygon.	Complex	✓
fields	Comma separated field or ring count.	Literal	✓
targetField	Output ring value field. ring_val is default.	Literal	-
ringGap	Gap of rings.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
anchor	Anchor features.	Complex	✓
ringmap	Ring map features.	Complex	✓

■ Constraints

- The **fields** parameter uses the number of consecutive fields or numbers of rings, such as a comma-separated yearly time series field.
- If **targetField** is set to Null, Ring_val field is the default value.
- **RingGap** parameter ranges from 1~9, and 1 is the default value.
- Output returns to two layers of ringmap polygons that created by the anchor line and the ring map displaying the leader lines.

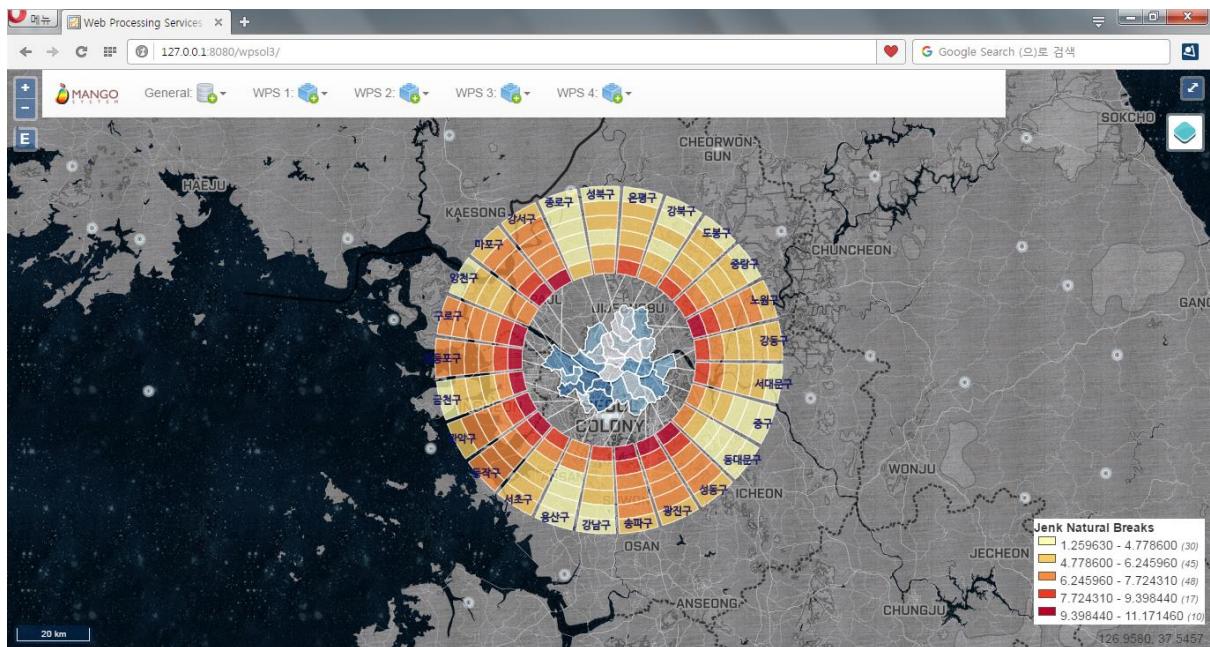
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RingMap</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
        xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:seoul_series"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>fields</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>a3_2000,a3_2001,a3_2002,a3_2003,a3_2004,a3_2005</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>targetField</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>ring_val</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:ResponseDocument>
      <wps:Output mimeType="text/xml; subtype=wfs-collection/1.0">
        <ows:Identifier>anchor</ows:Identifier>
      </wps:Output>
      <wps:Output mimeType="text/xml; subtype=wfs-collection/1.0">
        <ows:Identifier>ringmap</ows:Identifier>
      </wps:Output>
    </wps:ResponseDocument>
  </wps:ResponseForm>
</wps:Execute>
```

```
</wps:Execute>
```

■ Response

The following figure shows the result of creating the Ring Map using the natural population growth rate from 2000 to 2005 in Seoul.



4.2.9.7. Wind Rose Maps

Creates a Wind Rose Map by setting the point feature layer and center point.

■ Syntax

WindRoseMap (SimpleFeatureCollection inputFeatures, String weightField, Geometry center): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features that can be point, line, and polygon.	Complex	✓
weightField	Weight field.	Literal	-
center	Center (geometry) of wind rose.	Complex	-

■ Process Outputs

Identifier	Description	Type	Required
anchor	anchor features.	Complex	✓
windRose	Wind rose features.	Complex	✓

■ Constraints

- The **weightField** parameter must be a Numeric field. If sets the parameter, the sum of these field values is reflected in the result, otherwise the number of features is reflected.
- Output returns the anchor line and the windrose polygon layers to display the leader lines.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
```

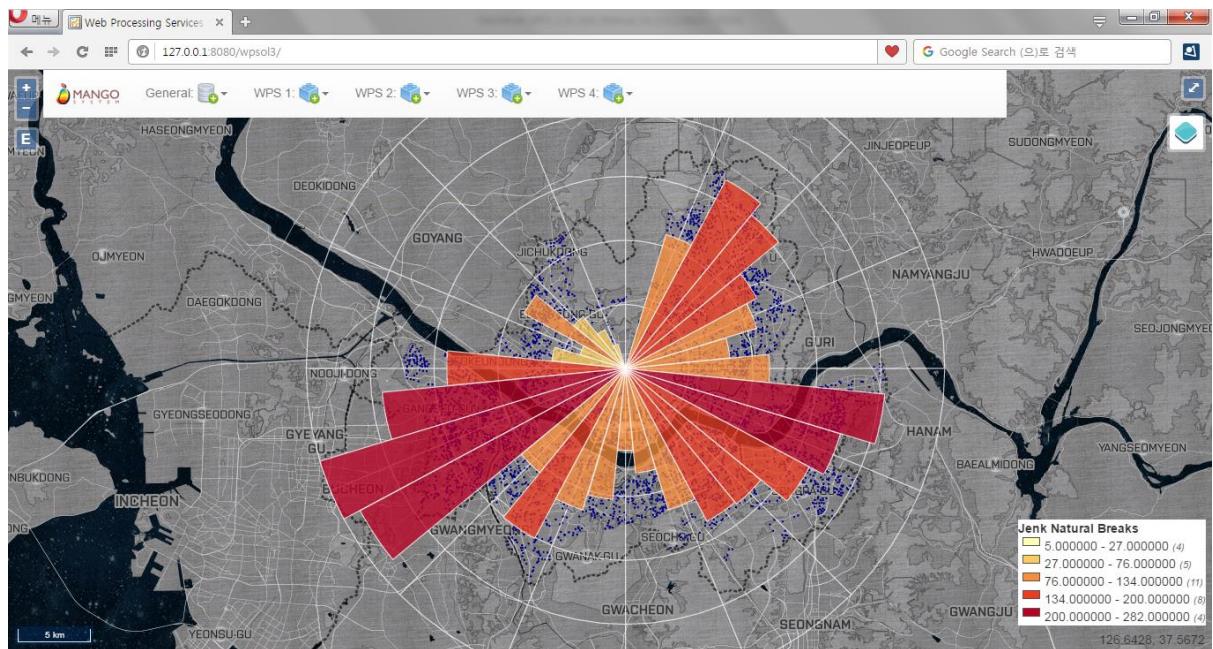
```

xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:WindRoseMap </ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures </ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:apartment"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>center </ows:Identifier>
<wps:Data>
<wps:ComplexData mimeType="application/wkt"><![CDATA[POINT(14135161.941
4518394.452)]]></wps:ComplexData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:ResponseDocument>
<wps:Output mimeType="text/xml; subtype=wfs-collection/1.0">
<ows:Identifier>anchor </ows:Identifier>
</wps:Output>
<wps:Output mimeType="text/xml; subtype=wfs-collection/1.0">
<ows:Identifier>windRose </ows:Identifier>
</wps:Output>
</wps:ResponseDocument>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of creating Wind Rose Map using distribution of apartments with Seoul City Hall as the center.



4.2.9.8. Hub Lines by ID

Creates a shortest distance of Hub line feature layers (**hubFeatures**) using the Join field of the Hub feature layers (**hubIdField**) and the Spoke feature layers (**spokeIdField**).

■ Syntax

```
HubLinesByID (SimpleFeatureCollection hubFeatures, String hubIdField,  
SimpleFeatureCollection spokeFeatures, String spokeIdField, Boolean preserveAttributes,  
Boolean useCentroid, Boolean useBezierCurve, Double maximumDistance);  
SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
hubFeatures	Hub Features.	Complex	✓
hubIdField	Hub id field.	Literal	✓
spokeFeatures	Spoke Features.	Complex	✓
spokeIdField	Spoke id field.	Literal	✓
preserveAttributes	Preserve spoke feature's attributes. Default is True	Literal	-
useCentroid	Use centroid of feature. Default is True	Literal	-
useBezierCurve	Use Bezier Curve. Default is False.	Literal	-
maximumDistance	Maximum distance.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If the **useCentroid** parameter is set to True and the geometry feature types of **hubFeatures**, **spokeFeatures** are line or polygon layers, creates the hub line using centroid of geometry.
- If the **useBezierCurve** parameter is set to True, creates the Bezier curve with the shortest line between the two features.

- If the **maximumDistance** parameter is greater than 0, creates a hub line only for features within this distance

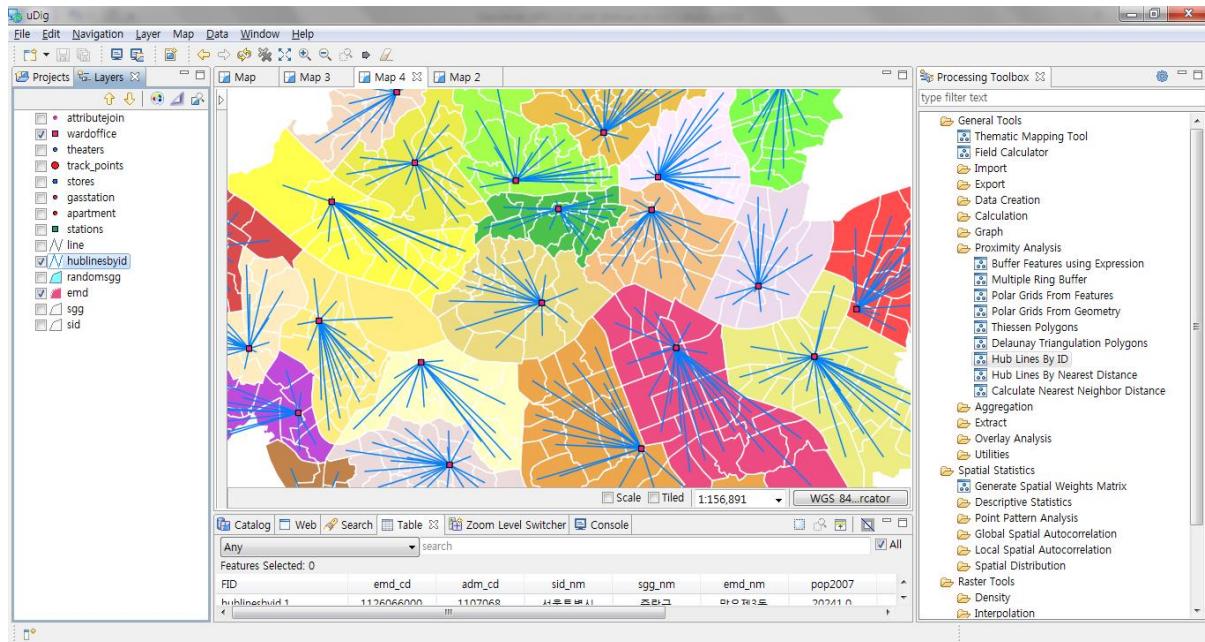
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:HubLinesByID</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>hubFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
        xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:wardoffice"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>hubIdField</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>sgg_nm</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>spokeFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
        xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:emd"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
```

```
<wps:Input>
  <ows:Identifier>spokIdField</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>sgg_nm</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>preserveAttributes</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>True</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>useCentroid</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>True</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of setting the city municipal offices as the hubs, setting the Eup-Myeon-Dong district polygon as a spoke layer, and creating the hub lines based on the Si-Gun-Gu administrative district code.



4.2.9.9. Hub Lines by Nearest Distance

Creates a hub line feature layer between the closest hub features (**hubFeatures**) in the spoke features (**spokeFeatures**), using the hub feature layers and the spoke feature layers.

■ Syntax

```
HubLinesByDistance (SimpleFeatureCollection spokeFeatures, SimpleFeatureCollection hubFeatures, String hubIdField, Boolean preserveAttributes, Boolean useCentroid, Boolean useBezierCurve, Double maximumDistance): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
spokeFeatures	Spoke Features.	Complex	✓
hubFeatures	Hub Features.	Literal	✓
hubIdField	Hub id field.	Literal	-
preserveAttributes	Preserve spoke feature's attributes. Default is True	Literal	-
useCentroid	Use centroid of feature. Default is True	Literal	-
useBezierCurve	Use Bezier Curve. Default is False.	Literal	-
maximumDistance	Maximum distance.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- If the **useCentroid** parameter is set to True and the geometry feature types of **hubFeatures**, **spokeFeatures** are lines or polygons, creates the hub lines using centroid of geometry.
- If the **useBezierCurve** parameter is set to True, creates the Bezier curve with the shortest line between the two features.

- If the **maximumDistance** parameter is greater than 0, creates a hub line only for features within set distance

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:HubLinesByDistance</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>spokeFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:wardoffice"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>hubFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:apartment"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>preserveAttributes</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>True</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
```

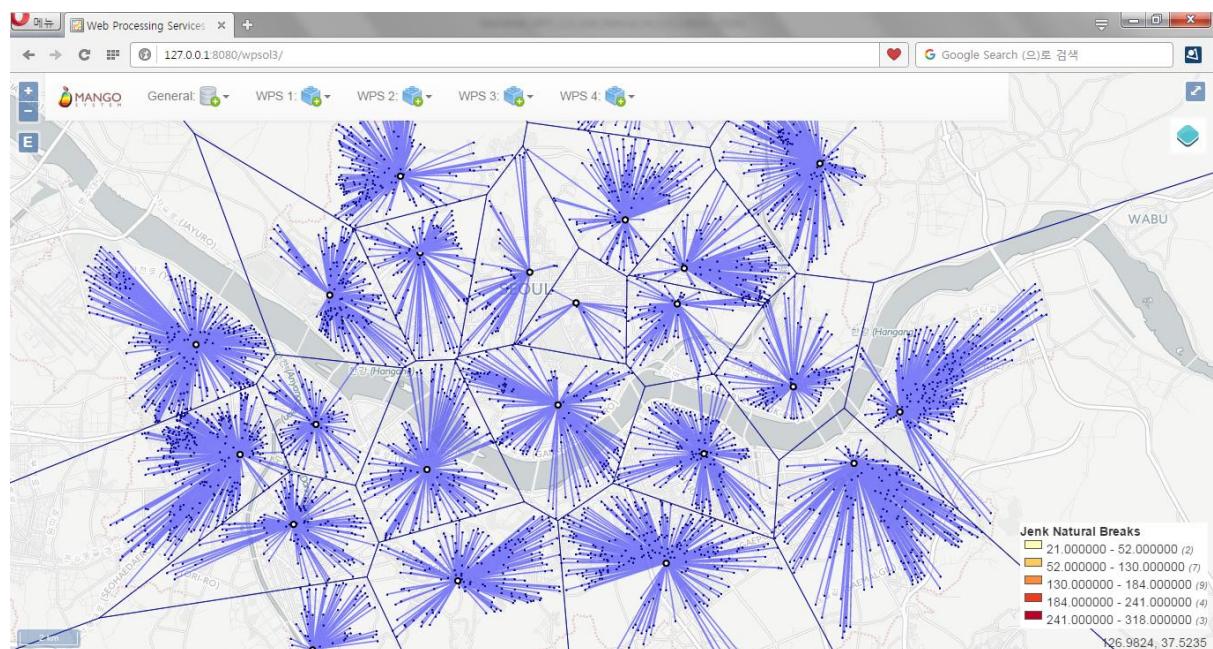
```

<ows:Identifier>useCentroid</ows:Identifier>
<wps:Data>
  <wps:LiteralData>True</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

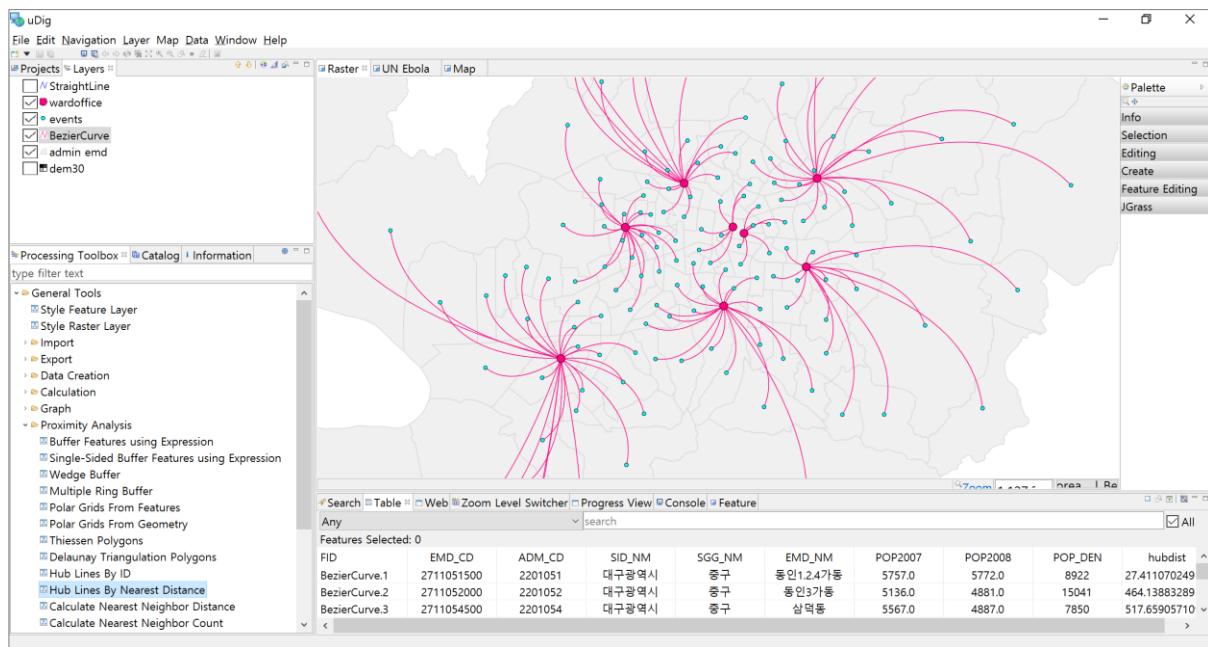
```

■ Response

The following figure shows the result of connecting the nearest apartments of the municipal offices by setting the municipal offices as the hubs and the apartment distribution as the spoke. It will be placed in the same area as Thiessen Polygon as shown in the following figure.



If uses the Bezier curve option, creates line layers as curves as follows.



4.2.9.10. Feature To Line

Creates line layers divided by the nodes that intersect with the polygon or line layers.

■ Syntax

FeatureToLine (SimpleFeatureCollection inputFeatures, Boolean preserveAttributes):

SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input features that can be line or polygon.	Complex	✓
preserveAttributes	Specifies whether to preserve or omit the input attributes in the output features. Default is True.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- The **inputFeature** layers must be line or polygon types.
- Default value of the **preserveAttributes** parameter is True, and if it is set to true, retains the property values of the original features.

■ Request Examples

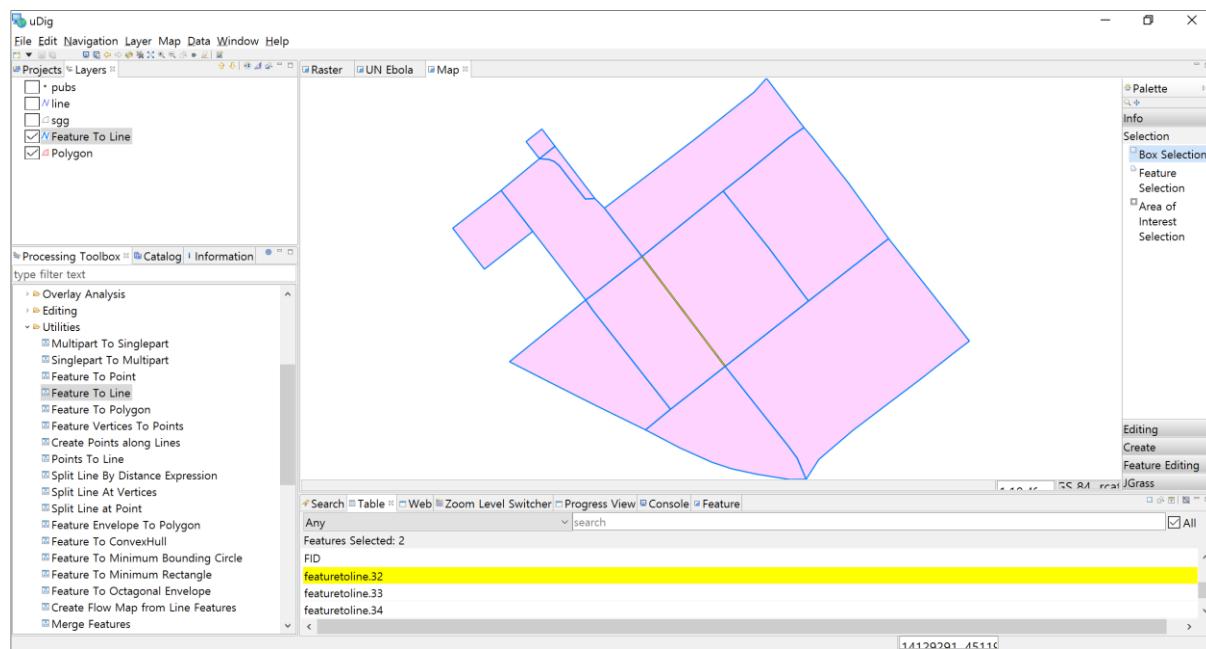
```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:FeatureToLine</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
```

```
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
  <wps:Body>
    <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
      xmlns:foss="http://www.opengeospatial.net/foss">
      <wfs:Query typeName="foss:ssg"/>
    </wfs:GetFeature>
  </wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>preserveAttributes</ows:Identifier>
  <wps>Data>
    <wps:LiteralData>True</wps:LiteralData>
  </wps>Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of converting the polygon layers to line layers.

The converted lines are divided into segments that share polygons as shown in the following figure.



4.2.9.11. Feature To Polygon

Creates polygon layers using polygon or line layers (**inputFeatures**).

■ Syntax

FeatureToPolygon (SimpleFeatureCollection inputFeatures, Double tolerance, SimpleFeatureCollection labelFeatures): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input features that can be line or polygon.	Complex	✓
tolerance	Tolerance. The default is 0.001 feature unit.	Literal	-
labelFeatures	The optional input point features that hold the attributes to be transferred to the output polygon features.	Complex	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- The **inputFeatures** layers must be line or polygon layer types.
- Default value of the **tolerance** parameter is 0.001, and the units of the inputFeatures coordinate system.
- The **labelFeatures** layers must be point layer types.
- If the **labelFeatures** parameter is not Null, uses this schema and assign the attribute values of the **labelFeatures** points contained in the polygon after polygon generation.

■ Request Examples

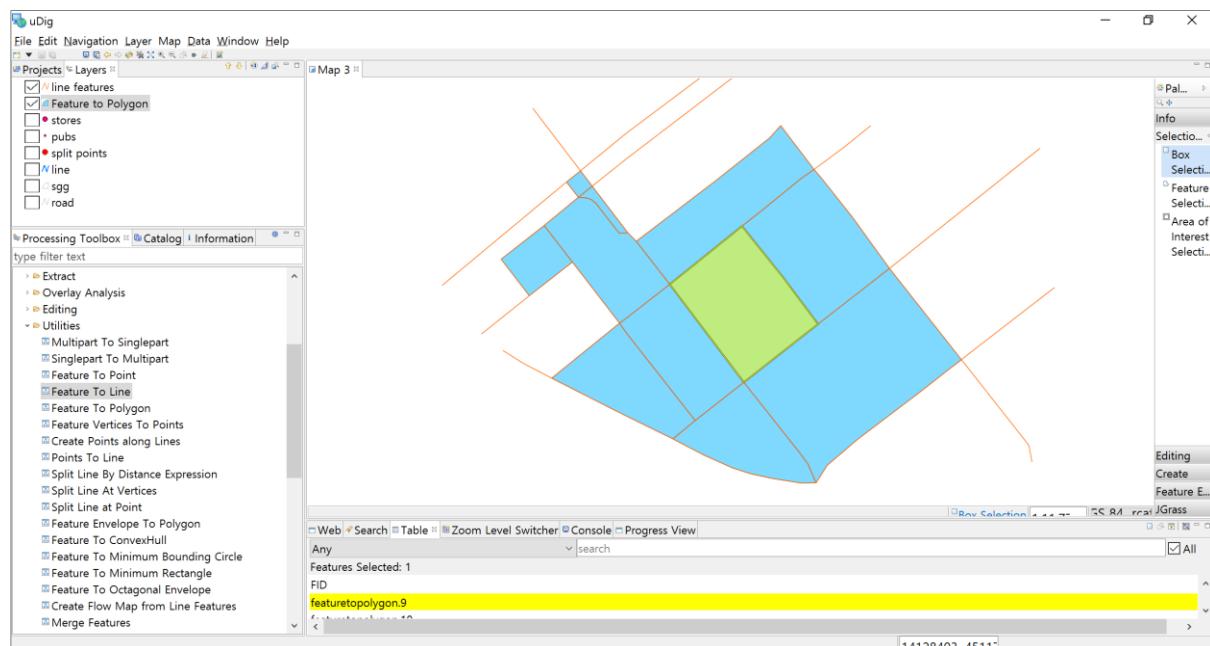
```

<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:FeatureToPolygon</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:line"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>tolerance</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>0.001</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of converting the line layers to polygon layers.



4.2.9.12. Feature Vertices to Points

Creates point feature layers based on the set vertex position (**location**) using polygon or line data (**inputFeatures**).

■ **Syntax**

VerticesToPoints (SimpleFeatureCollection inputFeatures, PointLocationType location):
SimpleFeatureCollection

■ **Parameters**

■ **Data Inputs**

Identifier	Description	Type	Required
inputFeatures	The input features that can be line or polygon.	Complex	✓
location	Specifies where an output point will be created. Default is All	Literal	-

■ **Process Outputs**

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ **Constraints**

- The **inputFeatures** parameter must be line or polygon layers.
- The **location** parameter has the following five options.

Options	Description
All	All vertices of line or polygon geometry, Default value.
Mid	Midpoint of line or polygon geometry .
Start	Starting point of line or polygon geometry.
End	Endpoint of line or polygon geometry.
BothEnds	Starting and ending points of the line or polygon geometry.

■ **Request Examples**

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

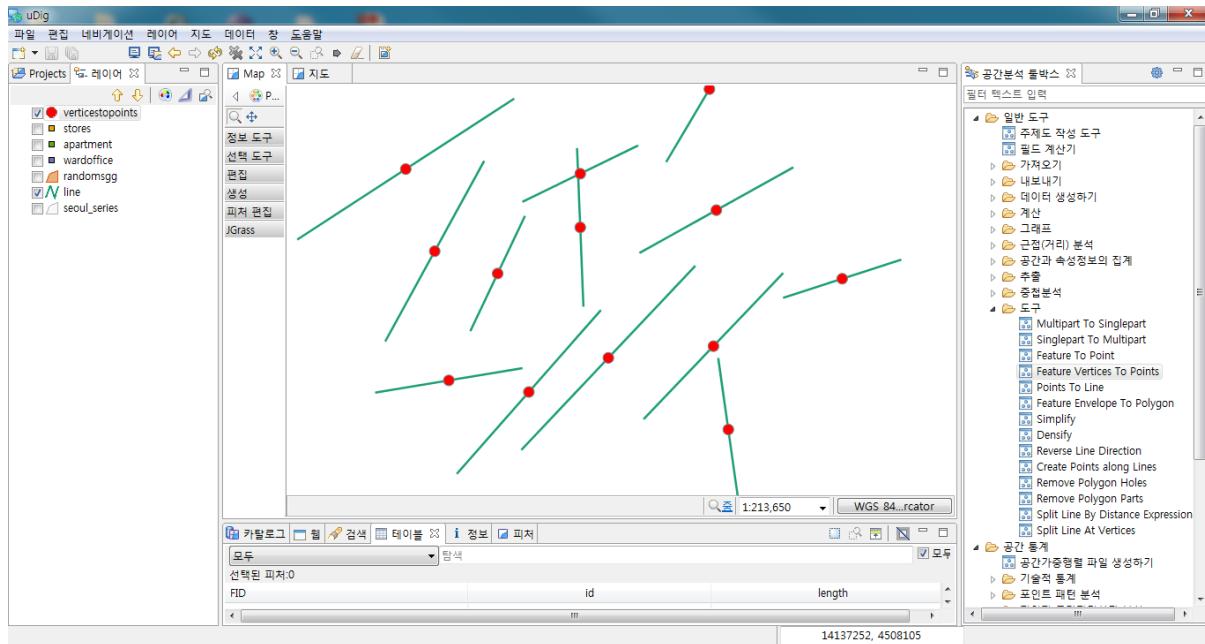
```

xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:VerticesToPoints</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:line" />
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>location</ows:Identifier>
<wps:Data>
<wps:LiteralData>Mid</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of converting the **Mid**(midpoint) of the line features to points.



4.2.9.13. Repair Geometry

Modifies the errors after validating the geometry of each feature in the feature layers (**inputFeatures**).

■ Syntax

RepairGeometry (inputFeatures SimpleFeatureCollection): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features that will be repaired.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Checks null geometry and self-intersection
- Validates coordinates
- Removes empty shell/holes and duplicated vertices/points

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RepairGeometry</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
```

```
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
  <wps:Body>
    <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
      xmlns:foss="http://www.opengeospatial.net/foss">
      <wfs:Query typeName="foss:seoul_series"/>
    </wfs:GetFeature>
  </wps:Body>
</wps:Reference>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

- None

4.2.9.14. Create Points along Line

Creating point feature layers with a constant distance (**distance**) as interval using polygon or line data (**lineFeatures**).

■ Syntax

PointsAlongLines (SimpleFeatureCollection lineFeatures, Expression distance):

SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
lineFeatures	The line or polygon features to be converted into points.	Complex	✓
distance	Field or Expression representing distance.	Literal	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output point features.	Complex	✓

■ Constraints

- The **inputFeatures** parameter must be line or polygon layers.
- The **distance** parameter can use the fields or the Function Expression formula.

■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:PointsAlongLines</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
```

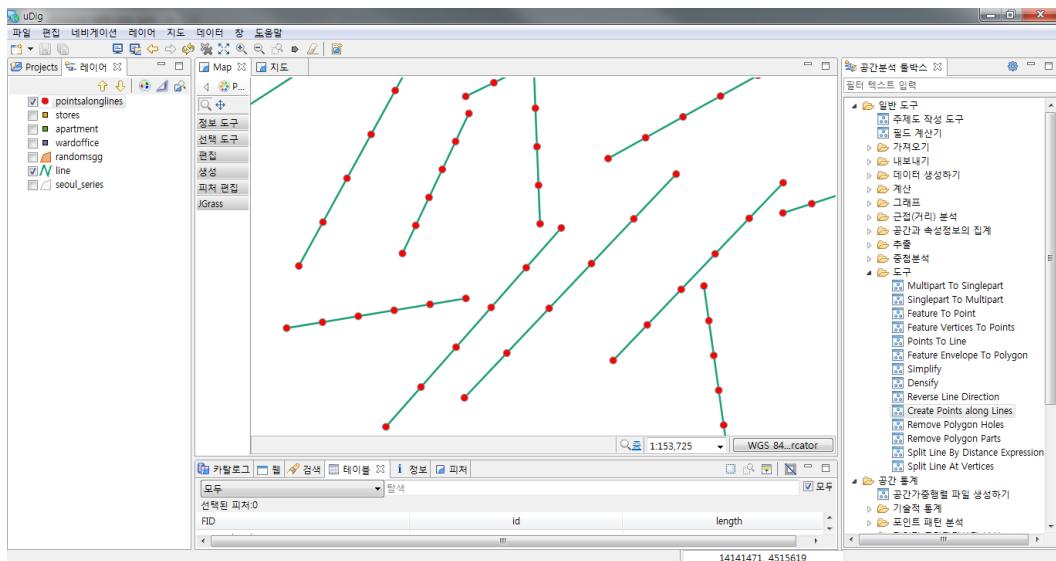
```

<ows:Identifier>lineFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
    <wps:Body>
        <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
            xmlns:foss="http://www.opengeospatial.net/foss"
                <wfs:Query typeName="foss:line" />
            </wfs:GetFeature>
        </wps:Body>
    </wps:Reference>
</wps:Input>
<wps:Input>
    <ows:Identifier>distance</ows:Identifier>
    <wps:Data>
        <wps:LiteralData>geomLength( [geom] ) / 5</wps:LiteralData>
    </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
        <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of creating points at 1/5 of the line length.



4.2.9.15. *Split Line at Point*

Splits the line feature layers (**lineFeatures**) into point feature layers (**pointFeatures**).

■ Syntax

```
SplitLineAtPoint (SimpleFeatureCollection lineFeatures, SimpleFeatureCollection  
pointFeatures, Double tolerance): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
lineFeatures	The line features to be splitted.	Complex	✓
pointFeatures	The point features whose locations will be used to split the line features.	Complex	✓
tolerance	Search radius. If tolerance is 0, the nearest point will be used to split the line feature.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output line features.	Complex	✓

■ Constraints

- The **inputFeatures** parameter must be line feature types.
- If the **tolerance** parameter is 0, all points that intersect with each line feature are used; if there are no intersecting point features, the closest point feature is used for line splitting.
- If the **tolerance** parameter is greater than or equal to 0, all point features within the search radius are used for partitioning.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"  
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
```

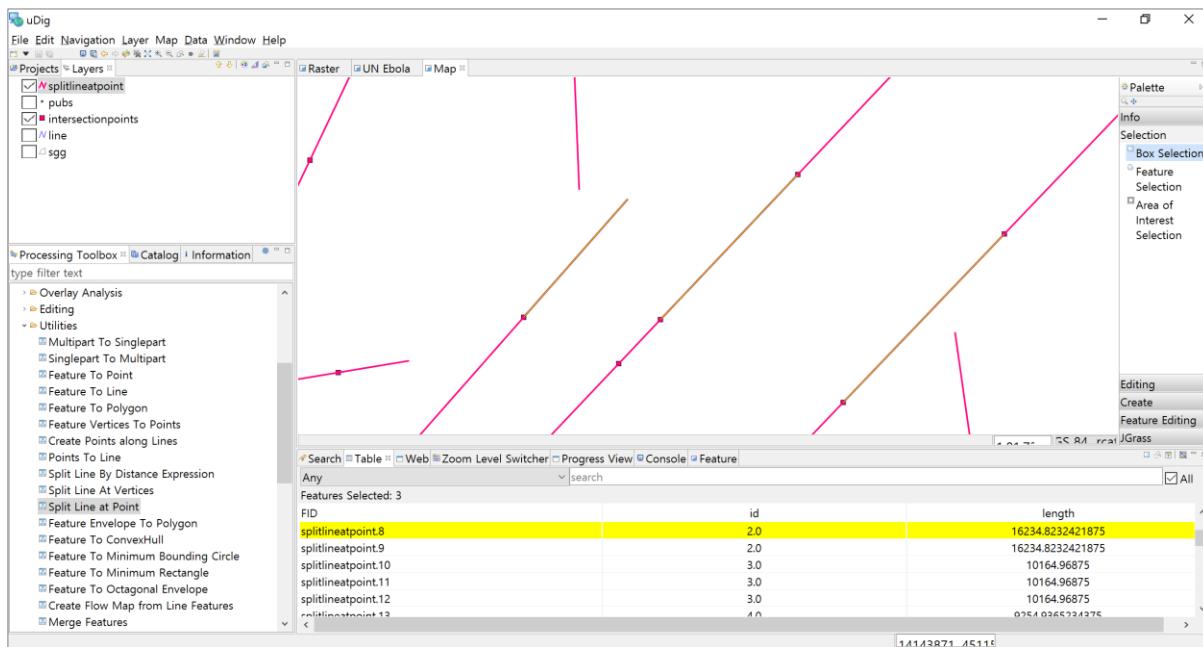
```

xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:SplitLineAtPoint</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>lineFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
            <wfs:Query typeName="foss:line"/>
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>pointFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
          <wfs:Query typeName="foss:stores"/>
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>tolerance</ows:Identifier>
    <wps>Data>
      <wps:LiteralData>50</wps:LiteralData>
    </wps>Data>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of splitting the line layers into point layers.



4.2.9.16. *Split Line at Vertices*

Creates line feature layers (**lineFeatures**) divided by vertices.

■ Syntax

SplitLineAtVertices (SimpleFeatureCollection **lineFeatures**): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
lineFeatures	The line or polygon features that will be splitted.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output line features.	Complex	✓

■ Constraints

- The **inputFeatures** parameter must be line or polygon layer types.

■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:SplitLineAtVertices</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>lineFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
            xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:randomsgg" />
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
  </wps:DataInputs>
</wps:Execute>
```

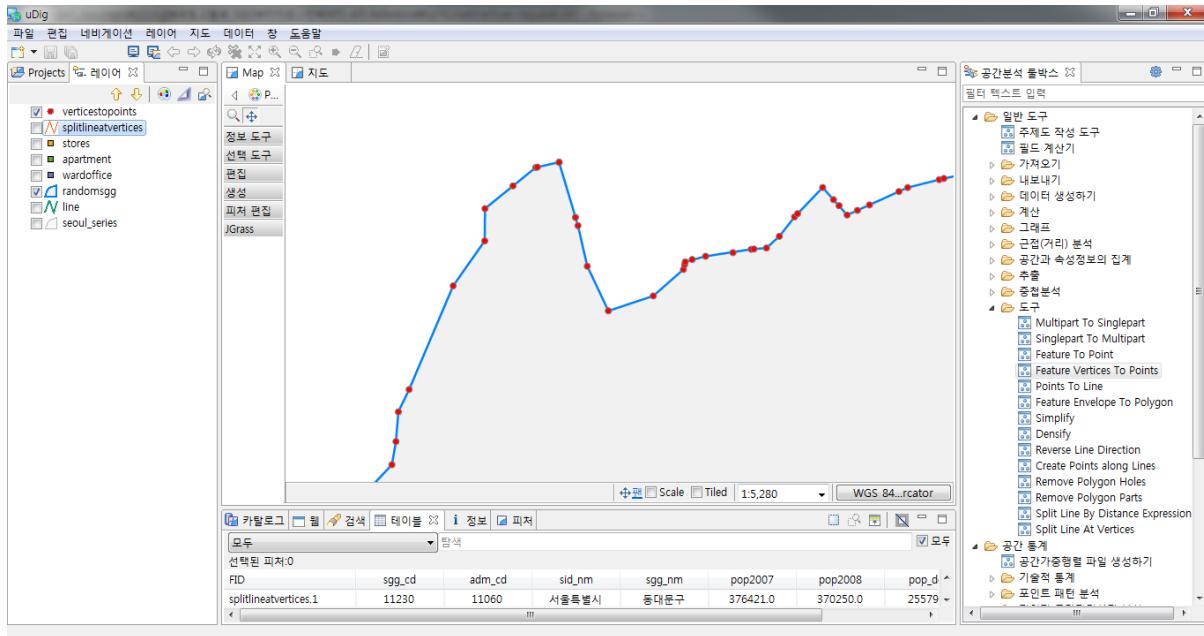
```

</wps:Body>
</wps:Reference>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of converting the boundary of polygon features to the line feature vertex by vertex.



4.2.9.17. *Split Line by Distance Expression*

Creates line feature layers (**lineFeatures**) divided by the set distance (**distance**).

■ Syntax

SplitLineByDistance (SimpleFeatureCollection **lineFeatures**, Expression **distance**):

SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
lineFeatures	The line features that will be splitted.	Complex	✓
distance	Field or Expression representing distance.	Literal	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output line features.	Complex	✓

■ Constraints

- The **inputFeatures** parameter must be line or polygon layer types.
- The **distance** parameter can use the fields or function expression formula.

■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:SplitLineByDistance</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>lineFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
```

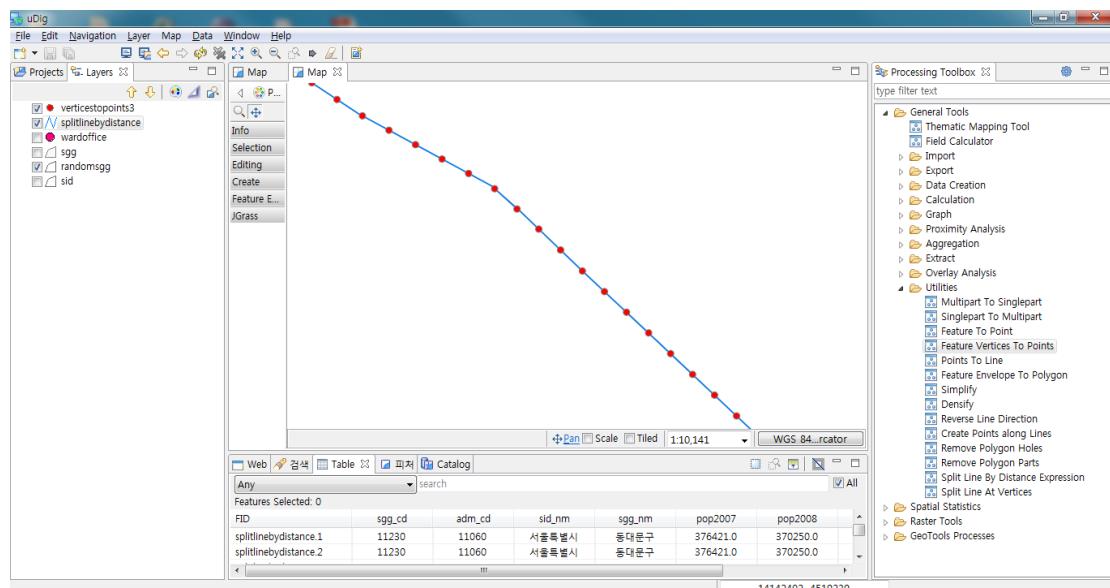
```

<wps:Body>
  <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
  xmlns:foss="http://www.opengeospatial.net/foss">
    <wfs:Query typeName="foss:randomsgg" />
  </wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>distance</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>100</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of dividing boundaries of polygon features into lines with an interval of 100 meters.



4.2.9.18. Intersection Points from Lines

Converts the nodes (**intersectFeatures**) where each line intersects in the two input polygon or line feature layers (**inputFeatures**) to points.

■ Syntax

IntersectionPoints (SimpleFeatureCollection **inputFeatures**, SimpleFeatureCollection **intersectFeatures**, String **intersectIDField**): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features that can be line or polygon type.	Complex	✓
intersectFeatures	Intersect that can be line or polygon type.	Complex	✓
intersectIDField	Intersect id field.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- **InputFeatures** and **intersectFeatures** feature layers can be set both line and polygon.
- The output layers contain all field values of **inputFeatures**. If **intersectIDField** is set, adds the value of **intersectFeatures**.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
```

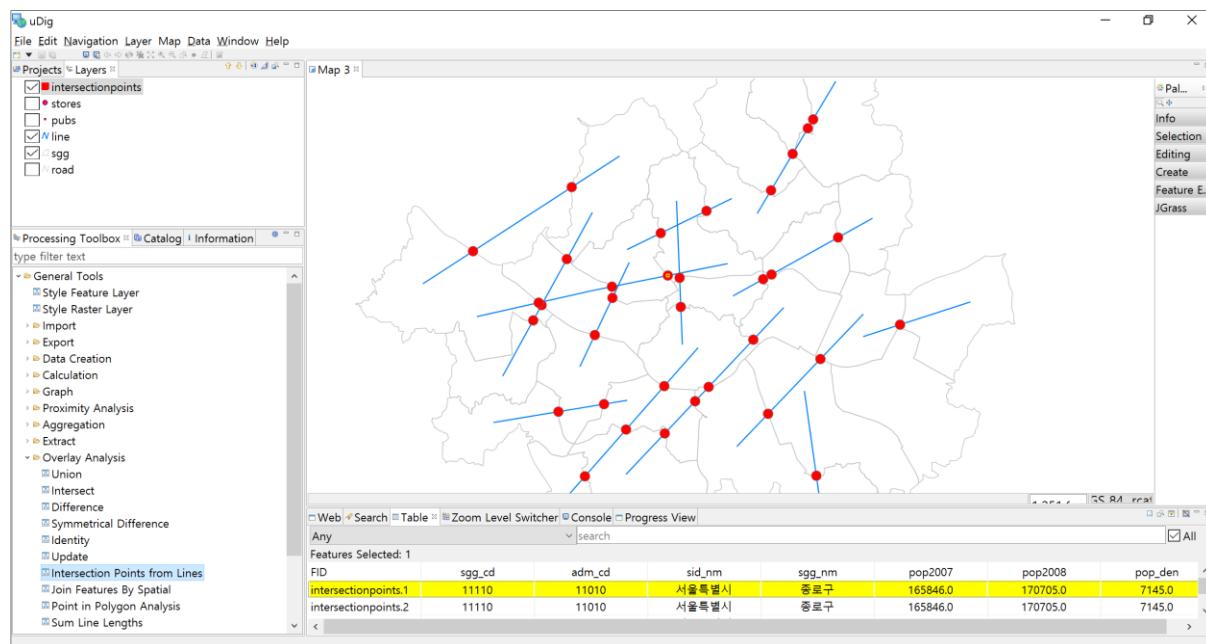
```

<ows:Identifier>statistics:IntersectionPoints</ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
        xmlns:foss="http://www.opengeospatial.net/foss">
          <wfs:Query typeName="foss:randomsgg"/>
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>intersectFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
        xmlns:foss="http://www.opengeospatial.net/foss">
          <wfs:Query typeName="foss:line"/>
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of analysis between polygon and line feature layers.



4.2.9.19. Create Flow Map from Line Features

Creates Polygon Flow Map feature layers using line feature layers (**lineFeatures**) consisting of Origin-Destination and attribute values (**odValue**, **doValue**) of it.

■ Syntax

FlowMap (SimpleFeatureCollection lineFeatures, Expression odValue, Expression doValue, Double maxSize): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
lineFeatures	The input line features.	Complex	✓
odValue	The o-d value expression. Ex) [field] or [field] * 0.5 etc...	Literal	✓
doValue	The d-o value expression. Ex) [field] or [field] * 0.5 etc...	Literal	-
maxSize	The maximum arrow size.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output polygon features.	Complex	✓

■ Constraints

- The **lineFeatures** parameter must be line layers.
- If the **maxSize** parameter is Null or 0, uses the value of dividing the smaller one between the width and height of the extents of the lineFeatures by 20.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
```

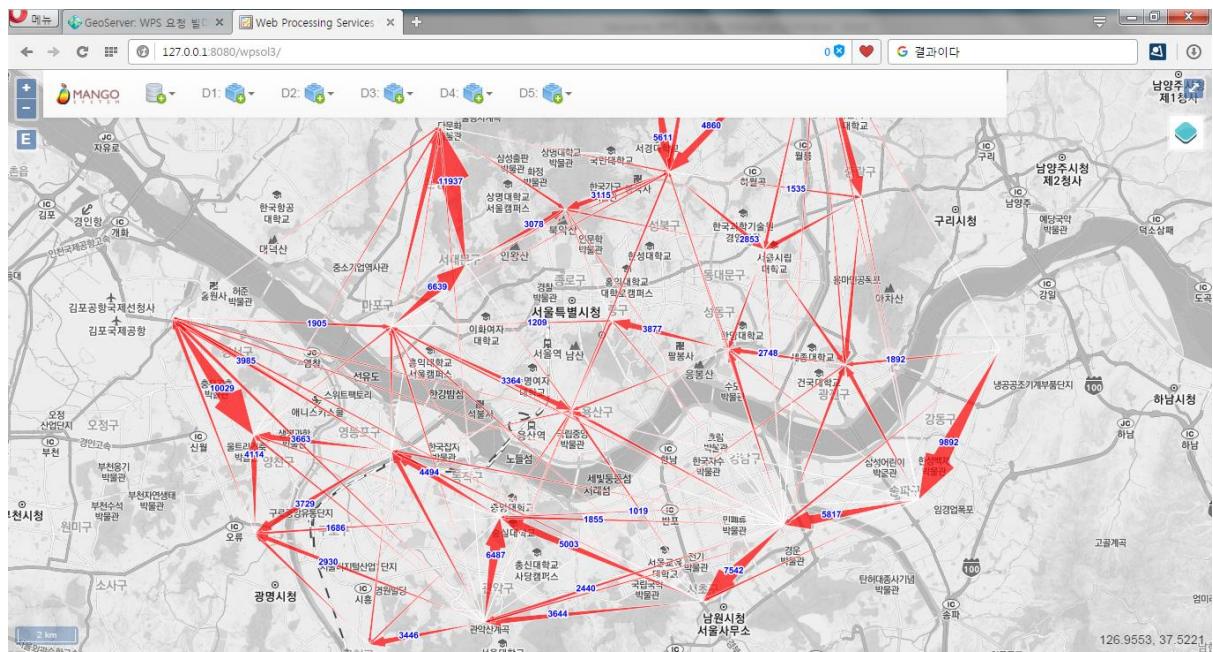
```

xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:FlowMap</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>lineFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:od_flow"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>odValue</ows:Identifier>
<wps:Data>
<wps:LiteralData>o_d</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>maxSize</ows:Identifier>
<wps:Data>
<wps:LiteralData>2500</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of generating the Flow Map using the population moving data of Seoul Gu-administrative district.



4.2.9.20. Feature To Octagonal Envelope

Converts each feature into the minimum bounding octagonal polygon that surrounds each feature in the feature layers (**inputFeatures**).

■ **Syntax**

FeatureToOctagonalEnvelope (SimpleFeatureCollection inputFeatures, Boolean singlePart):
SimpleFeatureCollection

■ **Parameters**

■ **Data Inputs**

Identifier	Description	Type	Required
inputFeatures	The input features that can be multipoint, line, polygon.	Complex	✓
singlePart	Specifies whether to use one circle for each entire multipart feature or one circle per part of a multipart feature.	Literal	-

■ **Process Outputs**

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ **Constraints**

- The **inputFeatures** can be points, lines, and polygons.
- If **singlePart** is set to True and the geometry is MultiPart, converts the geometry of all parts.

■ **Request Examples**

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0"
```

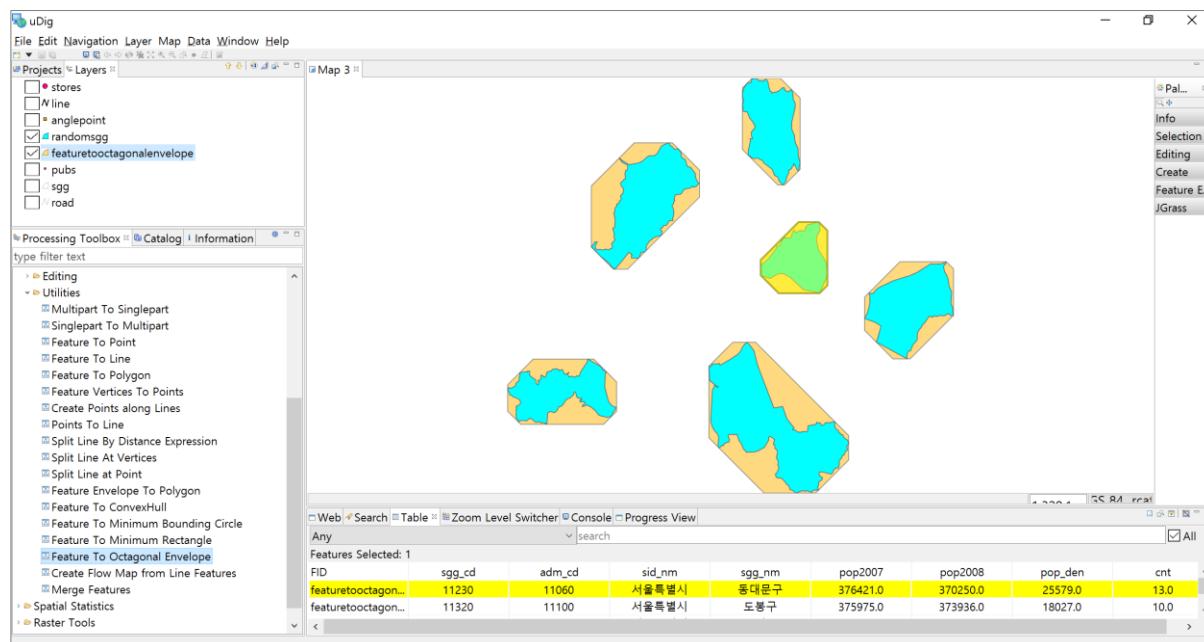
```

http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics: FeatureToOctagonalEnvelope</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
<wfs:Query typeName="foss:ssg"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of converting the polygon features to the minimum bounding octagonal polygons.



4.2.9.21. Feature To Minimum Rectangle

Converts each feature into the minimum bounding rectangle polygon that surrounds each feature in the feature layers (**inputFeatures**).

■ Syntax

FeatureToMinimumRectangle (SimpleFeatureCollection inputFeatures, Boolean singlePart):
SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input features that can be multipoint, line, polygon.	Complex	✓
singlePart	Specifies whether to use one circle for each entire multipart feature or one circle per part of a multipart feature.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- The **inputFeatures** can be points, lines, and polygons.
- If **singlePart** is set to True and the geometry is MultiPart, converts the geometry of all parts.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0"
```

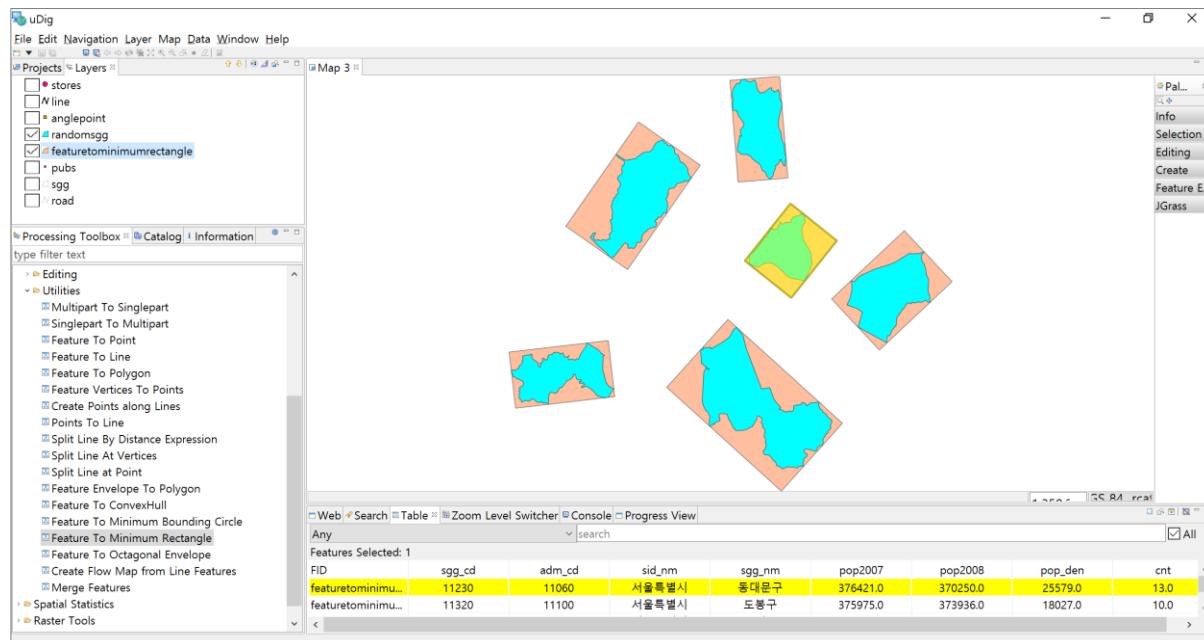
```

http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics: FeatureToMinimumRectangle</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
<wfs:Query typeName="foss:ssg"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of converting the polygon features to the minimum bounding rectangles.



4.2.9.22. Feature To ConvexHull

Converts each feature into a convex hull polygon that surrounds the feature in the feature layers (**inputFeatures**).

■ Syntax

FeatureToConvexHull (SimpleFeatureCollection inputFeatures, Boolean singlePart):

SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input features that can be multipoint, line, polygon.	Complex	✓
singlePart	Specifies whether to use one circle for each entire multipart feature or one circle per part of a multipart feature.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- The **inputFeatures** can be points, lines, and polygons.
- If **singlePart** is set to True and the geometry is MultiPart, converts the geometry of all parts.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
```

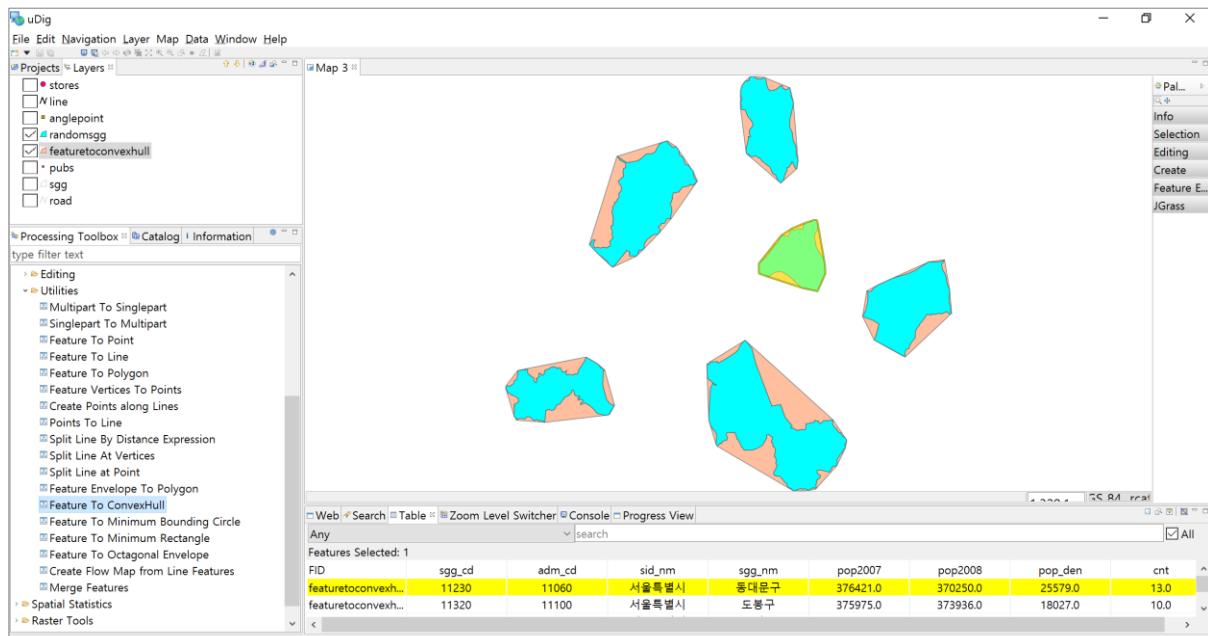
```

xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics: FeatureToConvexHull</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:ssg"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of converting the polygon features to ConvexHull.



4.2.9.23. Feature To Minimum Bounding Circle

Converts each feature into the minimum bounding circle that surrounds each feature in the feature layers (**inputFeatures**).

■ Syntax

FeatureToMinimumBoundingCircle (SimpleFeatureCollection **inputFeatures**, Boolean **singlePart**): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input features that can be multipoint, line, polygon.	Complex	✓
singlePart	Specifies whether to use one circle for each entire multipart feature or one circle per part of a multipart feature.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- The **inputFeatures** can be points, lines, and polygons.
- If **singlePart** is set to True and the geometry is MultiPart, converts the geometry of all parts.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0"
```

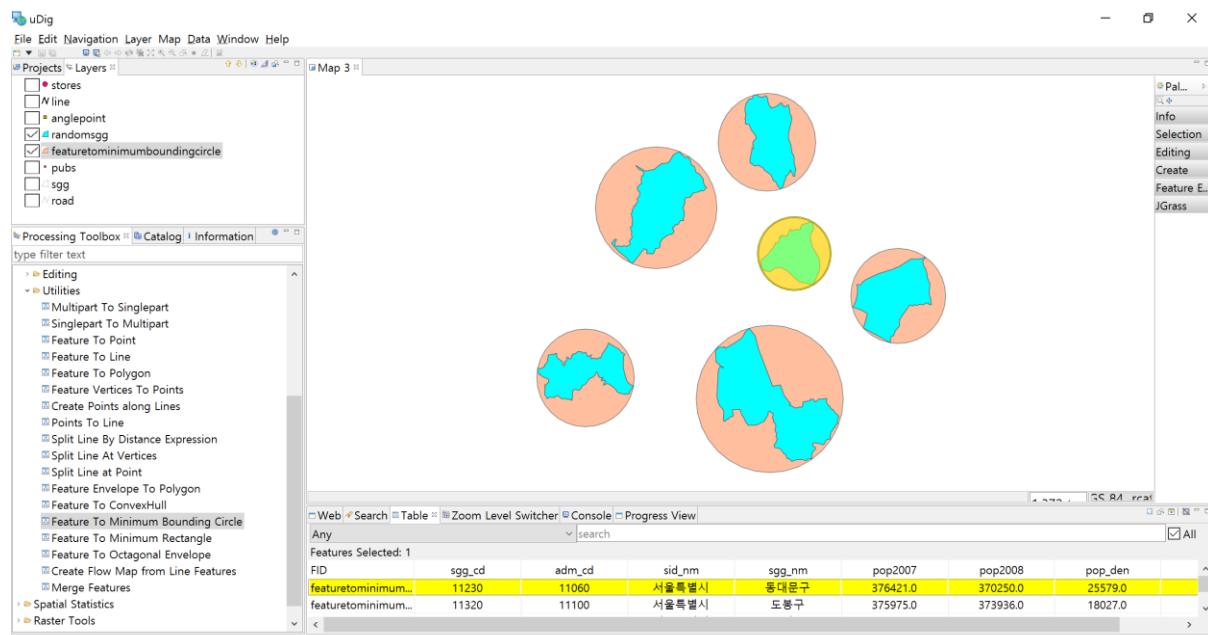
```

http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:FeatureToMinimumBoundingCircle</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:ssg"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of converting the polygon features to the minimum bounding circles.



4.3. Raster Analysis

These process groups are associated with raster analysis and processing.

4.3.1. Descriptive

These processes are for calculating the basic statistics for raster's property values.

4.3.1.1. Basic Statistics

Sets raster layers and specific areas to analyze basic statistics (Sum, Minimum, Maximum, Mean, Standard Deviation, etc.) for raster cell values contained in the area.

■ Syntax

StatisticsGridCoverage (GridCoverage2D inputCoverage, Geometry cropShape, Integer bandIndex): DataStatisticsResult

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The input gridcoverage to be calculated.	Complex	✓
cropShape	The Polygon or MultiPolygon to crop gridcoverage.	Complex	-
bandIndex	The zero-based band index, default index is a 0.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Result Statistics.	Complex	✓

■ Constraints

- If **cropShape** is set to Null, creates statistics for all cells in the input raster.
- The geometry type of **cropShape** must be Polygon or MultiPolygon.
- The **bandIndex** is zero-base and 0 is default value.
- Output is returned in XML format.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:StatisticsGridCoverage</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputCoverage</ows:Identifier>
      <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
        <wps:Body>
          <wcs:GetCoverage service="WCS" version="1.1.1">
            <ows:Identifier>foss:seoul_dem30</ows:Identifier>
            <wcs:DomainSubset>
              <gml:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#5181">
                <ows:LowerCorner>179171.39881047895 436569.3290600816</ows:LowerCorner>
                <ows:UpperCorner>216221.0981287582 466869.08315843146</ows:UpperCorner>
              </gml:BoundingBox>
            </wcs:DomainSubset>
            <wcs:Output format="image/tiff"/>
          </wcs:GetCoverage>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>cropShape</ows:Identifier>
      <wps:Data>
        <wps:ComplexData mimeType="application/wkt"><![CDATA[POLYGON
((.....))]]></wps:ComplexData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

■ Response

```
<?xml version="1.0" encoding="utf-8"?>
<DataStatistics>
  <Item>
    <TypeName>dem</TypeName>
    <PropertyName>Value</PropertyName>
    <Count>678064</Count>
    <InvalidCount>0</InvalidCount>
    <Minimum>1.0</Minimum>
    <Maximum>754.0</Maximum>
    <Range>753.0</Range>
    <Ranges>1.0 - 754.0</Ranges>
    <Sum>4.2785658E7</Sum>
    <Mean>63.09973394841785</Mean>
    <Variance>7285.154424054373</Variance>
    <StandardDeviation>85.35311607700315</StandardDeviation>
    <CoefficientOfVariance>1.3526699834705607</CoefficientOfVariance>
    <NoData class="double">-9999</NoData>
  </Item>
</DataStatistics>
```

4.3.1.2. Histogram

Sets raster layers and specific regions to extract unique values and frequencies of raster cells contained within the regions.

■ Syntax

HistogramGridCoverage (GridCoverage2D inputCoverage, Geometry cropShape, Integer bandIndex): GridCoverage2D

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The input gridcoverage to be calculated.	Complex	✓
cropShape	The Polygon or MultiPolygon to crop gridcoverage.	Complex	-
bandIndex	The zero-based band index, default index is a 0.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Result Statistics.	Complex	✓

■ Constraints

- The geometry type of **cropShape** must be Polygon or MultiPolygon tpye.
- The **bandIndex** is zero-base and 0 is the default value.
- Output is returned in XML format.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
```

```

<ows:Identifier>statistics:HistogramGridCoverage</ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputCoverage</ows:Identifier>
    <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
      <wps:Body>
        <wcs:GetCoverage service="WCS" version="1.1.1">
          <ows:Identifier>foss:landuse</ows:Identifier>
          <wcs:DomainSubset>
            <gml:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#5181">
              <ows:LowerCorner>179171.39881047895 436569.3290600816</ows:LowerCorner>
              <ows:UpperCorner>216221.0981287582 466869.08315843146</ows:UpperCorner>
            </gml:BoundingBox>
          </wcs:DomainSubset>
          <wcs:Output format="image/tiff"/>
        </wcs:GetCoverage>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>cropShape</ows:Identifier>
    <wps:Data>
      <wps:ComplexData mimeType="application/wkt"><![CDATA[MULTIPOLYGON
(((202045.8134286803 451170.87479061395, 202045.8134286803 456372.83051287895,
206947.46547550958 456372.83051287895, 206947.46547550958 451170.87479061395,
202045.8134286803 451170.87479061395)))]]></wps:ComplexData>
    </wps:Data>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of creating a statistic by clipping a specific area of raster layers with the same grade in the Environmental Conservation Value Assessment Map(ECVAM). Since the number of cells for each value is returned, multiplying by one cell area (CellSize * CellSize) yields the area by grade.

```
<?xml version="1.0" encoding="utf-8"?>
<Histogram>
  <TypeName>landuse</TypeName>
  <PropertyName>Value</PropertyName>
  <Area>25498176.913556</Area>
  <CellSize>30.0</CellSize>
  <HistogramItem>
    <Value>1</Value>
    <Frequency>876</Frequency>
  </HistogramItem>
  <HistogramItem>
    <Value>2</Value>
    <Frequency>543</Frequency>
  </HistogramItem>
  <HistogramItem>
    <Value>3</Value>
    <Frequency>292</Frequency>
  </HistogramItem>
  <HistogramItem>
    <Value>4</Value>
    <Frequency>1345</Frequency>
  </HistogramItem>
  <HistogramItem>
    <Value>5</Value>
    <Frequency>765</Frequency>
  </HistogramItem>
</Histogram>
```

4.3.2. Conversion

These processes are for conversion between vector data and raster data.

4.3.2.1. Features To Raster

Converts point, line, and polygon feature layers (**inputFeatures**) to raster.

■ Syntax

```
FeaturesToRaster (SimpleFeatureCollection inputFeatures, String inputField, Double  
cellSize, ReferencedEnvelope extent): GridCoverage2D
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input feature dataset to be converted to a raster.	Complex	✓
inputField	The field used to assign values to the output raster.	Literal	-
cellSize	The cell size for the output raster.	Literal	-
extent	The extent for the output raster.	Complex	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- The **inputField** parameter must be a numeric field or a constant value.
- Unless sets the **extent** parameter, uses the range of **inputFeatures**.
- Unless sets the **cellSize** parameter, the smaller one between width and height of the extent is divided by 250.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
```

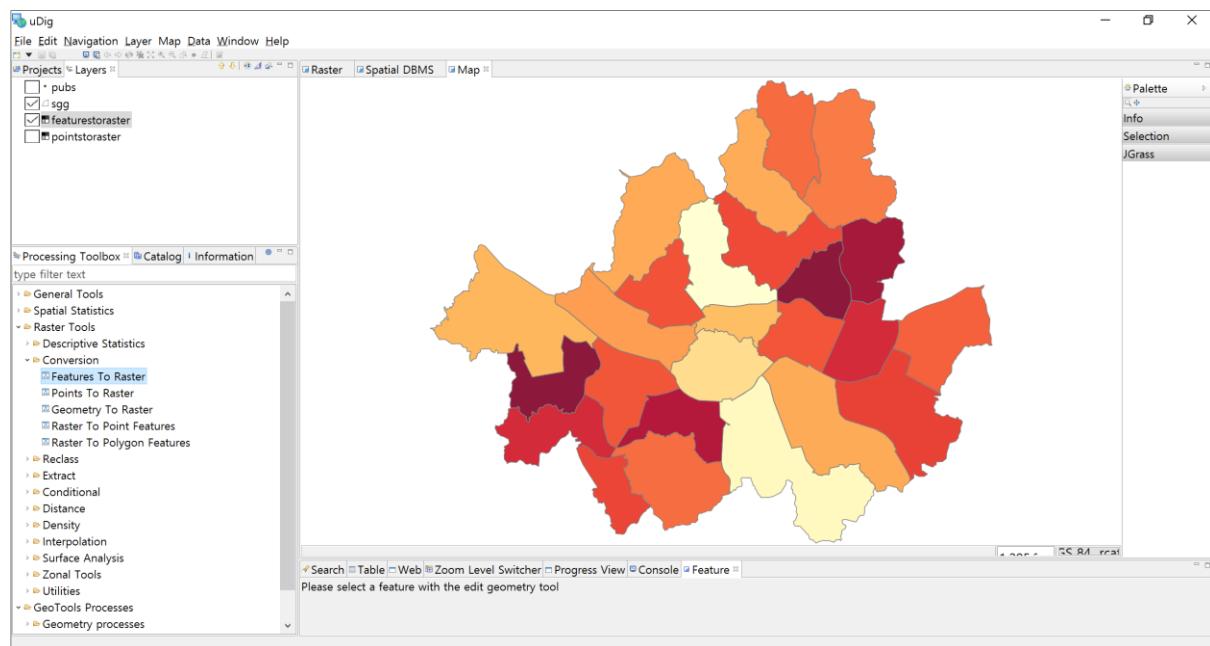
```

xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:FeaturesToRaster</ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
<xmlns:foss="http://www.opengeospatial.net/foss">
  <wfs:Query typeName="foss:ssg"/>
</wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>inputField</ows:Identifier>
    <wps>Data>
      <wps:LiteralData>pop_den</wps:LiteralData>
    </wps>Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>cellSize</ows:Identifier>
    <wps>Data>
      <wps:LiteralData>30</wps:LiteralData>
    </wps>Data>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="image/tiff">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of the converting the polygon geometry to raster data.



4.3.2.2. Points To Raster

Converts the point feature layers (**inputFeatures**) into raster using the cell value assignment method.

■ Syntax

```
PointsToRaster (SimpleFeatureCollection inputFeatures, String inputField,  
PointAssignmentType cellAssignment, Double cellSize, ReferencedEnvelope extent):  
GridCoverage2D
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The point or multipoint input feature dataset to be converted to a raster.	Complex	✓
inputField	The field used to assign values to the output raster.	Literal	✓
cellAssignment	The method to determine how the cell will be assigned a value when more than one feature falls within a cell.	Literal	-
cellSize	The cell size for the output raster.	Literal	-
extent	The extent for the output raster.	Complex	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- The **inputFeatures** parameter must be point or multiPoint feature types.
- The **inputField** parameter must be numeric fields or values(constant value).
- Unless sets the **extent** parameter, uses the range of the **inputGeometry**.
- Unless sets the **cellAssignment** parameter, chooses the smaller value between extent's width and height, and divides it by 250.

- The **cellAssignment** parameter defines how to assign cell values when more than one point is nested in a cell, and following options can be used.

Option	Description
MostFrequent	Frequency, If the frequency is the same, it arranged by the features order
Sum	Sum
Mean	Mean
Maximum	Maximum
Minimum	Minimum
Range	Range
Count	The number of points, Force assignment even if not a numeric field

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:PointsToRaster</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:pubs"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>inputField</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>gid</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>cellAssignment</ows:Identifier>
```

```

<wps:Data>
  <wps:LiteralData>Count</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>cellSize</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>30</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="image/tiff">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of converting the polygon geometry to raster data.



4.3.2.3. Geometry To Raster

Converts geometry (**inputGeometry**) to raster.

■ Syntax

GeometryToRaster (Geometry inputGeometry, CoordinateReferenceSystem forcedCRS, Number defaultValue, RasterPixelType pixelType, Double cellSize, ReferencedEnvelope extent): GridCoverage2D

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputGeometry	The input geometry to be converted to a raster dataset.	Complex	✓
forcedCRS	Coordinate reference system to use for input geometry.	Literal	-
defaultValue	The default value for the output pixel: 1(default).	Literal	-
pixelType	The pixel type for the output raster.	Literal	-
cellSize	The cell size for the output raster.	Literal	-
extent	The extent for the output raster.	Complex	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- If the **forcedCRS** parameter is set to Null, uses CRS value of **inputGeometry**, and CRS of **inputGeometry** must be set.
- If the **defaultValue** parameter is set to Null, the default value is 1(Integer).
- If the **pixelType** parameter is set to Null, uses Integer as the default value.
- The **pixelType** parameter can use BYTE, SHORT, INTEGER, FLOAT, and DOUBLE values.
- Unless sets the **extent** parameter, uses the range of the **inputGeometry**.

- Unless sets the **cellSize** parameter, chooses the smaller value between extent's width and height, and divides it by 250.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:GeometryToRaster</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputGeometry</ows:Identifier>
      <wps:Data>
        <wps:ComplexData mimeType="application/wkt"><![CDATA[POLYGON ((14134957.759521019
4527300.782425176, ... 14134957.759521019 4527300.782425176))]]></wps:ComplexData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>forcedCRS</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>EPSG:3857</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>defaultValue</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>1</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>pixelType</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>INTEGER</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>cellSize</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>50</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
</wps:Execute>
```

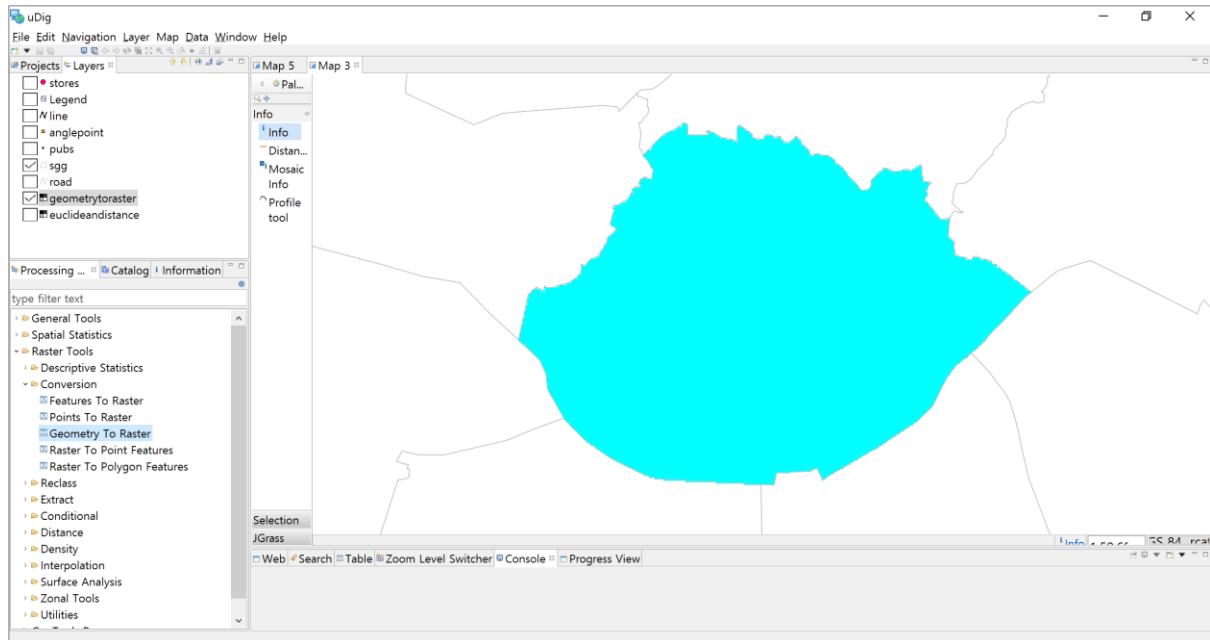
```

</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="image/tiff">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of converting the polygon geometry to raster data.



4.3.2.4. Raster To Point

Converts raster layers (**inputCoverage**) to point layers.

■ Syntax

RasterToPoint (GridCoverage2D inputCoverage, Integer bandIndex, String valueField):

SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The input gridcoverage to be converted.	Complex	✓
bandIndex	The zero-based band index, default index is 0.	Literal	-
valueField	The field used to assign values from the cells. Default is value.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- If **valueField** is set to Null, uses the fields of value names by default.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RasterToPoint</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputCoverage</ows:Identifier>
      <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
```

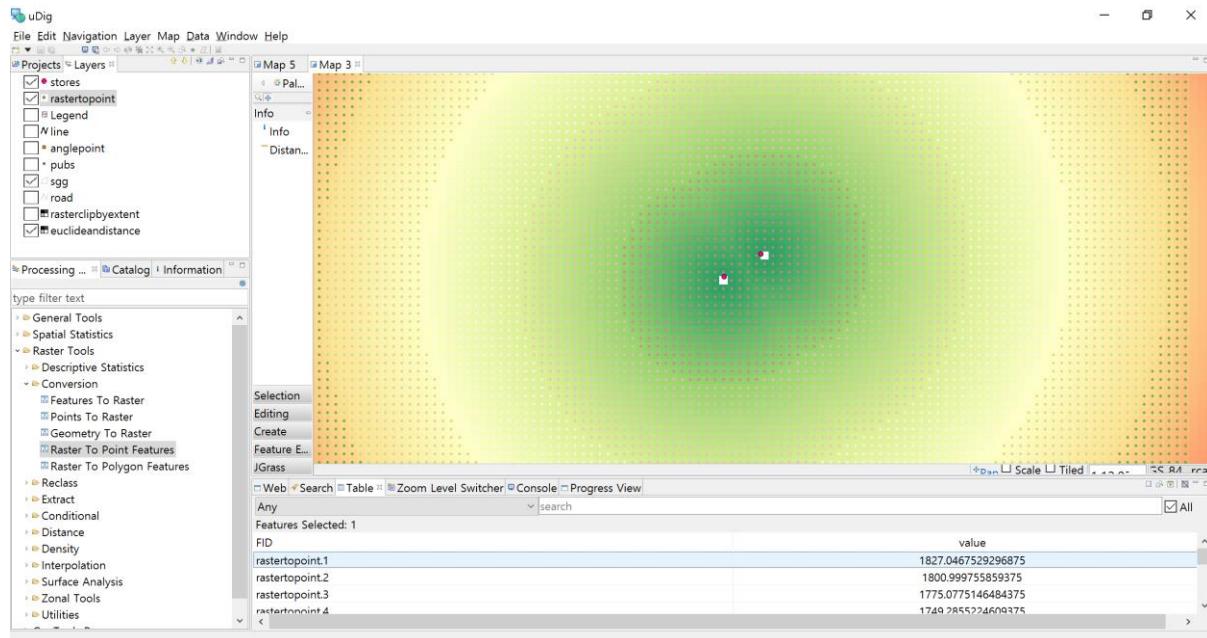
```

<wps:Body/>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>valueField</ows:Identifier>
<wps:Data>
<wps:LiteralData>val</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of converting raster data to points.



4.3.2.5. Raster to Polygon

Converts raster layers (**inputCoverage**) to polygon layers.

■ Syntax

RasterToPolygon (GridCoverage2D inputCoverage, Integer bandIndex, Boolean weeding, String valueField): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The input gridcoverage to be converted.	Complex	✓
bandIndex	The zero-based band index, default index is 0.	Literal	-
weeding	Determines if the output polygons will be smoothed into simpler shapes. Default is False.	Literal	-
valueField	The field used to assign values from the cells. Default is value.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- If **weeding** is set to True, simplifies it using the Douglas-Puecker algorithm. The tolerance is $\text{sqrt}(0.5) * \text{cell size}$.
- If **valueField** is set to Null, uses the fields of value names by default.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
```

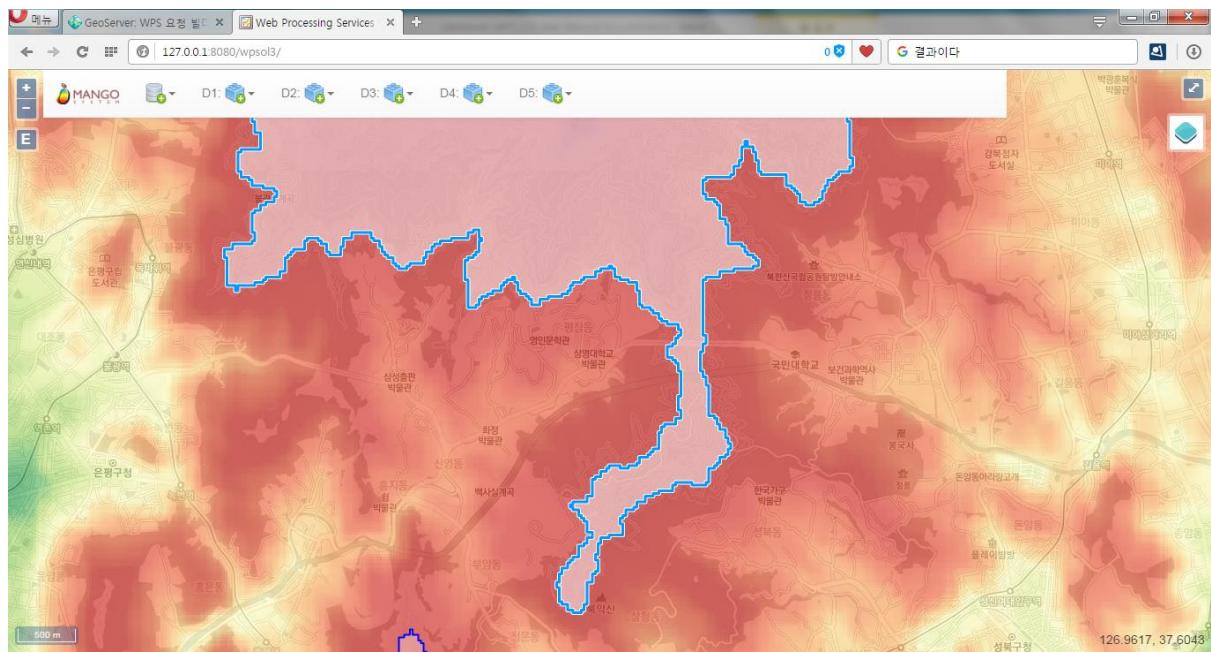
```

<ows:Identifier>statistics:RasterToPolygon</ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputCoverage</ows:Identifier>
    <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
      <wps:Body>
        <wcs:GetCoverage service="WCS" version="1.1.1">
          <ows:Identifier>foss:seoul_dem30</ows:Identifier>
          <wcs:DomainSubset>
            <ows:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#3857">
              <ows:LowerCorner>1.4111343323506365E7 4498971.750719266</ows:LowerCorner>
              <ows:UpperCorner>1.4158021303411832E7 4537343.6431004135</ows:UpperCorner>
            </ows:BoundingBox>
          </wcs:DomainSubset>
          <wcs:Output format="image/tiff"/>
        </wcs:GetCoverage>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>weeding</ows:Identifier>
    <wps>Data>
      <wps:LiteralData>True</wps:LiteralData>
    </wps>Data>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of converting the DEM layer to polygon layers after extracting features with elevation larger than 250 meters above sea level from DEM.



4.3.2.6. Raster to Image

Converts a raster layer (**coverage**) to an image using WMS parameter.

■ Syntax

RasterToImage (GridCoverage2D coverage, String bbox, CoordinateReferenceSystem crs, Style style, Integer width, Integer height, String format, Boolean transparent, String bgColor): Image

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
coverage	The input gridcoverage to be converted.	Complex	✓
bbox	Bounding box corners (lower left, upper right): minx, miny, maxx, maxy.	Literal	-
crs	CRS for Bounding Box. Ex) EPSG:3857	Literal	-
style	Styled Layer Descriptor (SLD) style containing a raster symbolizer.	Complex	-
width	Image width in pixels of resulting map.	Literal	✓
height	Image height in pixels of resulting map.	Literal	✓
format	Output format of map. Valid values are image/jpeg, image/png (Default), and image/gif.	Literal	-
transparent	Map background transparency. Default is True.	Literal	-
bgColor	Hexidecimal red-blue-green color value for the map background color. Default is 0xFFFFFFFF (white).	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output image.	Complex	✓

■ Constraints

- If the **bbox** and **crs** parameter are set to Null, uses the extents and coordinate system of the **coverage**.
- If the **style** parameter is set to Null, applies the Equal Interval Style using the minimum / maximum value.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RasterToImage</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>coverage</ows:Identifier>
      <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
        <wps:Body>
          <wcs:GetCoverage service="WCS" version="1.1.1">
            <ows:Identifier>foss:seoul_dem30</ows:Identifier>
            <wcs:DomainSubset>
              <ows:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#3857">
                <ows:LowerCorner>1.4111343323506365E7 4498971.750719266</ows:LowerCorner>
                <ows:UpperCorner>1.4158021303411832E7 4537343.6431004135</ows:UpperCorner>
              </ows:BoundingBox>
            </wcs:DomainSubset>
            <wcs:Output format="image/tiff"/>
          </wcs:GetCoverage>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>width</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>500</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>height</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>400</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>format</ows:Identifier>
      <wps>Data>
```

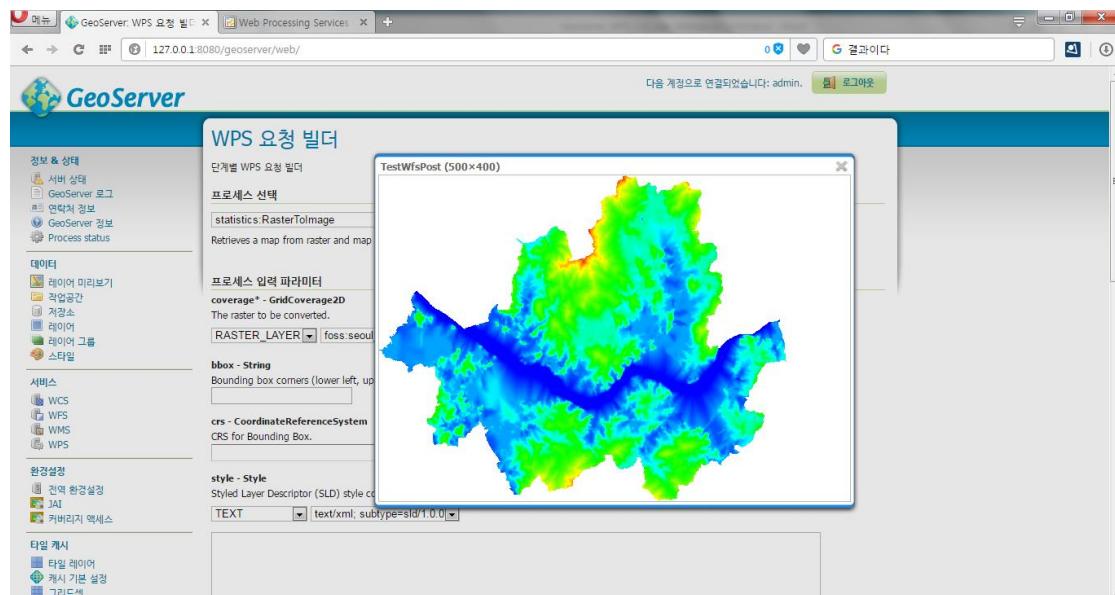
```

<wps:LiteralData>image/png</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>transparent</ows:Identifier>
<wps:Data>
<wps:LiteralData>True.</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="image/png">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of requesting the Seoul DEM layer to an image (500 by 400 pixels). The requested result can be added to the image layer in OpenLayers.



4.3.3. Distance

Analysis the distance and adjacency of vector and raster data.

4.3.3.1. Euclidean Distance

Creates a raster dataset based on Euclidean distance between each cell and feature.

■ Syntax

EuclideanDistance (SimpleFeatureCollection inputFeatures, Double maximumDistance, Double cellSize, ReferencedEnvelope extent): GridCoverage2D

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input features for which to calculate the distance.	Complex	✓
maximumDistance	Defines the threshold that the accumulative distance values cannot exceed.	Literal	-
cellSize	The cell size for the output raster.	Literal	-
extent	The extent for the output raster.	Complex	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- If set the **maximumDistance** parameter, areas above the specified distance will be assigned the No Data value.
- Unless set the **extent** parameter, use the range of the **inputFeatures** layer.
- Unless set the **cellSize** parameter, choose the smaller value between Extent's Width and Height, and divide it by 250.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
```

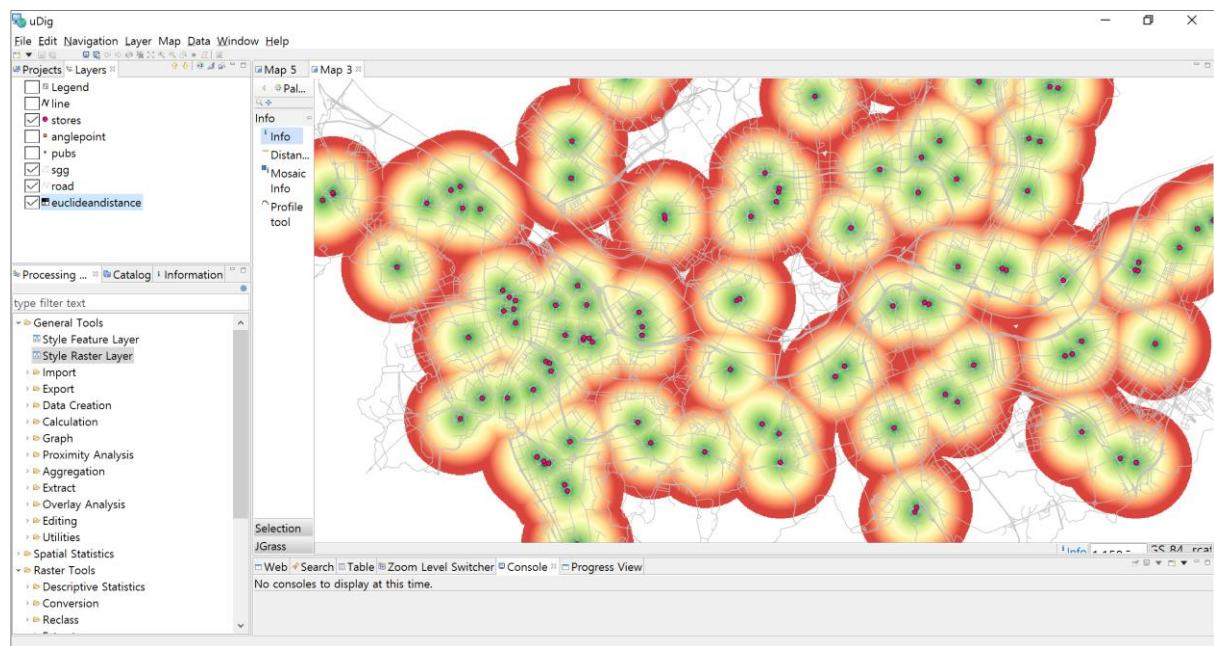
```

xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:EuclideanDistance</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
            <wfs:Query typeName="foss:stores"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>maximumDistance</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>2500</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>cellSize</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>30</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="image/tiff">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of Euclidean Distance analysis with maximum distance of 2.5km and cell size of 30m, using Seoul big store point data.



4.3.4. Math

Createsthe new raster using a filter or formula.

4.3.4.1. Math Operation

Use the formula to create a new raster.

■ Syntax

RasterMath (GridCoverage2D inputCoverage, Integer bandIndex, Expression expression):

GridCoverage2D

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The input raster to be calculated.	Complex	✓
bandIndex	The zero-based band index, default index is a 0.	Literal	-
expression	A mathematical expression that evaluates raster cells. Ex> log(Value * 250)	Literal	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- The functions available in the expression parameter use the Filter Function⁴ of GeoTools.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
```

⁴ http://docs.geoserver.org/stable/en/user/filter/function_reference.html

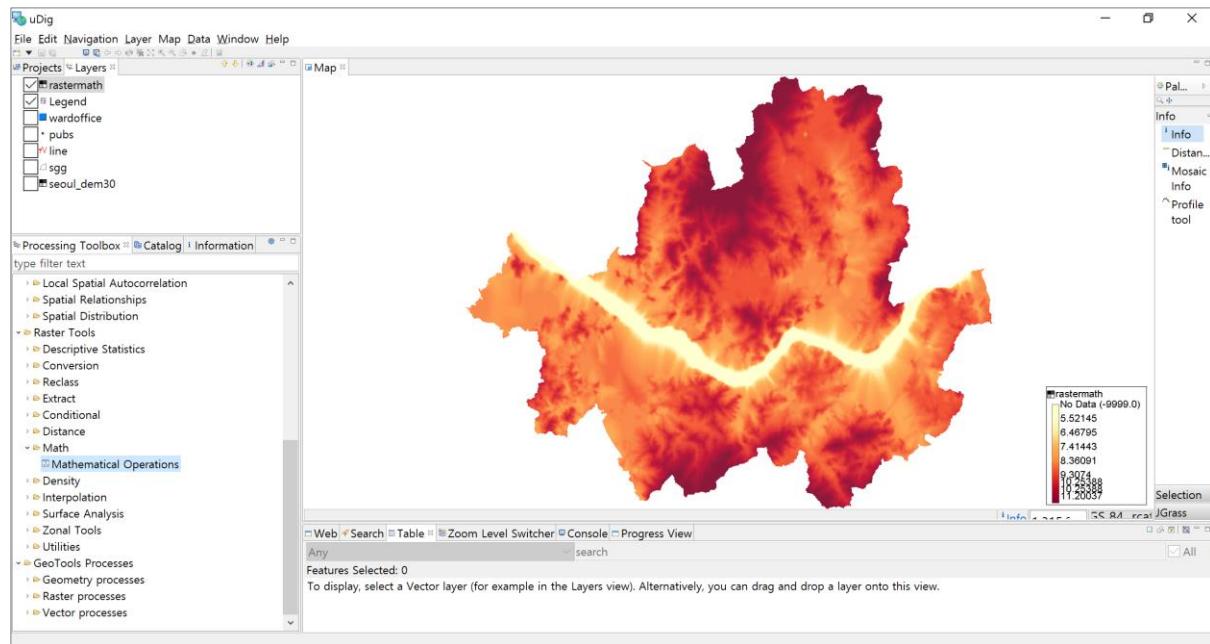
```

xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RasterMath</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputCoverage</ows:Identifier>
      <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
        <wps:Body>
          <wcs:GetCoverage service="WCS" version="1.1.1">
            <ows:Identifier>cite:dem05</ows:Identifier>
            <wcs:DomainSubset>
              <ows:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#5181">
                <ows:LowerCorner>321541.5348100797 235208.86605789233</ows:LowerCorner>
                <ows:UpperCorner>359950.1365611528 281299.18815918005</ows:UpperCorner>
              </ows:BoundingBox>
            </wcs:DomainSubset>
            <wcs:Output format="image/tiff"/>
          </wcs:GetCoverage>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>bandIndex</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>0</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>expression</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>log(Value * 250)</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="image/tiff">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows result of applying $\log(\text{Value} * 250)$ calculation formula in Seoul DEM raster data.



4.3.4.2. Set Null

Converts the cell value that corresponds to the filter condition to a NoData value, or replaces the existing NoData value with another value.

■ Syntax

RasterSetNull (GridCoverage2D inputCoverage, Integer bandIndex, Filter filter, Boolean replaceNoData, Double newValue): GridCoverage2D

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The input raster to be evaluated.	Complex	✓
bandIndex	The zero-based band index, default index is a 0.	Literal	-
filter	A logical expression that determines which of the input cells are to be nodata. ex> Value > 250	Literal	✓
replaceNoData	If true, nodata value will be replaced as a newValue parameter. Default is False.	Literal	
newValue	The new valid value to replace nodata.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- The field name of the **filter** parameter must be Value.
- If the value of the **replaceNoData** parameter is True, replace the existing NoData value with the value of the **newValue** parameter.

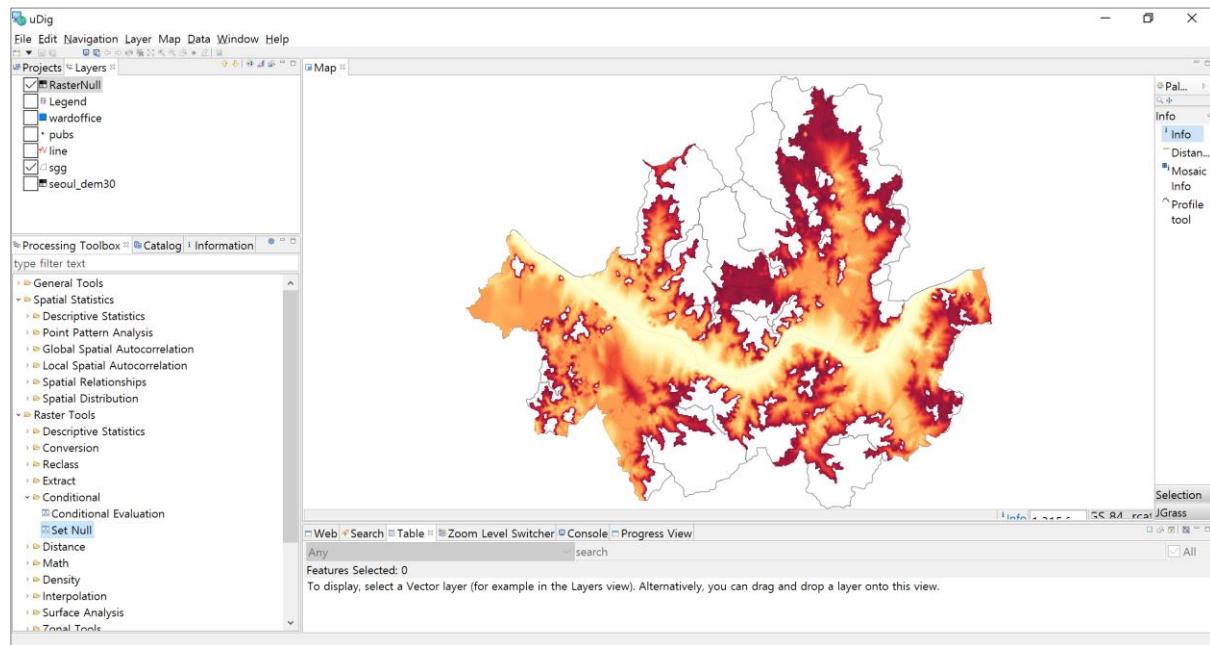
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
```

```
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RasterSetNull</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputCoverage</ows:Identifier>
      <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
        <wps:Body>
          <wcs:GetCoverage service="WCS" version="1.1.1">
            <ows:Identifier>cite:dem30</ows:Identifier>
            <wcs:DomainSubset>
              <ows:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#5181">
                <ows:LowerCorner>322223.9108718962 235108.16003333713</ows:LowerCorner>
                <ows:UpperCorner>360624.44972726464 281188.51233892346</ows:UpperCorner>
              </ows:BoundingBox>
            </wcs:DomainSubset>
            <wcs:Output format="image/tiff"/>
          </wcs:GetCoverage>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>bandIndex</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>0</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>filter</ows:Identifier>
      <wps>Data>
        <wps:ComplexData mimeType="text/plain; subtype=cql"><![CDATA[Value >
250]]></wps:ComplexData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="image/tiff">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of processing NoData areas, where the value is more than 50 in the Seoul DEM raster data.



4.3.5. Classification

Reclass the raster values to a specific range of values.

4.3.5.1. Reclass

Reclass the raster data to the given range and the assigned value of each range.

■ Syntax

RasterReclass (GridCoverage2D inputCoverage, Integer bandIndex, String ranges):

GridCoverage2D

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The input raster to be reclassified.	Complex	✓
bandIndex	The zero-based band index, default index is a 0.	Literal	-
ranges	Ranges that defines how the values will be reclassified. ex) 0.0 30.0 1; 30.0 270.0 2; 270.0 365.0 3	Literal	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- The values in the **ranges** parameter treated as NoData values.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RasterReclass</ows:Identifier>
  <wps:DataInputs>
```

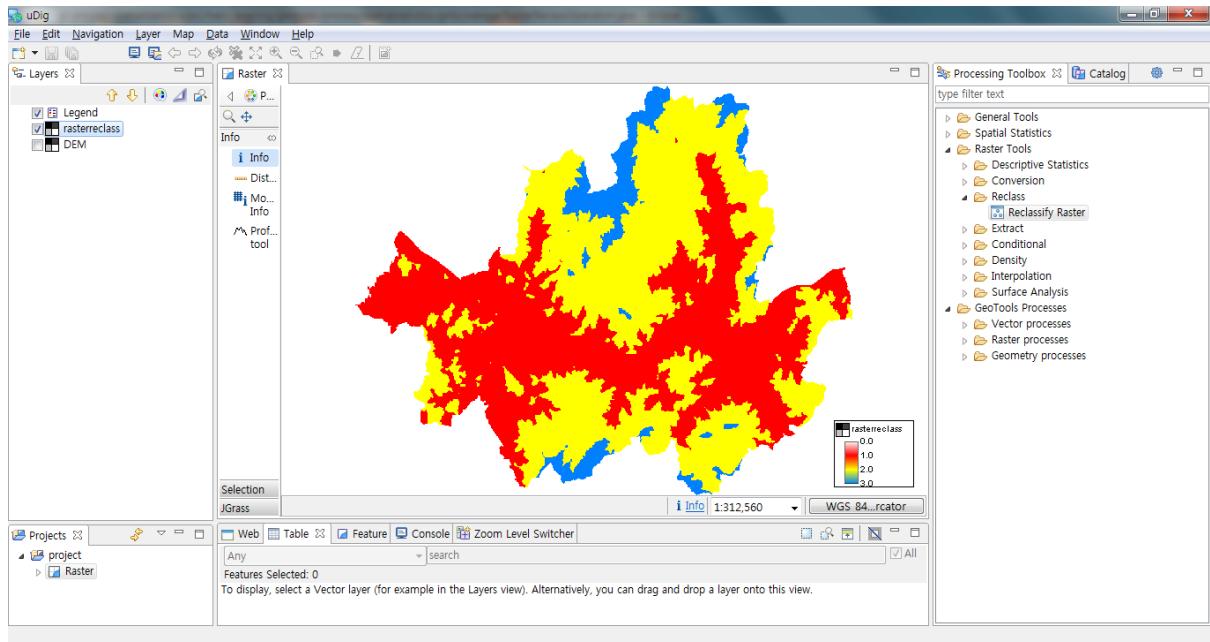
```

<wps:Input>
  <ows:Identifier>inputCoverage</ows:Identifier>
  <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
    <wps:Body>
      <wcs:GetCoverage service="WCS" version="1.1.1">
        <ows:Identifier>foss:seoul_dem30</ows:Identifier>
        <wcs:DomainSubset>
          <ows:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#3857">
            <ows:LowerCorner>1.4111343323506365E7 4498971.750719266</ows:LowerCorner>
            <ows:UpperCorner>1.4158021303411832E7 4537343.6431004135</ows:UpperCorner>
          </ows:BoundingBox>
        </wcs:DomainSubset>
        <wcs:Output format="image/tiff"/>
      </wcs:GetCoverage>
    </wps:Body>
  </wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>ranges</ows:Identifier>
  <wps>Data>
    <wps:LiteralData>0.1 30.0 1; 30.0 200.0 2; 200.0 500.0 3</wps:LiteralData>
  </wps>Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="image/tiff">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of Reclass the Seoul DEM layer, 0.1 ~ 30.0 is 1 value, 30.0 ~ 200.0 is 2 value, and 200.0 ~ 500.0 is 3 value.



4.3.6. Extraction

Extract the raster that matches a space or attribute filter condition.

4.3.6.1. Extract by Attributes

Extract the raster that matches a space or attribute filter condition.

■ Syntax

RasterExtraction (GridCoverage2D inputCoverage, Integer bandIndex, Filter filter):

GridCoverage2D

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The input raster from which cells will be extracted.	Complex	✓
bandIndex	The zero-based band index, default index is a 0.	Literal	-
filter	A logical expression that selects a subset of raster cells. ex> Value > 250	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- The field name of the **filter** parameter must be Value.
- The **filter** parameter can use a spatial filter.

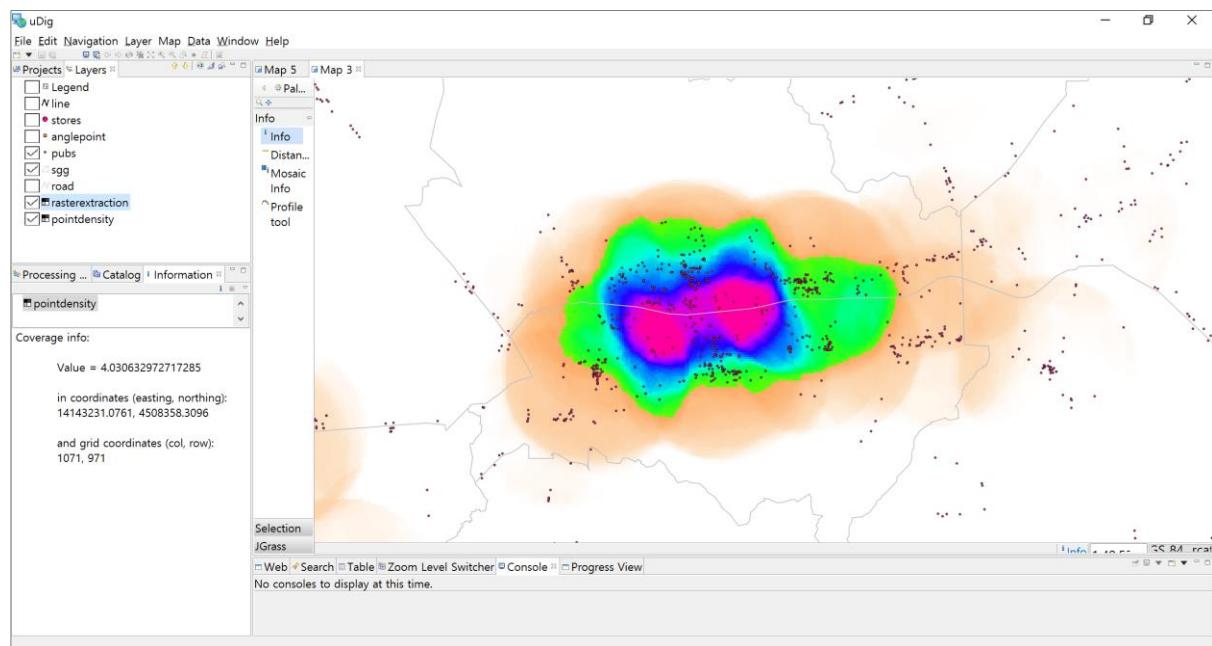
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
```

```
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:RasterExtraction</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputCoverage</ows:Identifier>
<wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
<wps:Body/>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>filter</ows:Identifier>
<wps:Data>
<wps:ComplexData mimeType="text/xml; subtype=filter/1.0"><![CDATA[Value >
250]]></wps:ComplexData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="image/tiff">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of extracting the area where the cell value is 250 or more($\text{value} \geq 250$) from the raster data.



4.3.6.2. Extract by Extent

Extract the intersecting raster data.

■ Syntax

RasterClipByExtent (GridCoverage2D inputCoverage, ReferencedEnvelope extent):

GridCoverage2D

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The input gridcoverage to be clipped.	Complex	✓
extent	The Reference envelope to clip gridcoverage.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- The **extent** parameter is of type BoundingBoxData and consists of crs, dimensions, LowerCorner, UpperCorner as follows.

```
<wps:Data>
  <wps:BoundingBoxData crs="EPSG:3857" dimensions="2">
    <ows:LowerCorner>0.0 0.0</ows:LowerCorner>
    <ows:UpperCorner>1.0 1.0</ows:UpperCorner>
  </wps:BoundingBoxData>
</wps:Data>
```

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
  xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
  xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
```

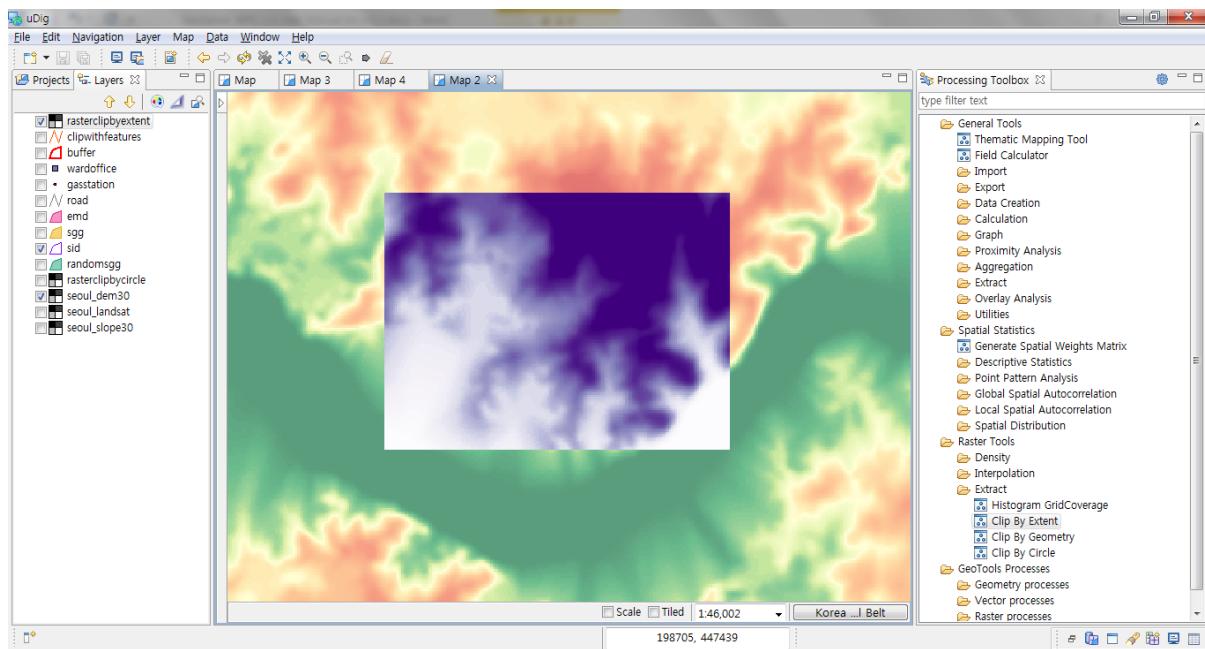
```

xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RasterClipByExtent</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputCoverage</ows:Identifier>
      <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
        <wps:Body>
          <wcs:GetCoverage service="WCS" version="1.1.1">
            <ows:Identifier>foss:seoul_dem30</ows:Identifier>
            <wcs:DomainSubset>
              <gml:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#5181">
                <ows:LowerCorner>179171.39881047895 436569.3290600816</ows:LowerCorner>
                <ows:UpperCorner>216221.0981287582 466869.08315843146</ows:UpperCorner>
              </gml:BoundingBox>
            </wcs:DomainSubset>
            <wcs:Output format="image/tiff"/>
          </wcs:GetCoverage>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>cropShape</ows:Identifier>
      <wps:Data>
        <wps:BoundingBoxData crs="EPSG:5181" dimensions="2">
          <ows:LowerCorner>196200.93382496 446742.832084541</ows:LowerCorner>
          <ows:UpperCorner>200948.405261965 450277.401141511</ows:UpperCorner>
        </wps:BoundingBoxData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="image/tiff">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of extracting the raster data by setting BoundingBox(MinX, MinY, MaxX, MaxY, CRS) area.



4.3.6.3. Extract by Geometry

Extract the intersecting raster data by setting Polygon Geometry.

■ Syntax

RasterClipByGeometry (GridCoverage2D inputCoverage, Geometry cropShape):

GridCoverage2D

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The input gridcoverage to be clipped.	Complex	✓
cropShape	The Polygon or MultiPolygon to clip gridcoverage.	Complex	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- The geometry type of **cropShape** must be Polygon or MultiPolygon features.

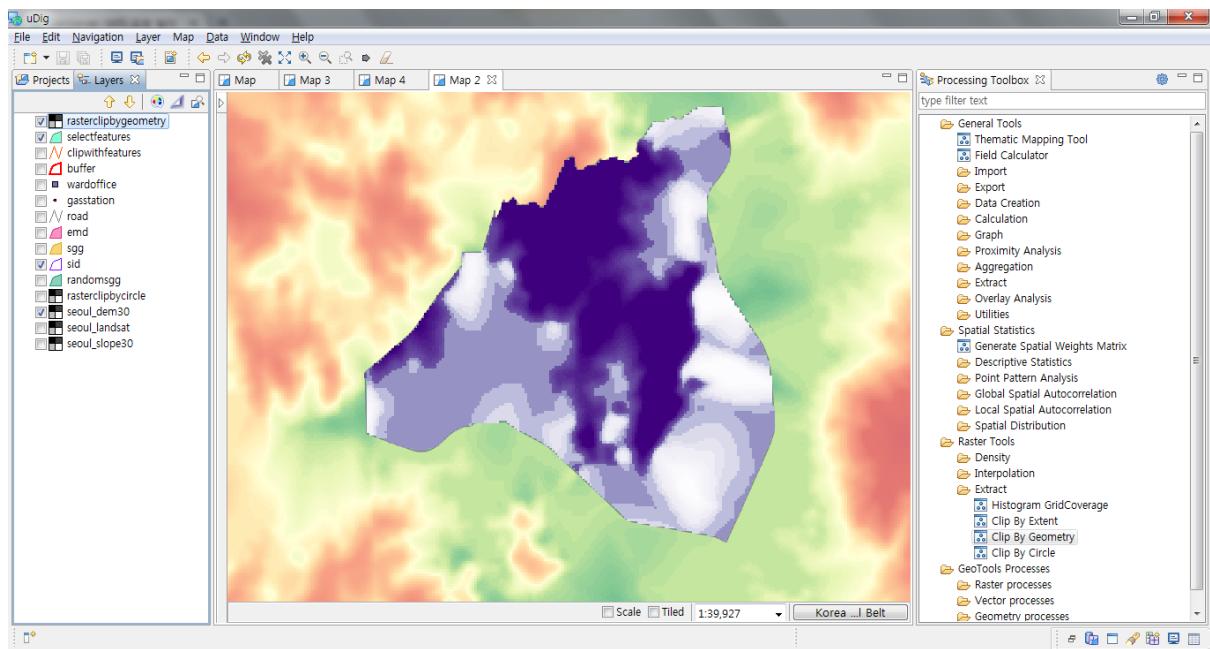
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
    <ows:Identifier>statistics:RasterClipByGeometry</ows:Identifier>
    <wps:DataInputs>
        <wps:Input>
            <ows:Identifier>inputCoverage</ows:Identifier>
            <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
                <wps:Body>
                    <wcs:GetCoverage service="WCS" version="1.1.1">
```

```
<ows:Identifier>foss:seoul_dem30</ows:Identifier>
<wcs:DomainSubset>
  <gml:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#5181">
    <ows:LowerCorner>179171.39881047895 436569.3290600816</ows:LowerCorner>
    <ows:UpperCorner>216221.0981287582 466869.08315843146</ows:UpperCorner>
  </gml:BoundingBox>
</wcs:DomainSubset>
<wcs:Output format="image/tiff"/>
</wcs:GetCoverage>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>cropShape </ows:Identifier>
  <wps:Data>
    <wps:ComplexData mimeType="application/wkt"><![CDATA[MULTIPOLYGON (((206338 456264,
....., 206338 456264))))]]></wps:ComplexData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="image/tiff">
    <ows:Identifier>result </ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of extracting the raster data, set by the Polygon or MultiPolygon Geometry.



4.3.6.4. Extract by Circle

Set the circle based on the center point and radius, extract it by the intersecting raster data.

■ Syntax

RasterClipByCircle (GridCoverage2D inputCoverage, Geometry center, Double radius, Boolean inside): GridCoverage2D

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The input gridcoverage to be clipped.	Complex	✓
center	The center point of the circle defining the area to be extracted.	Complex	✓
radius	Radius of the circle defining the area to be extracted.	Literal	✓
inside	Default is True	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- If the **inside** parameter is False, return the area excluding the circle.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RasterClipByCircle</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
```

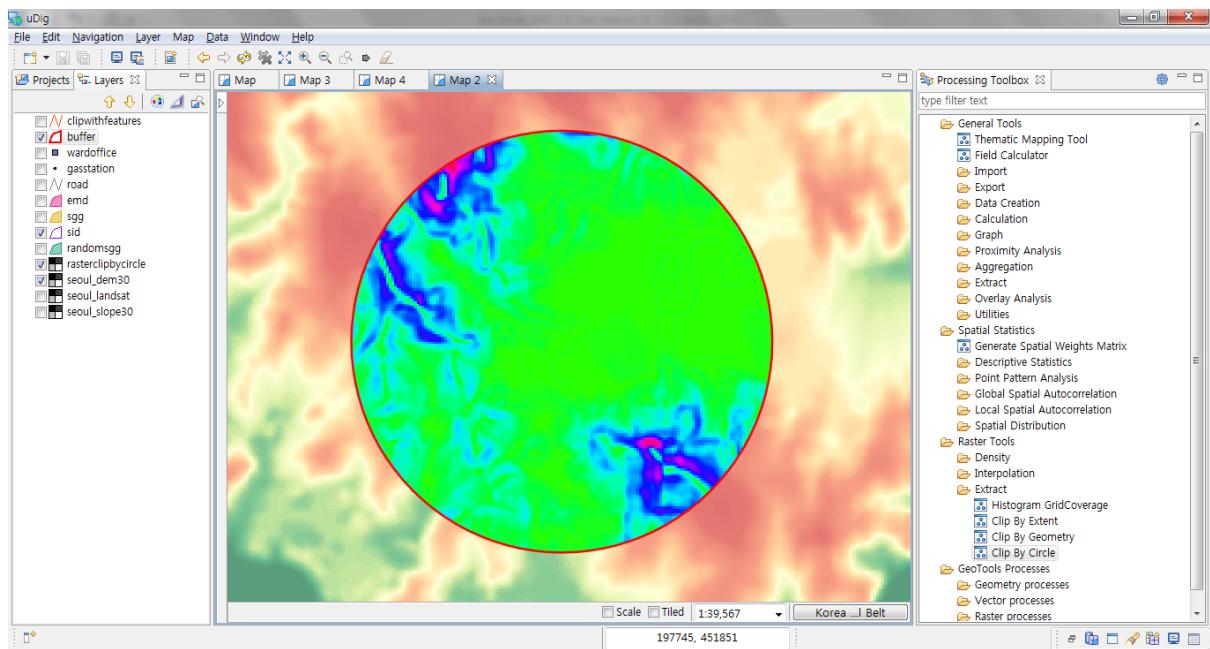
```

<ows:Identifier>inputCoverage</ows:Identifier>
<wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
  <wps:Body>
    <wcs:GetCoverage service="WCS" version="1.1.1">
      <ows:Identifier>foss:seoul_dem30</ows:Identifier>
      <wcs:DomainSubset>
        <gml:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#5181">
          <ows:LowerCorner>179171.39881047895 436569.3290600816</ows:LowerCorner>
          <ows:UpperCorner>216221.0981287582 466869.08315843146</ows:UpperCorner>
        </gml:BoundingBox>
      </wcs:DomainSubset>
      <wcs:Output format="image/tiff"/>
    </wcs:GetCoverage>
  </wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>center</ows:Identifier>
  <wps:Data>
    <wps:ComplexData mimeType="application/wkt"><![CDATA[POINT(197598
451746)]]></wps:ComplexData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>radius</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>1500</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="image/tiff">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of extracting the raster data, corresponding to a circle with a radius of 1500 meters around a certain point.



4.3.6.5. Conditional Expression

Converts the raster's cell value to a True or False value according to the filter conditions.

■ Syntax

RasterCon (GridCoverage2D inputCoverage, Integer bandIndex, Filter filter, Integer trueValue, Integer falseValue): GridCoverage2D

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The input gridcoverage to be clipped.	Complex	✓
bandIndex	The zero-based band index, default index is a 0.	Literal	-
filter	A logical expression that determines which of the input cells are to be true or false. ex> Value > 250	Complex	✓
trueValue	The input whose values will be used as the output cell values if the condition is true.	Literal	✓
falseValue	The input whose values will be used as the output cell values if the condition is false. Default is NoData.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- The field name of the **filter** parameter must be Value.
- The **trueValue** and **falseValue** parameter must be Integer values.
- If the **falseValue** parameter value is Null, apply the NoData value.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
```

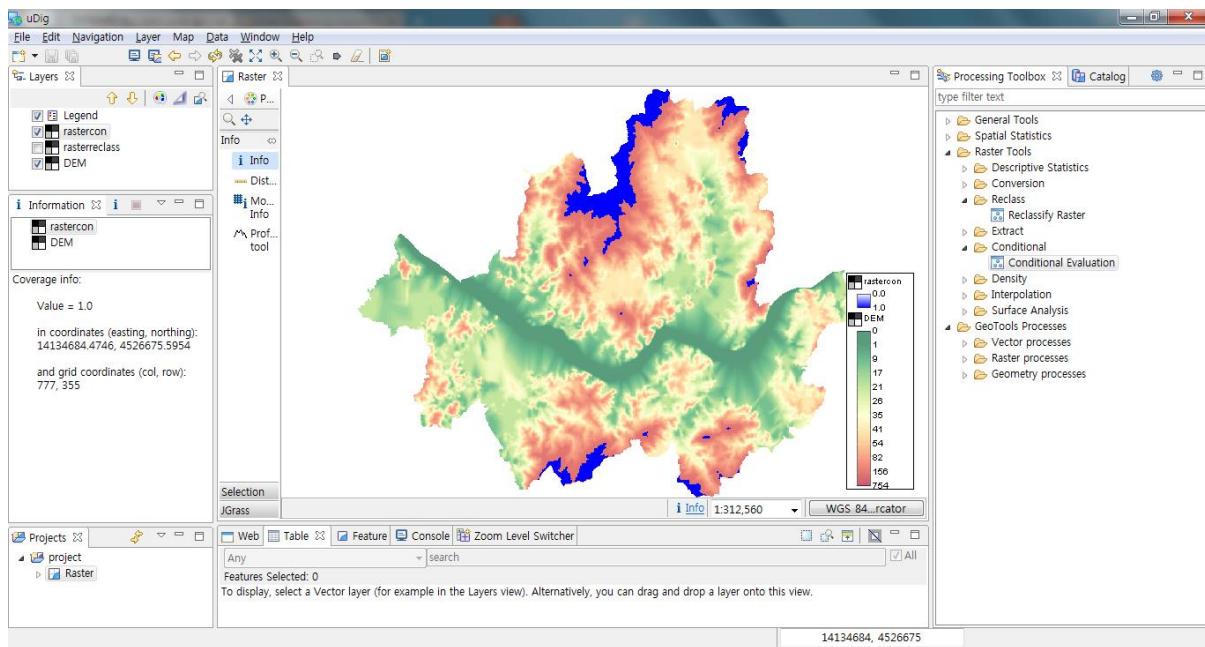
```

xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RasterCon</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputCoverage</ows:Identifier>
      <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
        <wps:Body>
          <wcs:GetCoverage service="WCS" version="1.1.1">
            <ows:Identifier>foss:seoul_dem30</ows:Identifier>
            <wcs:DomainSubset>
              <ows:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#3857">
                <ows:LowerCorner>1.4111343323506365E7 4498971.750719266</ows:LowerCorner>
                <ows:UpperCorner>1.4158021303411832E7 4537343.6431004135</ows:UpperCorner>
              </ows:BoundingBox>
            </wcs:DomainSubset>
            <wcs:Output format="image/tiff"/>
          </wcs:GetCoverage>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>filter</ows:Identifier>
      <wps:Data>
        <wps:ComplexData mimeType="text/plain; subtype=cql"><![CDATA[Value >
250]]></wps:ComplexData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>trueValue</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>1</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="image/tiff">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of extracting the raster data, corresponding to a circle with a radius of 1500 meters around a certain point.



4.3.7. Density

Perform density analysis of the raster.

4.3.7.1. Kernel Density

Perform Kernel Density analysis based on point features and various kernel functions.

■ Syntax

```
KernelDensity (SimpleFeatureCollection inputFeatures, KernelType kernelType, String
populationField, Double searchRadius, Double cellSize, ReferencedEnvelope extent);
GridCoverage2D
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input point features for which to calculate the density.	Complex	✓
kernelType	Kernel functions.	Literal	-

populationField	The field denoting population values for each feature.	Literal	-
searchRadius	The search radius within which to calculate density.	Literal	-
cellSize	The cell size for the output raster.	Literal	-
extent	The extent for the output raster.	Complex	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- Kernel Type⁵ consists of BINARY, COSINE, DISTANCE, EPANECHNIKOV, GAUSSIAN, INVERSE_DISTANCE, QUADRATIC, QUARTIC, TRIANGULAR, TRIWEIGHT, TRICUBE.
- Unless set the **extent** parameter, use the range of the **inputFeatures** layer.
- Unless set the **cellSize** parameter, choose the smaller value between Extent's Width and Height, and divide it by 250.
- Unless set the **searchRadius** parameter, choose the smaller value between Extent's Width and Height, and divide it by 30.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:KernelDensity</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
```

⁵ https://en.wikipedia.org/wiki/Kernel_%28statistics%29

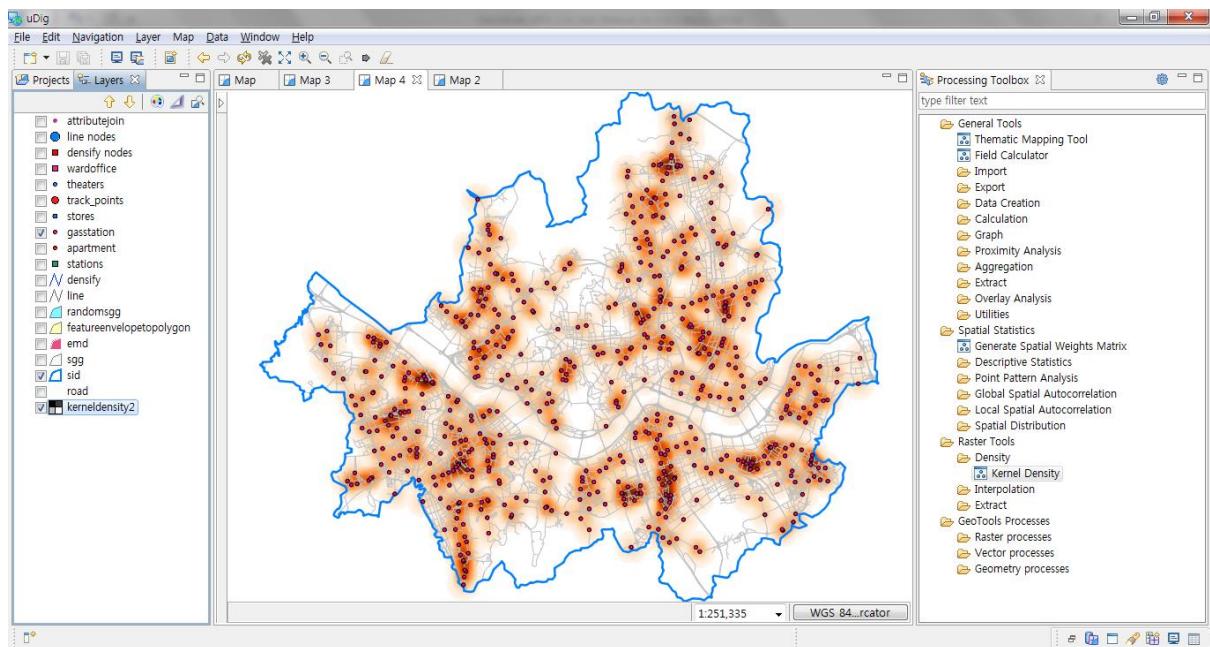
```

<wps:Body>
  <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
  xmlns:foss="http://www.opengeospatial.net/foss">
    <wfs:Query typeName="foss:gasstation"/>
  </wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>kernelType</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>QUADRATIC</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>cellSize</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>30</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>extent</ows:Identifier>
  <wps:Data>
    <wps:BoundingBoxData crs="EPSG:3857" dimensions="2">
      <ows:LowerCorner>1.4111357E7 4498975.0</ows:LowerCorner>
      <ows:UpperCorner>1.4158036E7 4537337.0</ows:UpperCorner>
    </wps:BoundingBoxData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="image/tiff">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of Kernel Density analysis of 30 meters cell size based on Seoul gas station data.



4.3.7.2. Point Density

Perform density analysis by setting point features and neighbors.

■ Syntax

```
PointDensity (SimpleFeatureCollection inputFeatures, String populationField, String neighborhood, Double cellSize, ReferencedEnvelope extent): GridCoverage2D
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input point features for which to calculate the density.	Complex	✓
populationField	The field denoting population values for each feature.	Literal	-
neighborhood	Neighborhood: Ex> Circle + Radius Ex> Rectangle + width + height	Literal	-
cellSize	The cell size for the output raster.	Literal	-
extent	The extent for the output raster.	Complex	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- Unless set the **neighborhood** parameter, use the Circle + radius (the width and height of the extent of **inPutFeatures**, whichever is smaller, divided by 30).
- Unless set the **extent** parameter, use the range of the inputFeatures layer.
- Unless set the **cellSize** parameter, choose the smaller value between Extent's Width and Height, and divide it by 250.

■ Request Examples

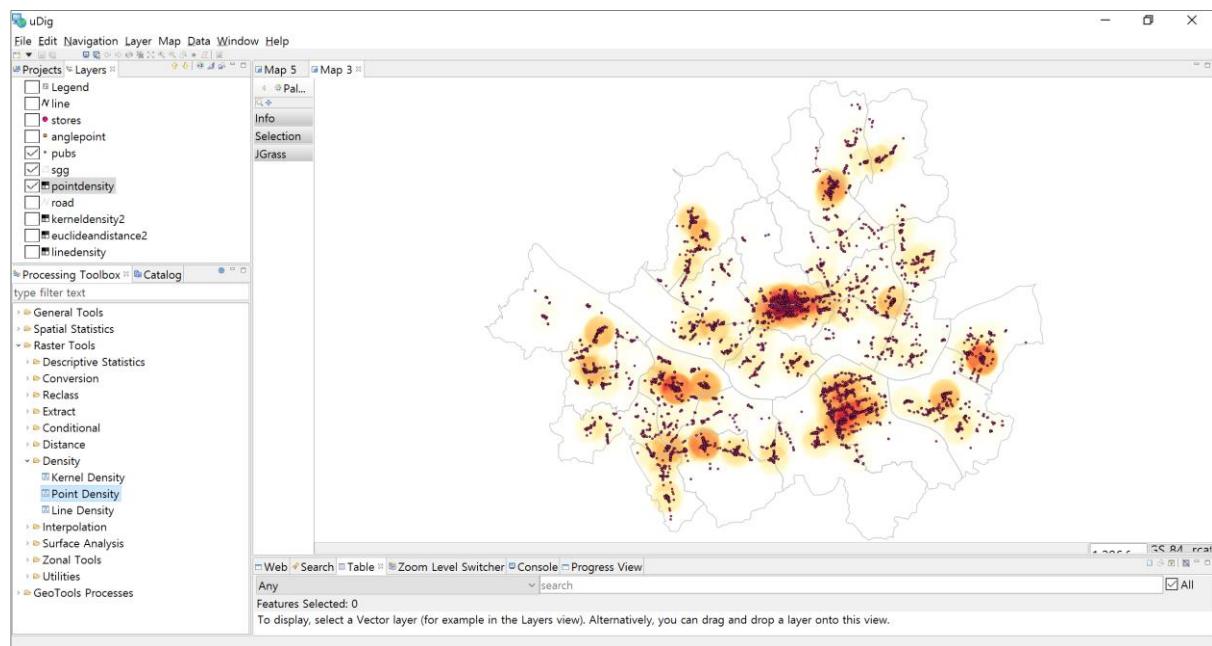
```

<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:PointDensity</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:pubs"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>cellSize</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>30</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="image/tiff">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of Kernel Density analysis of 30 meters cell size based on Seoul point data.



4.3.7.3. Line Density

Performs line density analysis using line features and search radius.

■ Syntax

```
LineDensity (SimpleFeatureCollection inputFeatures, String populationField, Double  
searchRadius, Double cellSize, ReferencedEnvelope extent): GridCoverage2D
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input line features for which to calculate the density.	Complex	✓
populationField	The field denoting population values for each feature.	Literal	-
searchRadius	The search radius within which to calculate density.	Literal	-
cellSize	The cell size for the output raster.	Literal	-
extent	The extent for the output raster.	Complex	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- Unless set the **searchRadius** parameter, choose the smaller value between Extent's Width and Height, and divide it by 30.
- Unless set the **extent** parameter, use the range of the **inputFeatures** layer.
- Unless set the **cellSize** parameter, choose the smaller value between Extent's Width and Height, and divide it by 250.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
```

```

xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:LineDensity</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:road"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>cellSize</ows:Identifier>
<wps>Data>
<wps:LiteralData>30</wps:LiteralData>
</wps>Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="image/tiff">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of Line Density analysis of 30 meters cell size based on Seoul major road data.



4.3.8. Interpolation

Perform interpolation analysis, using points and attribute values.

4.3.8.1. IDW (Inverse Distance Weighted)

Perform the Inverse Distance Weighted(IDW) Interpolation analysis using the point feature layers.

■ Syntax

```
IDW (SimpleFeatureCollection inputFeatures, String inputField, Double power, RadiusType  
radiusType, Integer numberOfPoints, Double distance, Double cellSize,  
ReferencedEnvelope extent): GridCoverage2D
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input point features for which to calculate the density.	Complex	✓
inputField	The field that holds a height or magnitude value for each point.	Literal	✓
power	The exponent (default 2.0) of distance.	Literal	-
radiusType	The search radius type Variable, Fixed	Literal	-
numberOfPoints	The numberOfPoints is an integer value specifying the number of nearest input sample points to be used to perform the interpolation.	Literal	-
distance	The distance specifies the distance, in map units, by which to limit the search for the nearest input sample points.	Literal	-
cellSize	The cell size for the output raster.	Literal	-
extent	The extent for the output raster.	Complex	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- Unless set the Extent parameter, use the range of the inputFeatures layer.
- Unless set the **cellSize** parameter, choose the smaller value between Extent's Width and Height, and divide it by 250.

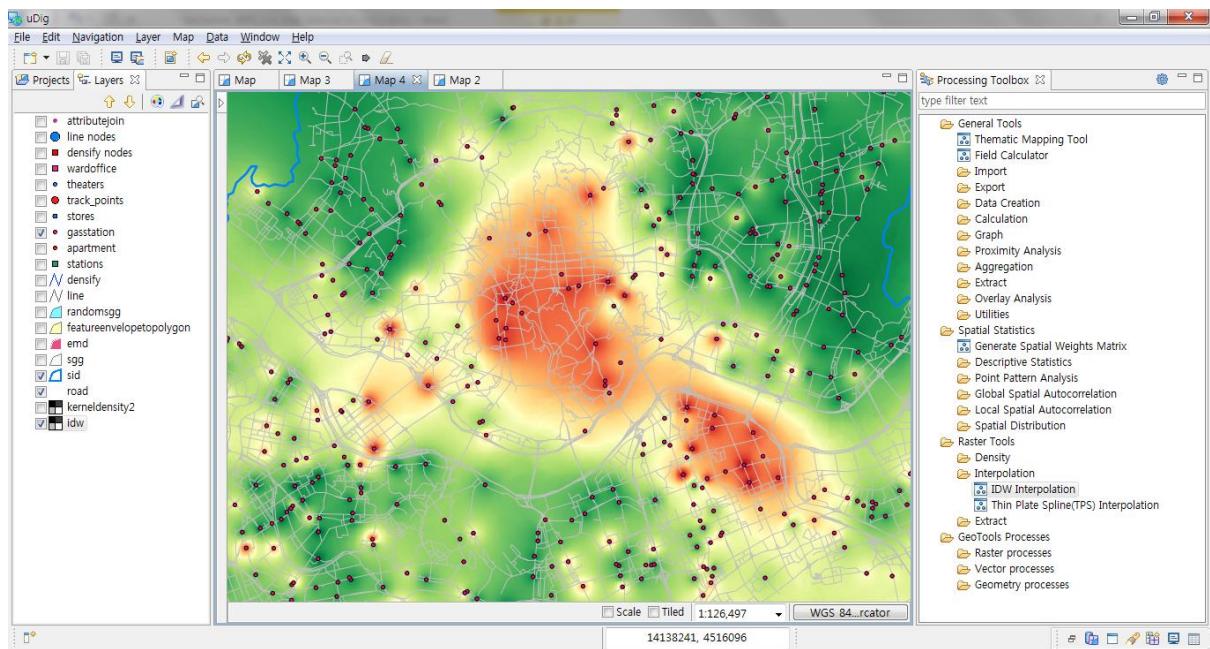
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:IDW</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
        xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:gasstation"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>inputField</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>price</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>power</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>2.0</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>radiusType</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>Variable</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
</wps:Execute>
```

```
</wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>numberOfPoints</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>24</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>cellSize</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>30</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>extent</ows:Identifier>
  <wps:Data>
    <wps:BoundingBoxData crs="EPSG:3857" dimensions="2">
      <ows:LowerCorner>1.4111357E7 4498975.0</ows:LowerCorner>
      <ows:UpperCorner>1.4158036E7 4537337.0</ows:UpperCorner>
    </wps:BoundingBoxData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="image/tiff">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of Inverse Distance Weighted(IDW) analysis of 30 meter cell size based on oil price information of Seoul gas station.



4.3.8.2. TPS (Thin Plate Spline)

Performs Thin Plate Spline(TPS) interpolation analysis using the point feature layers.

■ Syntax

TPS (SimpleFeatureCollection inputFeatures, String inputField, Double cellSize, ReferencedEnvelope extent): GridCoverage2D

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input point features for which to calculate the density.	Complex	✓
inputField	The field that holds a height or magnitude value for each point.	Literal	✓
cellSize	The cell size for the output raster.	Literal	-
extent	The extent for the output raster.	Complex	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- Unless set the **extent** parameter, use the range of the **inputFeatures** layer.
- Unless set the **cellSize** parameter, choose the smaller value between Extent's Width and Height, and divide it by 250.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
```

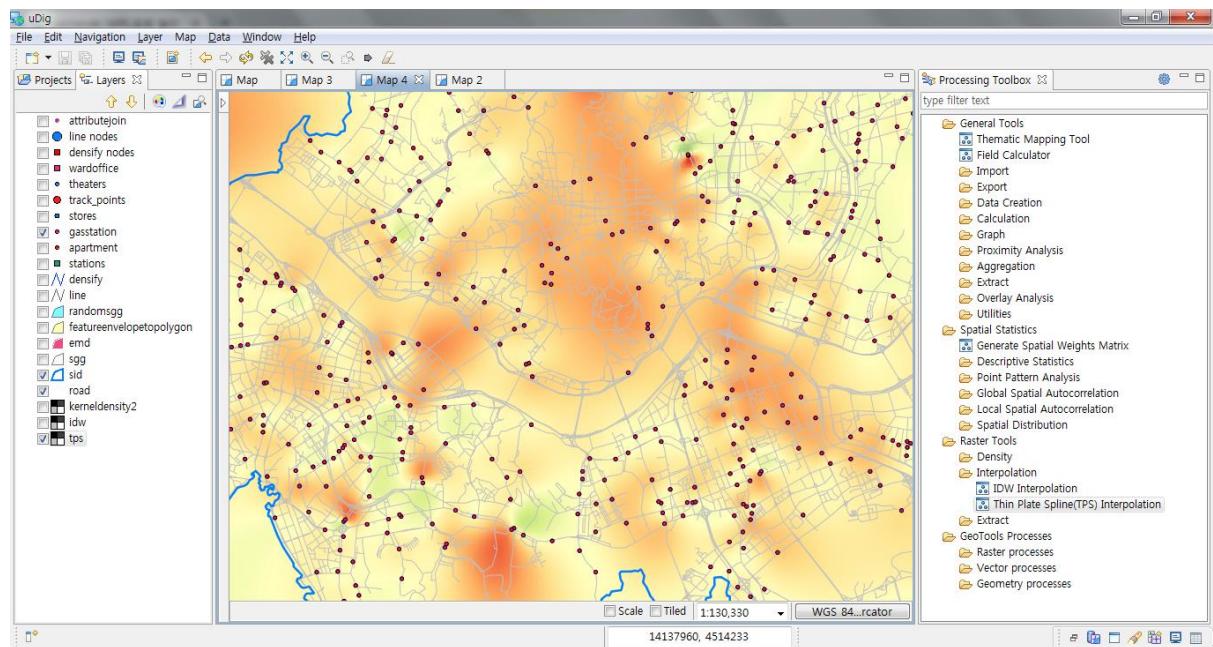
```

<ows:Identifier>statistics:TPS</ows:Identifier>
<wps:DataInputs>
    <wps:Input>
        <ows:Identifier>inputFeatures</ows:Identifier>
        <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
            <wps:Body>
                <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
                    <wfs:Query typeName="foss:gasstation"/>
                </wfs:GetFeature>
            </wps:Body>
        </wps:Reference>
    </wps:Input>
    <wps:Input>
        <ows:Identifier>inputField</ows:Identifier>
        <wps:Data>
            <wps:LiteralData>price</wps:LiteralData>
        </wps:Data>
    </wps:Input>
    <wps:Input>
        <ows:Identifier>cellSize</ows:Identifier>
        <wps:Data>
            <wps:LiteralData>30</wps:LiteralData>
        </wps:Data>
    </wps:Input>
    <wps:Input>
        <ows:Identifier>extent</ows:Identifier>
        <wps:Data>
            <wps:BoundingBoxData crs="EPSG:3857" dimensions="2">
                <ows:LowerCorner>1.4111357E7 4498975.0</ows:LowerCorner>
                <ows:UpperCorner>1.4158036E7 4537337.0</ows:UpperCorner>
            </wps:BoundingBoxData>
        </wps:Data>
    </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
    <wps:RawDataOutput mimeType="image/tiff">
        <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of Thin Plate Spline(TPS) analysis of 30 meter cell size based on oil price information of Seoul gas station.



4.3.9. Surface Analysis

Perform the terrain analysis.

4.3.9.1. Raster Profile

Converts the raster data such as DEM and line layers to point data after longitudinal section(Profile) analysis.

■ Syntax

RasterProfile (GridCoverage2D inputCoverage, Geometry userLine, Double interval):
SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The input surface raster.	Complex	✓
userLine	LineString or MultiLineString geometry.	Literal	✓
interval	The interval of distance. Default = length of geometry / 20.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output line features.	Complex	✓

■ Constraints

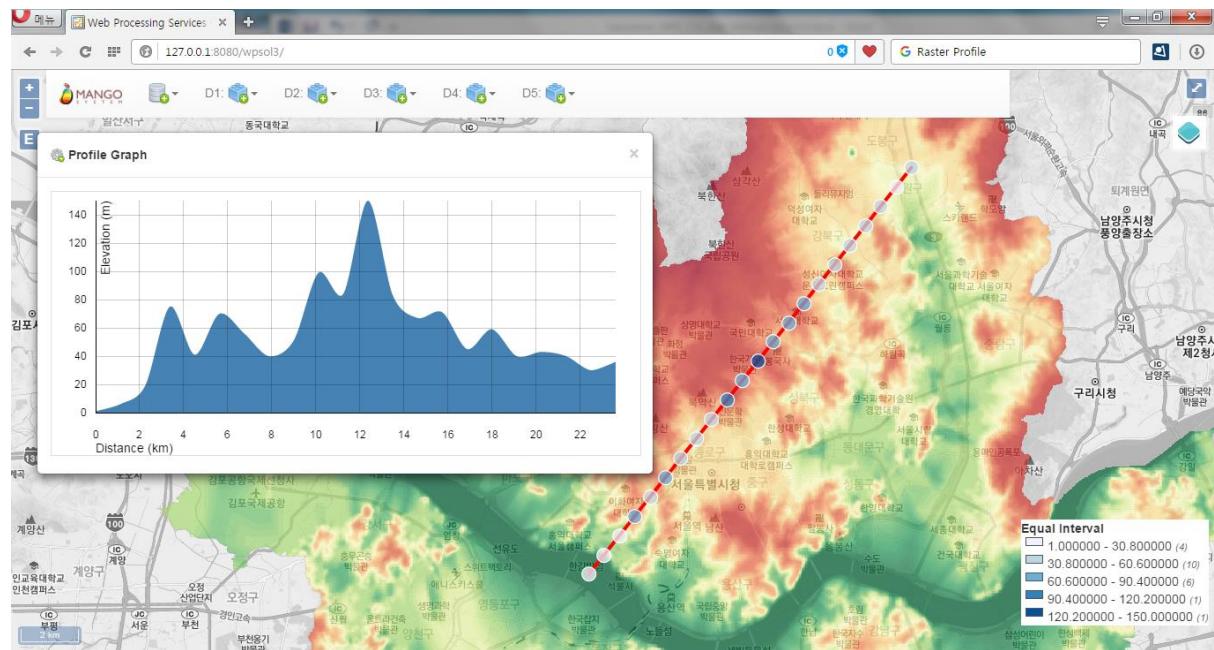
- Unless set the **interval** parameter, apply the value that divided the **userLine** length by 20.
- The Output point layers contain the distance(cumulative distance) and value (cell value of the raster, such as height value) field.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RasterProfile</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputCoverage</ows:Identifier>
      <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
        <wps:Body>
          <wcs:GetCoverage service="WCS" version="1.1.1">
            <ows:Identifier>foss:seoul_dem30</ows:Identifier>
            <wcs:DomainSubset>
              <ows:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#3857">
                <ows:LowerCorner>1.4111343323506365E7 4498971.750719266</ows:LowerCorner>
                <ows:UpperCorner>1.4158021303411832E7 4537343.6431004135</ows:UpperCorner>
              </ows:BoundingBox>
            </wcs:DomainSubset>
            <wcs:Output format="image/tiff"/>
          </wcs:GetCoverage>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>userLine</ows:Identifier>
      <wps:Data>
        <wps:ComplexData mimeType="application/wkt"><![CDATA[LineString(14130049 4513932,
14144040 4531525)]]></wps:ComplexData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of Profile analysis as point layers and graph using Seoul DEM.



4.3.9.2. Radial Line Of Sight

Perform Radial Line Of Sight analysis using observation points and radius using DEM raster data.

■ Syntax

```
RadialLineOfSight (GridCoverage2D inputCoverage, Geometry observerPoint, Double  
observerOffset, Double radius, Integer sides, Boolean useCurvature, Boolean  
useRefraction, Double refractionFactor): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input surface raster.	Complex	✓
observerPoint	The observer's coordinate.	Complex	✓
observerOffset	The observer's offset above the surface raster. The default is 0.0 units.	Literal	✓
radius	The radius from the observer point, for which the radial visibility will be calculated.	Literal	✓
sides	The number of sides. The default sides is 180.	Literal	-
useCurvature	Indicates whether the earth's curvature should be taken into consideration for the line-of-sight analysis. Default is False.	Literal	-
useRefraction	Indicates whether atmospheric refraction should be taken into consideration when generating a line of sight from a functional surface. Default is False.	Literal	-
refractionFactor	The refraction factor. The default refraction factor is 0.13.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- The **useRefraction** and **refractionFactor** parameter are only applied if the **useCurvature** parameter is True.

- If the **useCurvature** parameter is True and the **useRefraction** parameter is False, then **refractionFactor** applies a value of 0.13.
- The output line layers includes the Angle, Visible field, the visible field if the Visible field value is 1, and the invisible area if it is 0.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RadialLineOfSight</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputCoverage</ows:Identifier>
      <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
        <wps:Body>
          <wcs:GetCoverage service="WCS" version="1.1.1">
            <ows:Identifier>foss:seoul_dem30</ows:Identifier>
            <wcs:DomainSubset>
              <ows:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#3857">
                <ows:LowerCorner>1.4111343323506365E7 4498971.750719266</ows:LowerCorner>
                <ows:UpperCorner>1.4158021303411832E7 4537343.6431004135</ows:UpperCorner>
              </ows:BoundingBox>
            </wcs:DomainSubset>
            <wcs:Output format="image/tiff"/>
          </wcs:GetCoverage>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>observerPoint</ows:Identifier>
      <wps:Data>
        <wps:ComplexData mimeType="application/wkt"><![CDATA[POINT(14136287.706512472
4516237.6022168035)]]></wps:ComplexData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
```

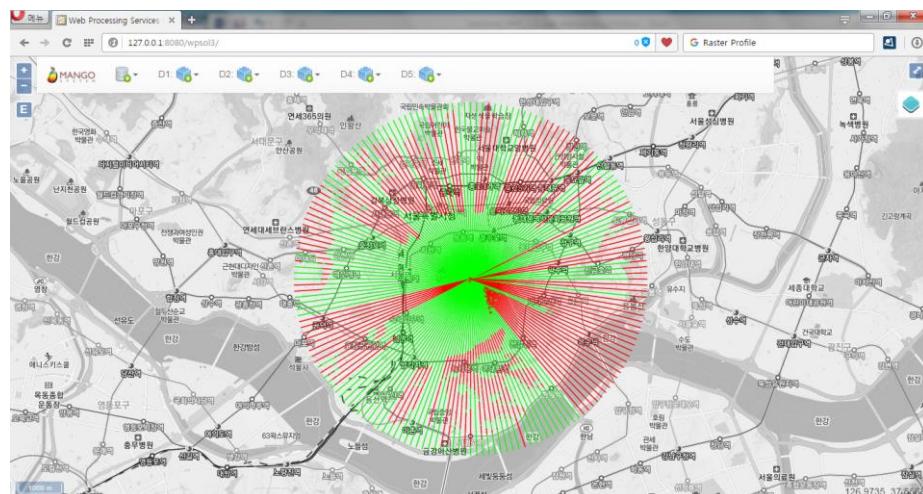
```

<ows:Identifier>observerOffset</ows:Identifier>
<wps:Data>
  <wps:LiteralData>1.8</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>radius</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>5000</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of visible area analysis of 5000m radius based on Namsan of Seoul DEM.



4.3.9.3. Linear Line Of Sight

Performs Linear Line Of Sight analysis using observation points and target points using DEM raster data.

■ Syntax

```
LinearLineOfSight (GridCoverage2D inputCoverage, Geometry observerPoint, Double  
observerOffset, Geometry targetPoint, Boolean useCurvature, Boolean useRefraction,  
Double refractionFactor): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The input surface raster.	Complex	✓
observerPoint	The observer's coordinate.	Complex	✓
observerOffset	The observer's offset above the surface rater. The default is 0.0 units.	Literal	✓
targetPoint	The target's coordinate.	Complex	✓
useCurvature	Indicates whether the earth's curvature should be taken into consideration for the line-of-sight analysis. Default is False.	Literal	-
useRefraction	Indicates whether atmospheric refraction should be taken into consideration when generating a line of sight from a functional surface. Default is False.	Literal	-
refractionFactor	The refraction factor. The default refraction factor is 0.13.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- If the **useCurvature** parameter is True, the **useRefraction** and **refractionFactor** parameter are only applied.

- If the **useCurvature** parameter is True and the **useRefraction** parameter is False, then **refractionFactor** applies a value of 0.13.
- The output line layers include a visible field, a visible area when the Visible field value is 1, and a non-visible area when the value is 0.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:LinearLineOfSight</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputCoverage</ows:Identifier>
      <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
        <wps:Body>
          <wcs:GetCoverage service="WCS" version="1.1.1">
            <ows:Identifier>foss:seoul_dem30</ows:Identifier>
            <wcs:DomainSubset>
              <ows:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#3857">
                <ows:LowerCorner>1.4111343323506365E7 4498971.750719266</ows:LowerCorner>
                <ows:UpperCorner>1.4158021303411832E7 4537343.6431004135</ows:UpperCorner>
              </ows:BoundingBox>
            </wcs:DomainSubset>
            <wcs:Output format="image/tiff"/>
          </wcs:GetCoverage>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>observerPoint</ows:Identifier>
      <wps:Data>
        <wps:ComplexData mimeType="application/wkt"><![CDATA[POINT(14136291.5572
4516245.7128)]]></wps:ComplexData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
```

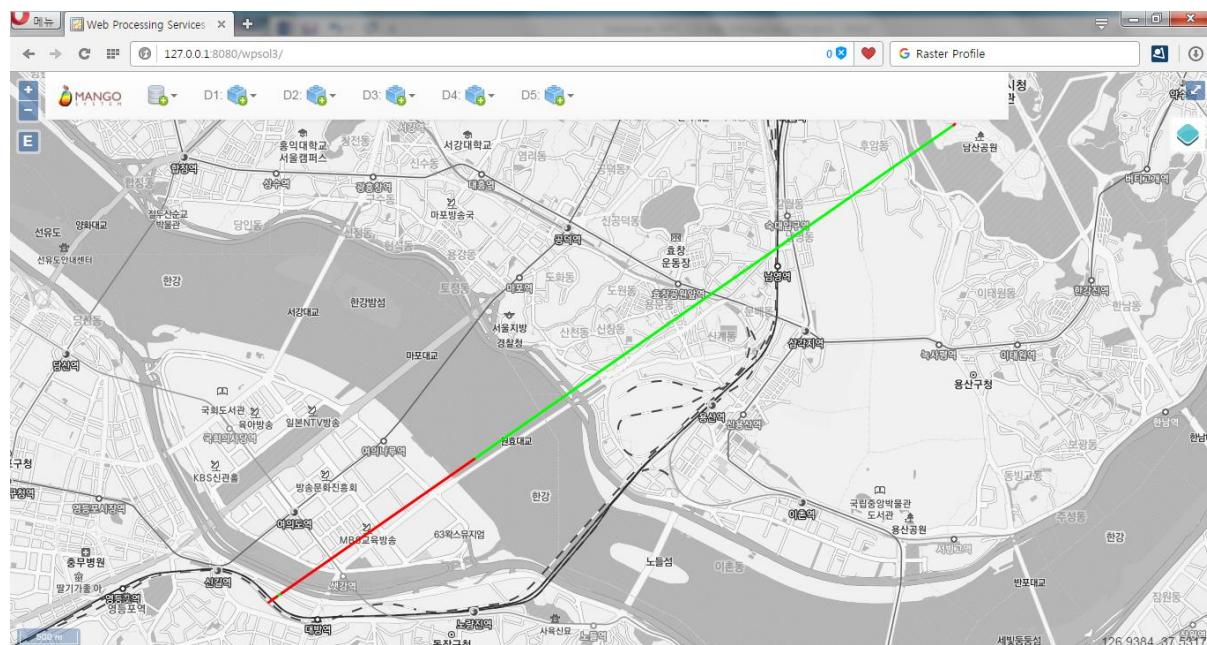
```

<ows:Identifier>observerOffset</ows:Identifier>
<wps:Data>
  <wps:LiteralData>1.8</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>targetPoint</ows:Identifier>
  <wps:Data>
    <wps:ComplexData mimeType="application/wkt"><![CDATA[POINT(14128854.8064
4511075.0345)]]></wps:ComplexData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of analyzing the line of sight of Yeouido, based on Namsan of Seoul DEM.



4.3.9.4. Find Highest/Lowest Points

Find the highest and lowest positions in a specific area of raster data, such as DEM, and return to the point.

■ Syntax

RasterHighLowPoints (GridCoverage2D inputCoverage, Integer bandIndex, Geometry cropShape, HighLowType valueType): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The input gridcoverage to be processed.	Complex	✓
bandIndex	The zero-based band index, default index is a 0.	Literal	-
cropShape	The Polygon or MultiPolygon to clip gridcoverage.	Complex	-
valueType	Value Type (Both, High, Low). Default is High.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Raster.	Complex	✓

■ Constraints

- The **cropShape** parameter must be Polygon or MultiPolygon.
- The **valueType** parameter uses the values of Both, High, and Low.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RasterHighLowPoints</ows:Identifier>
  <wps:DataInputs>
```

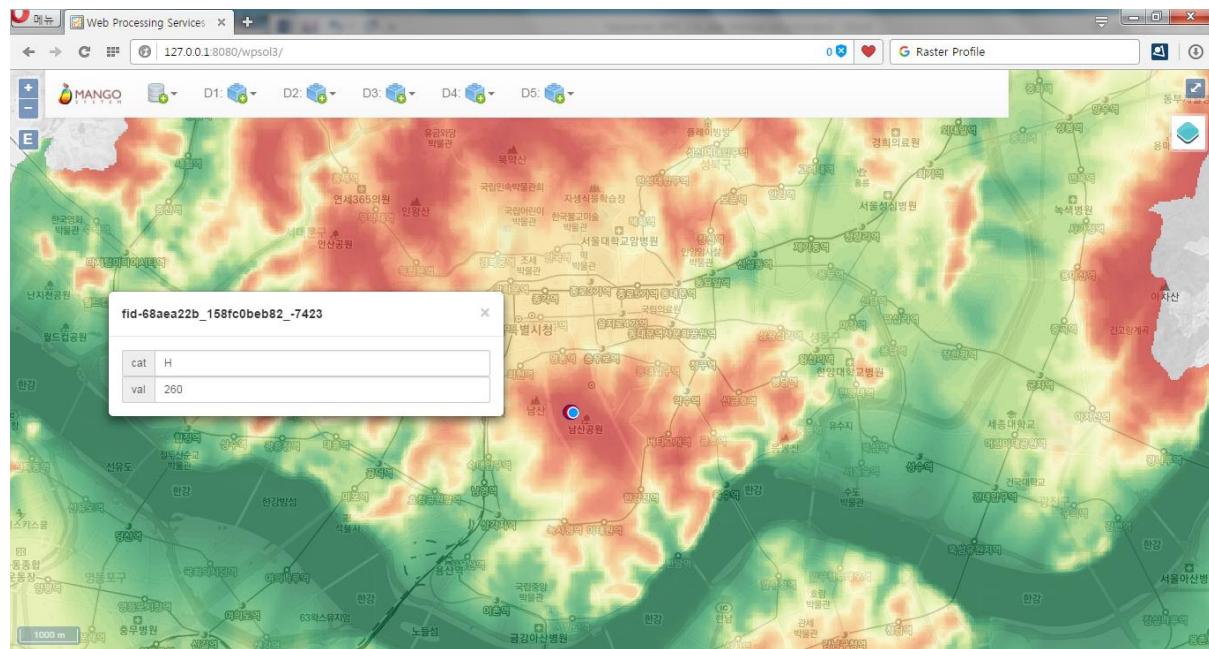
```

<wps:Input>
  <ows:Identifier>inputCoverage</ows:Identifier>
  <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
    <wps:Body>
      <wcs:GetCoverage service="WCS" version="1.1.1">
        <ows:Identifier>foss:seoul_dem30</ows:Identifier>
        <wcs:DomainSubset>
          <ows:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#3857">
            <ows:LowerCorner>1.4111343323506365E7 4498971.750719266</ows:LowerCorner>
            <ows:UpperCorner>1.4158021303411832E7 4537343.6431004135</ows:UpperCorner>
          </ows:BoundingBox>
        </wcs:DomainSubset>
        <wcs:Output format="image/tiff"/>
      </wcs:GetCoverage>
    </wps:Body>
  </wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>valueType</ows:Identifier>
  <wps>Data>
    <wps:LiteralData>Both</wps:LiteralData>
  </wps>Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the analysis result of the highest point based on the current map range of Seoul DEM.



4.3.10. Zonal

Performs the zonal statistics.

4.3.10.1. *Zonal Statistics*

Calculates statistics on the values of raster data for each area of section data.

■ Syntax

ZonalStatistics (SimpleFeatureCollection zoneFeatures, GridCoverage2D inputCoverage, Integer bandIndex): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
zoneFeatures	The Dataset (polygon features) that defines the zones.	Complex	✓
targetField	The output field to be calculated.	Literal	-
valueCoverage	The Raster that contains the values on which to calculate a statistic.	Complex	✓
bandIndex	The zero-based band index, default index is a 0.	Literal	-
statisticsType	Zonal statisticx type to be calculated.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- The **zoneFeatures** parameter must be Polygon or MultiPolygon type.
- Unless set the **targetField** parameter, use val value as default.
- If there are multiple raster bands, the **bandIndex** uses 0 value as default.
- The **statisticsType** parameter can use the following options.

Option	Description
Count	Number of cells
Sum	Sum of cells
Mean	Mean of cells (Default)
Minimum	Minimum of cells
Maximum	Maximum of cells
Range	Range of cells
StdDev	Standard deviation

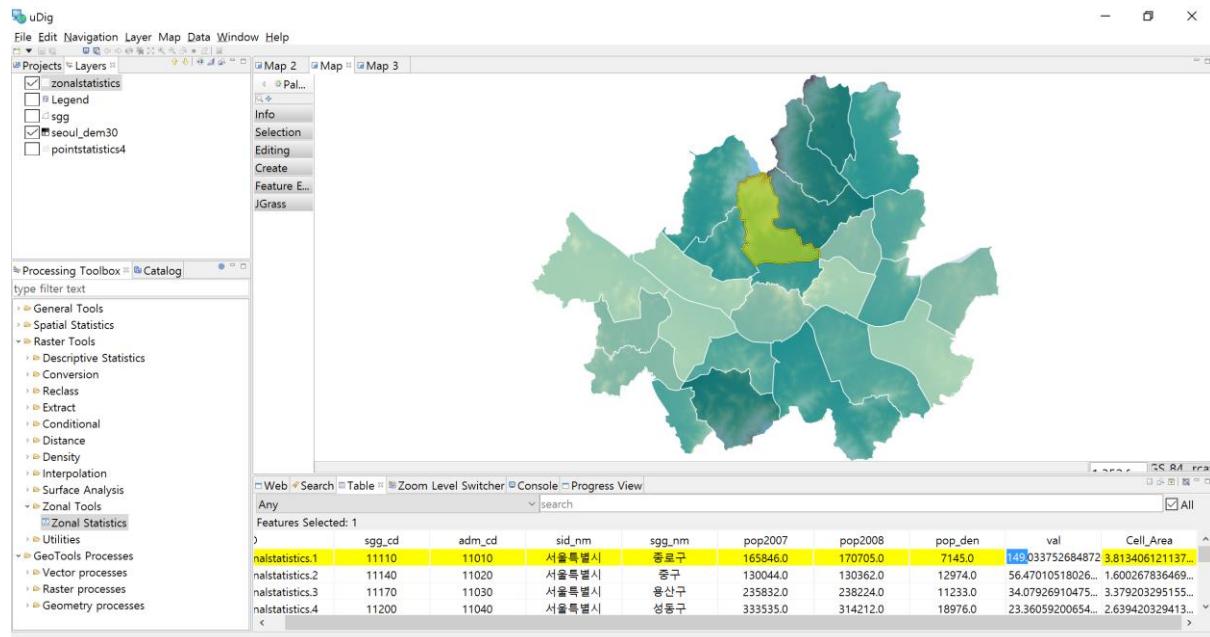
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:ZonalStatistics</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>zoneFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
<wfs:Query typeName="foss:ssg"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>targetField</ows:Identifier>
<wps:Data>
<wps:LiteralData>h_mean</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>valueCoverage</ows:Identifier>
<wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
<wps:Body>
<wcs:GetCoverage service="WCS" version="1.1.1">
<ows:Identifier>foss:seoul_dem30</ows:Identifier>
```

```
<wcs:DomainSubset>
  <ows:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#3857">
    <ows:LowerCorner>1.4111343323506365E7 4498971.750719266</ows:LowerCorner>
    <ows:UpperCorner>1.4158021303411832E7 4537343.6431004135</ows:UpperCorner>
  </ows:BoundingBox>
</wcs:DomainSubset>
<wcs:Output format="image/tiff"/>
</wcs:GetCoverage>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>bandIndex</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>0</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>statisticsType</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>Mean</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the analysis result of average elevation by administrative boundary(Si-Gun-Gu) using Seoul DEM.



4.3.11. Projection

Define the coordinate system, transforming and changing cell size, of raster data.

4.3.11.1. *Resample*

Change the resolution (cell size) of the raster data.

■ Syntax

```
RasterResample (GridCoverage2D inputCoverage, Double cellSize, ResampleType resamplingType): GridCoverage2D
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The raster dataset for which you want to change the spatial resolution.	Complex	✓
cellSize	The cell size for the output raster.	Literal	✓
resamplingType	The resampling algorithm to be used. Ex> NEAREST (default), BILINEAR, BICUBIC.	Complex	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output raster.	Complex	✓

■ Constraints

- The **resamplingType** parameter can use the NEAREST(default), BILINEAR, and BICUBIC options.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
```

```
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:RasterResample </ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputCoverage </ows:Identifier>
    <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
      <wps:Body>
        <wcs:GetCoverage service="WCS" version="1.1.1">
          <ows:Identifier>foss:seoul_dem30 </ows:Identifier>
          <wcs:DomainSubset>
            <ows:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#3857">
              <ows:LowerCorner>1.4111343323506365E7 4498971.750719266 </ows:LowerCorner>
              <ows:UpperCorner>1.4158021303411832E7 4537343.6431004135 </ows:UpperCorner>
            </ows:BoundingBox>
          </wcs:DomainSubset>
          <wcs:Output format="image/tiff"/>
        </wcs:GetCoverage>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>cellSize </ows:Identifier>
    <wps>Data>
      <wps:LiteralData>50 </wps:LiteralData>
    </wps>Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>resamplingType </ows:Identifier>
    <wps>Data>
      <wps:LiteralData>NEAREST </wps:LiteralData>
    </wps>Data>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="image/tiff">
    <ows:Identifier>result </ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

4.3.11.2. Redefine Projection

Define or redefine the coordinate system of the raster data with a new coordinate system.

■ Syntax

RasterForceCRS (GridCoverage2D inputCoverage, CoordinateReferenceSystem forcedCRS):
GridCoverage2D

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The raster dataset for which you want to redefine the coordinate reference system.	Complex	✓
forcedCRS	Coordinate reference system to use for input raster dataset.	Literal	✓

■ Process Outputs

Identifier	Description	Type	Required
Result	Output raster.	Complex	✓

■ Constraints

- The **forcedCRS** parameter must be in the [EPSG: code number] format.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RasterForceCRS</ows:Identifier>
  <wps:DataInputs>
```

```
<wps:Input>
  <ows:Identifier>inputCoverage</ows:Identifier>
  <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
    <wps:Body>
      <wcs:GetCoverage service="WCS" version="1.1.1">
        <ows:Identifier>foss:seoul_dem30</ows:Identifier>
        <wcs:DomainSubset>
          <ows:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#3857">
            <ows:LowerCorner>1.4111343323506365E7 4498971.750719266</ows:LowerCorner>
            <ows:UpperCorner>1.4158021303411832E7 4537343.6431004135</ows:UpperCorner>
          </ows:BoundingBox>
        </wcs:DomainSubset>
        <wcs:Output format="image/tiff"/>
      </wcs:GetCoverage>
    </wps:Body>
  </wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>forcedCRS</ows:Identifier>
  <wps>Data>
    <wps:LiteralData>EPSG:5181</wps:LiteralData>
  </wps>Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="image/tiff">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of redefining the original Daegu EPSG:5181 DEM as EPSG:5186.



4.3.11.3. Reproject

Converts the coordinate system of raster data.

■ Syntax

RasterReproject (GridCoverage2D inputCoverage, CoordinateReferenceSystem targetCRS, ResampleType resamplingType, Double cellSize, CoordinateReferenceSystem forcedCRS): GridCoverage2D

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputCoverage	The raster dataset for which you want to change the spatial reference system.	Complex	✓
targetCRS	Target coordinate reference system to use for reprojection.	Literal	✓
resamplingType	The resampling algorithm to be used. NEAREST (default), BILINEAR, BICUBIC.	Literal	-
cellSize	The cell size for the output raster.	Literal	-

forcedCRS	Coordinate reference system to use for input raster dataset.	Literal	-
------------------	--	---------	---

■ Process Outputs

Identifier	Description	Type	Required
result	Output raster.	Complex	✓

■ Constraints

- The **targetCRS** and **forcedCRS** parameter must be [EPSG: code number] format.
- The **resamplingType** parameter can use the NEAREST (default), BILINEAR, BICUBIC options.
- Unless set the **cellSize** parameter, the original cell size is used. If converting from the geographic coordinate system to the projection coordinate system, the converted cell size is applied.
- If the **forcedCRS** parameter is set, converting the original raster data to the defined coordinate system.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:RasterReproject</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputCoverage</ows:Identifier>
      <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
        <wps:Body>
          <wcs:GetCoverage service="WCS" version="1.1.1">
            <ows:Identifier>foss:seoul_dem30</ows:Identifier>
```

```
<wcs:DomainSubset>
  <ows:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#3857">
    <ows:LowerCorner>1.4111343323506365E7 4498971.750719266</ows:LowerCorner>
    <ows:UpperCorner>1.4158021303411832E7 4537343.6431004135</ows:UpperCorner>
  </ows:BoundingBox>
</wcs:DomainSubset>
<wcs:Output format="image/tiff"/>
</wcs:GetCoverage>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>targetCRS</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>EPSG:5181</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>resamplingType</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>NEAREST</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>cellSize</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>50</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="image/tiff">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

4.4. Spatial Statistics Analysis

A process group associated with spatial statistical analysis.

4.4.1. Descriptive

Calculates the statistical information using Geometry or attribute value of field.

4.4.1.1. Basic Statistics

Perform basic statistical analysis based on field values in the feature layers.

■ Syntax

```
StatisticsFeatures (SimpleFeatureCollection inputFeatures, String inputFields, String  
caseField): DataStatisticsResult
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The input features containing the field(s) that will be used to calculate statistics.	Complex	✓
inputFields	Single field or comma (,) separated numeric field(s) containing attribute values used to calculate the specified statistic.	Literal	✓
caseField	The field used to group features for separate statistics calculations.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output Statistics.	Complex	✓

■ Constraints

- If the **caseField** parameter is set, statistical information is generated for each unique value of the **caseField**.
- Output is returned in XML format.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:StatisticsFeatures</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
        xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:korea_sgg"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>inputFields</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>a3_2005</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>caseField</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>sid_nm</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

■ Response

This is the result of analyzed basic statistic by administrative boundary(Si-Gun-Gu) using the a3_2000 field value of the national Si-Gun-Gu administrative district is converted XML format.

```
<?xml version="1.0" encoding="utf-8"?>
<DataStatistics>
  <Item>
    <TypeName>korea_sgg</TypeName>
    <CaseValue>강원도</CaseValue>
    <PropertyName>a3_2000</PropertyName>
    <Count>18</Count>
    <InvalidCount>0</InvalidCount>
    <Minimum>0.24774</Minimum>
    <Maximum>7.81668</Maximum>
    <Range>7.56894</Range>
    <Ranges>0.24774 - 7.81668</Ranges>
    <Sum>79.64533043000002</Sum>
    <Mean>4.424740579444445</Mean>
    <Variance>6.976857255428096</Variance>
    <StandardDeviation>2.641374122578643</StandardDeviation>
    <CoefficientOfVariance>0.5969557028607279</CoefficientOfVariance>
  </Item>
  <Item>
    <TypeName>korea_sgg</TypeName>
    <CaseValue>경기도</CaseValue>
    <PropertyName>a3_2000</PropertyName>
    <Count>31</Count>
    <InvalidCount>0</InvalidCount>
    <Minimum>0.0</Minimum>
    <Maximum>15.46253</Maximum>
    <Range>15.46253</Range>
    <Ranges>0.0 - 15.46253</Ranges>
    <Sum>271.03358996</Sum>
    <Mean>8.74301903096774</Mean>
    <Variance>21.209124717119646</Variance>
    <StandardDeviation>4.605336547649872</StandardDeviation>
    <CoefficientOfVariance>0.5267444267635456</CoefficientOfVariance>
  </Item>
</DataStatistics>
```

4.4.1.2. Pearson Correlation Coefficient

Calculates the Pearson's Correlation Coefficient using two or more attribute fields.

■ Syntax

Pearson (SimpleFeatureCollection inputFeatures, String inputFields): PearsonResult

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features to be calculated.	Complex	✓
inputFields	The comma separated numeric field(s) containing attribute values used to calculate the specified statistic.	Literal	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Result Pearson Correlation Coefficient.	Complex	✓

■ Constraints

- Output is returned in XML format.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
    <ows:Identifier>statistics:Pearson</ows:Identifier>
    <wps:DataInputs>
        <wps:Input>
            <ows:Identifier>inputFeatures</ows:Identifier>
            <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
                <wps:Body>
```

```

<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
    <wfs:Query typeName="foss:korea_sgg"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
    <ows:Identifier>inputFields</ows:Identifier>
<wps:Data>
    <wps:LiteralData>pop2008, pop_den</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml">
        <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

This is the result of analyzing Pearson correlation, using two fields of the administrative boundary(Si-Gun-Gu). Ouput is converted XML format.

```

<?xml version="1.0" encoding="utf-8"?>
<PearsonResult>
    <PropertyName Name="pop2008">
        <Item Name="pop2008">
            <Value>1.0</Value>
        </Item>
        <Item Name="pop_den">
            <Value>0.3002549407911261</Value>
        </Item>
    </PropertyName>
    <PropertyName Name="pop_den">
        <Item Name="pop2008">
            <Value>0.3002549407911261</Value>
        </Item>
        <Item Name="pop_den">
            <Value>1.0</Value>
        </Item>
    </PropertyName>

```

```
</PropertyName>  
</PearsonResult>
```

4.4.1.3. Standardized Score of Dissimilarity

Calculates the standardized score of dissimilarity(SSD, degree of concentration), using the two attribute field values.

■ Syntax

StandardizedScores (SimpleFeatureCollection inputFeatures, Expression xField, Expression yField, String targetField): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features for which the standardized score of dissimilarity will be calculated.	Complex	✓
xField	X Value Field.	Literal	✓
yField	Y Value Field.	Literal	✓
targetField	Target Field. std_scr is default.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Unless set the **targetField** parameter, the std_scr is default.,

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
  xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
  xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
  xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
  xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
  http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:StandardizedScores</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
```

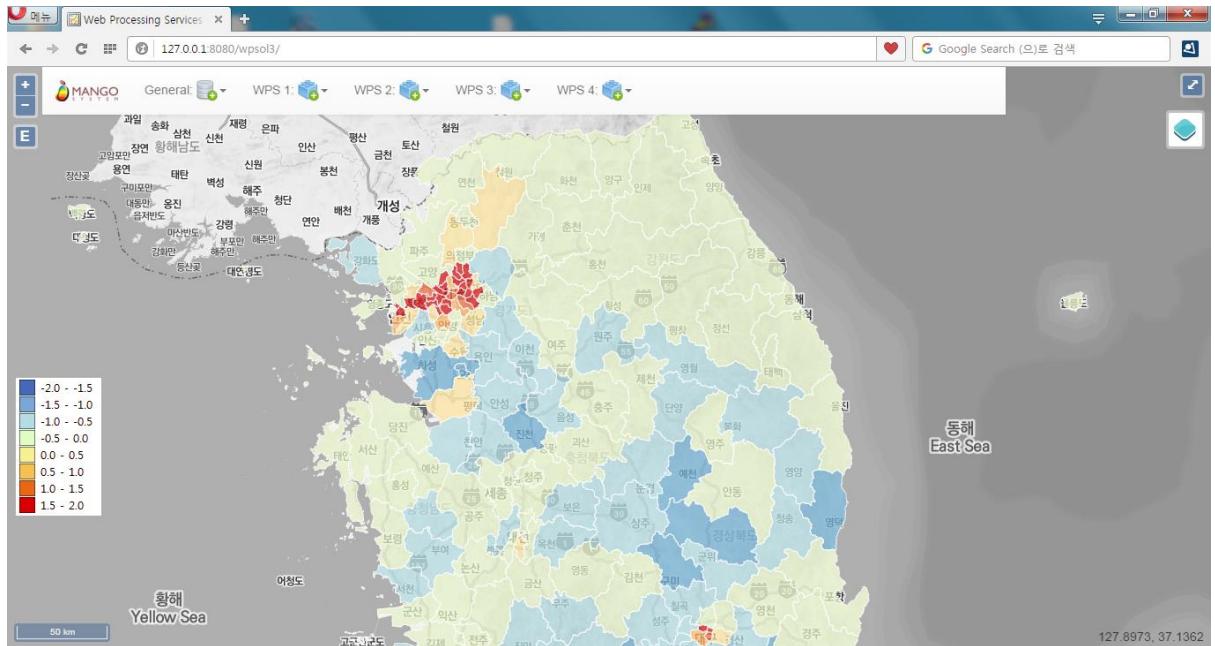
```

<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
  <wfs:Query typeName="foss:korea_sgg"/>
  </wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>xField</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>a0_2005</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>yField</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>a3_2005</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>targetField</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>std_scr</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1" >
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
■

```

■ Response

The following figure shows the result of the SSD analysis using the two fields of the administrative boundary(Si-Gun-Gu). Using the property values of targetField, visualize the SSD.



4.4.1.4. Focal Location Quotients

Calculates the Focal Location Quotients(FLQ, degree of specialization) using two attribute field values.

■ Syntax

FocalLQ (SimpleFeatureCollection inputFeatures, String fieldName1, String fieldName2, Double searchDistance): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features for which the focal LQ will be calculated.	Complex	✓
fieldName1	X Value Field.	Literal	✓
fieldName2	Y Value Field.	Literal	✓
searchDistance	The maximum search distance.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- The fields of the Output layer return values of flq, flqd, and fz.

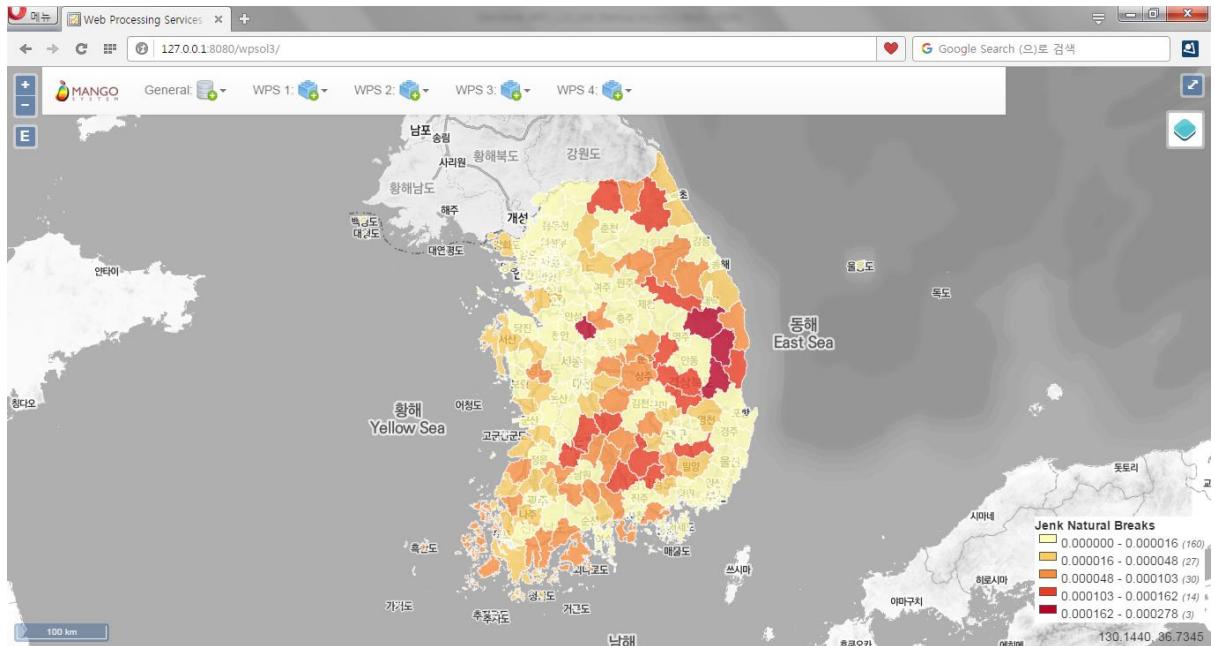
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:FocalLQ</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
```

```
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
  <wps:Body>
    <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
      xmlns:foss="http://www.opengeospatial.net/foss">
      <wfs:Query typeName="foss:korea_sgg"/>
    </wfs:GetFeature>
  </wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>xField</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>a0_2005</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>yField</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>a3_2005</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the visualized result of analyzing the FLQ, using the two fields of the administrative boundary(Si-Gun-Gu).



4.4.2. Distributions

Analyze the distribution of vector data patterns.

4.4.2.1. Mean Center

Returns the geographic center or center of concentration for all features in the feature layer.

■ Syntax

```
MeanCenter (SimpleFeatureCollection inputFeatures, String weightField, String caseField,  
String dimensionField): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	A features for which the mean center will be calculated.	Complex	✓
weightField	The numeric field used to create a weighted mean center.	Literal	-
caseField	The field used to group features for separate mean center calculations.	Literal	-
dimensionField	A numeric field containing attribute values from which an average value will be calculated.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Calculates using Centroid of **inputFeatures**.

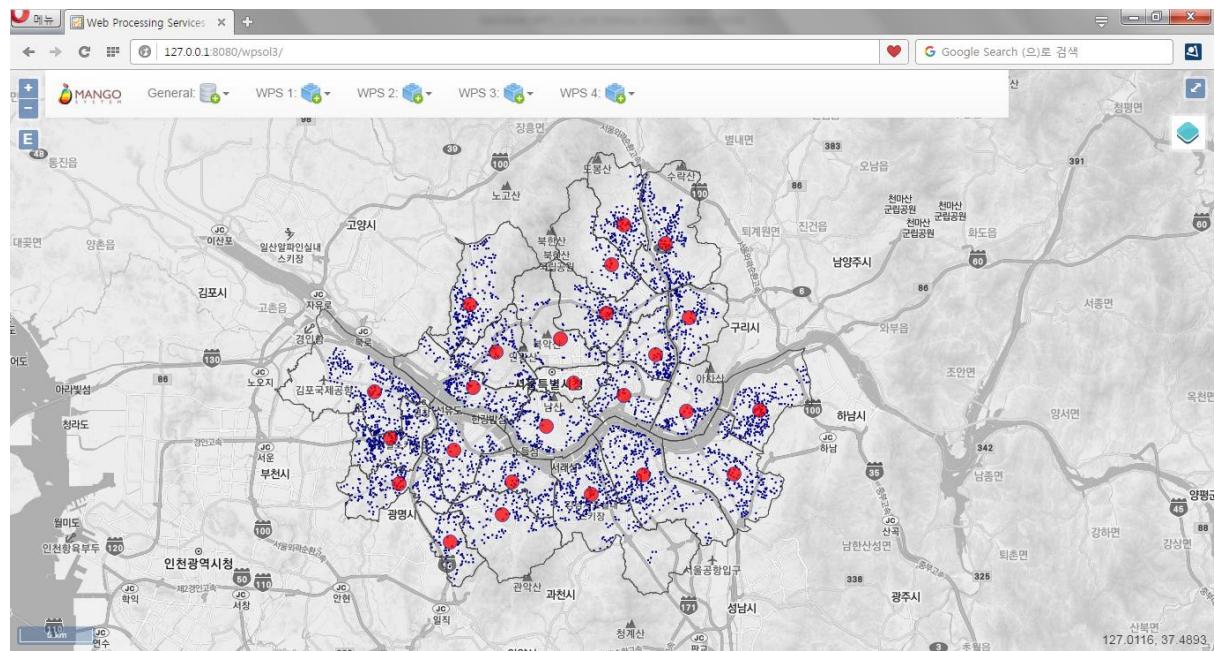
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"  
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
```

```
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:MeanCenter</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3">
            xmlns:foss="http://www.opengeospatial.net/foss">
              <wfs:Query typeName="foss:apartment"/>
            </wfs:GetFeature>
          </wps:Body>
        </wps:Reference>
      </wps:Input>
      <wps:Input>
        <ows:Identifier>caseField</ows:Identifier>
        <wps:Data>
          <wps:LiteralData>sgg_nm</wps:LiteralData>
        </wps:Data>
      </wps:Input>
    </wps:DataInputs>
    <wps:ResponseForm>
      <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
        <ows:Identifier>result</ows:Identifier>
      </wps:RawDataOutput>
    </wps:ResponseForm>
  </wps:Execute>
```

■ Response

The following figure shows the result of analyzing the Mean Center about the distribution of Seoul apartment by administrative boundary(Si-Gun-Gu).



4.4.2.2. Median Center

Returns the point at which the sum of the total distances is the smallest(Median Center) for all features in the feature layers.

■ Syntax

MedianCenter (SimpleFeatureCollection inputFeatures, String weightField, String caseField, String attributeFields): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	A features for which the median center will be calculated.	Complex	✓
weightField	The numeric field used to create a weighted median center.	Literal	-
caseField	The field used to group features for separate median center calculations.	Literal	-
attributeFields	(Comma separated) Numeric field(s) for which the data median value will be computed.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Calculate using Centroid of **inputFeatures**.

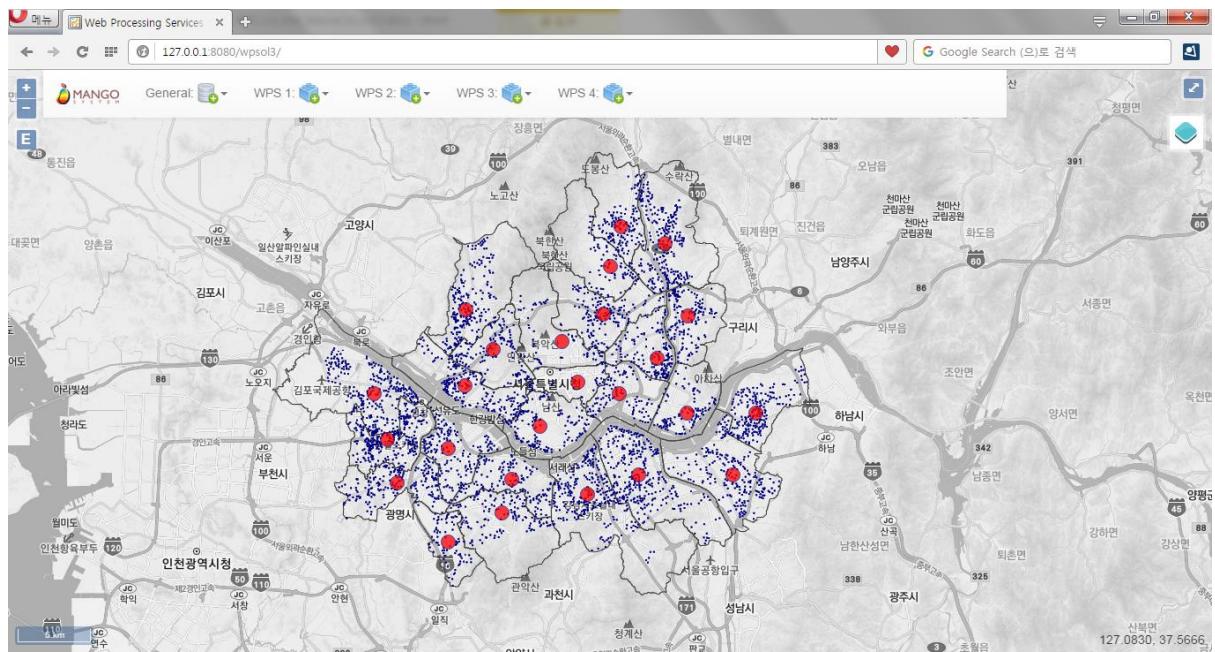
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
```

```
<ows:Identifier>statistics:MedianCenter</ows:Identifier>
<wps:DataInputs>
    <wps:Input>
        <ows:Identifier>inputFeatures</ows:Identifier>
        <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
            <wps:Body>
                <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
                    <wfs:Query typeName="foss:apartment"/>
                </wfs:GetFeature>
            </wps:Body>
        </wps:Reference>
    </wps:Input>
    <wps:Input>
        <ows:Identifier>caseField</ows:Identifier>
        <wps:Data>
            <wps:LiteralData>sgg_nm</wps:LiteralData>
        </wps:Data>
    </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
        <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

The following figure shows the result of analyzing the Median Center about the distribution of Seoul apartment by administrative boundary(Si-Gun-Gu)..



4.4.2.3. Central Feature

Returns the Central Feature with the smallest sum of the total distances for all features in the feature layer.

■ Syntax

```
CentralFeature (SimpleFeatureCollection inputFeatures, DistanceMethod distanceMethod,  
String weightField, String selfPotentialWeightField, String caseField):
```

```
SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features containing a distribution of features from which to identify the most centrally located feature.	Complex	✓
distanceMethod	Specifies how distances are calculated from each feature to neighboring features. Euclidean (default) or Manhattan.	Literal	-
weightField	The numeric field used to weight distances in the origin-destination distance matrix.	Literal	-
selfPotentialWeightField	The field representing self-potential. The distance or weight between a feature and itself.	Literal	-
caseField	The field used to group features for separate central feature computations.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Calculates using Centroid of **inputFeatures**.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
```

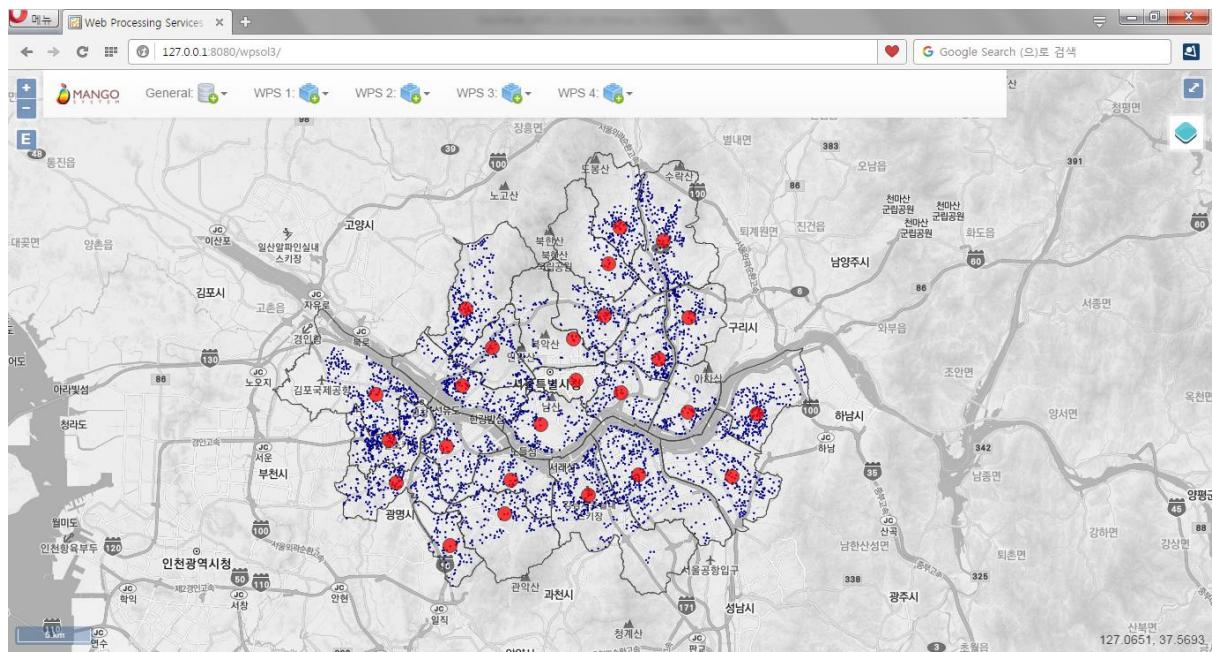
```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:CentralFeature</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:apartment"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>distanceMethod</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>Euclidean</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>caseField</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>sgg_nm</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of analyzing the Central Feature about the distribution of Seoul apartment by administrative boundary(Si-Gun-Gu).



4.4.2.4. Standard Distance

Measures the extent to which all features in the feature layers are centered or scattered by the Mean Center.

■ Syntax

```
StandardDistance (SimpleFeatureCollection inputFeatures, String circleSize, String weightField, String caseField): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features containing a distribution of features for which the standard deviational ellipse will be calculated.	Complex	✓
circleSize	The size (1, 2, 3) of output circles in standard deviations.	Literal	-
weightField	The numeric field used to weight locations according to their relative importance.	Literal	-
caseField	The field used to group features for separate standard distance calculations.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Calculate using Centroid of **inputFeatures**.
- The Circle generally includes features of 68% for 1_Standard_Deviation, 95% for 2_Standard_Deviation, and 99% for 3_Standard_Deviation.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
```

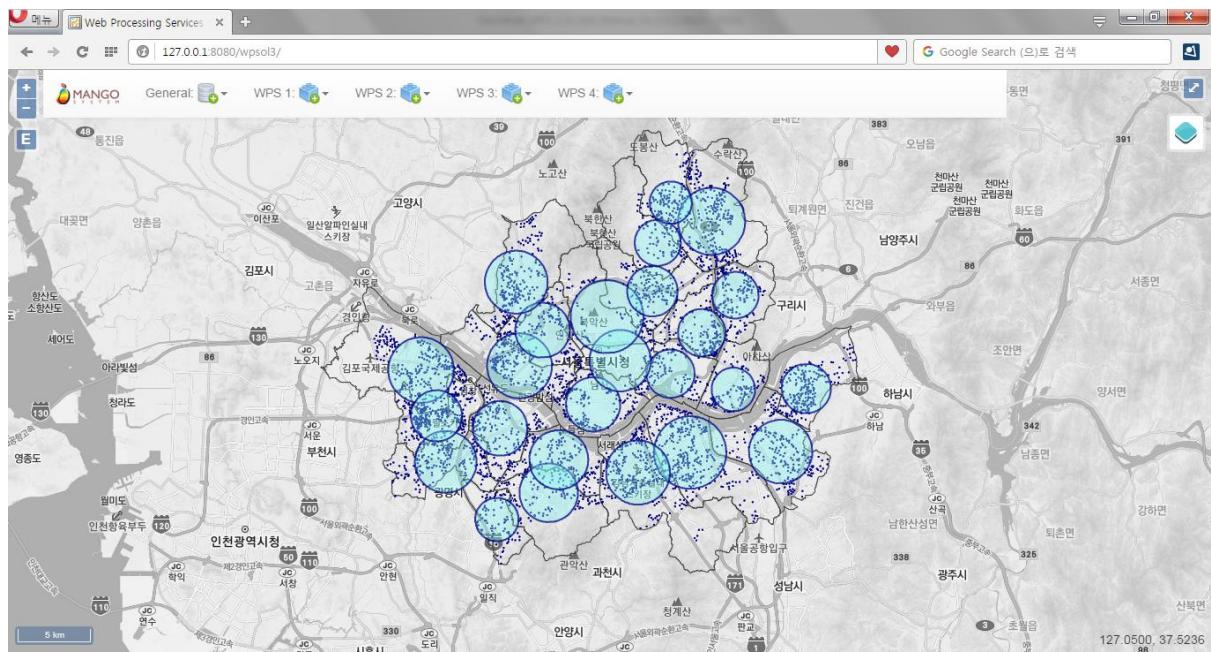
```

xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:StandardDistance</ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3">
<xmlns:foss="http://www.opengeospatial.net/foss">
  <wfs:Query typeName="foss:apartment"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>circleSize</ows:Identifier>
  <wps>Data>
    <wps:LiteralData>1_Standard_Deviation</wps:LiteralData>
  </wps>Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>caseField</ows:Identifier>
  <wps>Data>
    <wps:LiteralData>sgg_nm</wps:LiteralData>
  </wps>Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of analyzing the Standard Distance about the distribution of Seoul apartment by administrative boundary(SiGun-Gu).



4.4.2.5. Standard Deviational Ellipse

Measures the degree which all features of the feature layer are concentrated or centered by the Mean Center and the orientation of the distribution.

■ Syntax

StandardDeviationalEllipse (SimpleFeatureCollection inputFeatures, String ellipseSize, String weightField, String caseField): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features containing a distribution of features for which the standard deviational ellipse will be calculated.	Complex	✓
ellipseSize	The size (1, 2, 3) of output ellipses in standard deviations.	Literal	-
weightField	The numeric field used to weight locations according to their relative importance.	Literal	-
caseField	The field used to group features for separate directional distribution calculations.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Calculates using Centroid of **inputFeatures**.
- The Ellipse typically includes features of 68% for 1_Standard_Deviation, 95% for 2_Standard_Deviation, and 99% for 3_Standard_Deviation.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
```

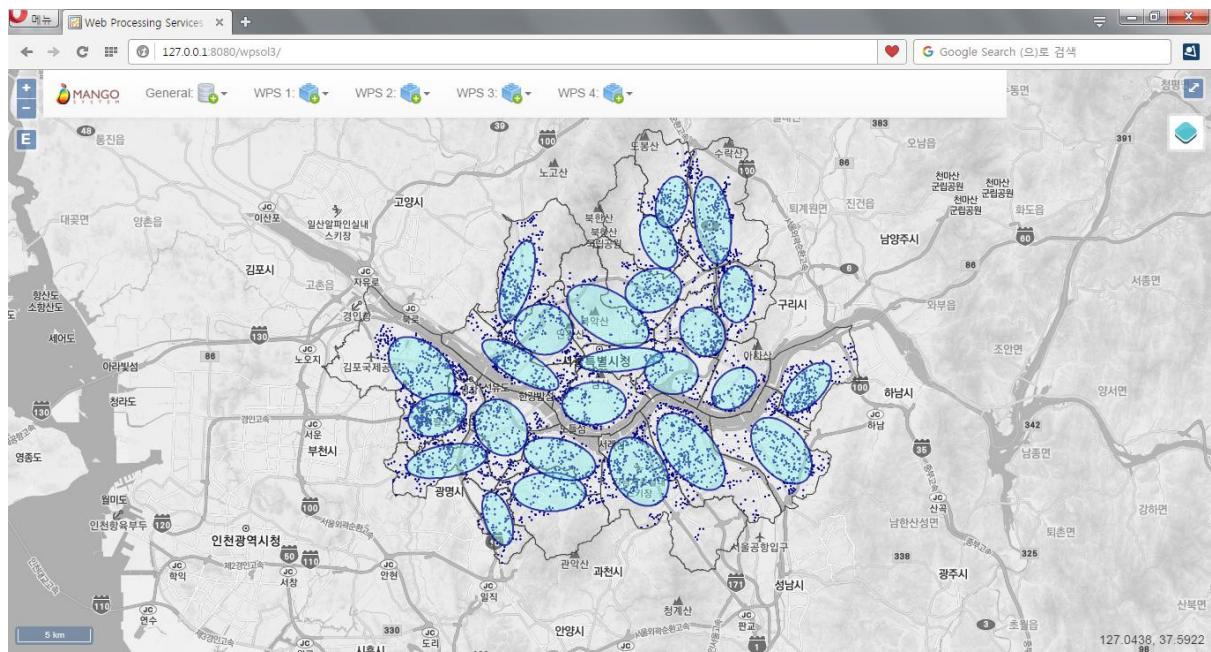
```

xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:StandardDeviationEllipse</ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3">
<xmlns:foss="http://www.opengeospatial.net/foss">
  <wfs:Query typeName="foss:apartment"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>ellipseSize</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>1_Standard_Deviation</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>caseField</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>sgg_nm</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of analyzing the Standard Deviational Ecclipe about the distribution of Seoul apartment by administrative boundary(Si-Gun-Gu)district.



4.4.2.6. Linear Directional Mean

Identify the geographic center, the average length, and the direction for all features of line feature layers.

■ Syntax

LinearDirectionalMean (SimpleFeatureCollection inputFeatures, Boolean orientationOnly, String caseField): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The line features containing vectors for which the mean direction will be calculated.	Complex	✓
orientationOnly	The From and To nodes are utilized in calculating the mean.	Literal	-
caseField	The field used to group features for separate directional mean calculations.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- The **inputFeatures** must be line features type.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
  xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
  xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
  xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
  xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
  http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:LinearDirectionalMean</ows:Identifier>
  <wps:DataInputs>
```

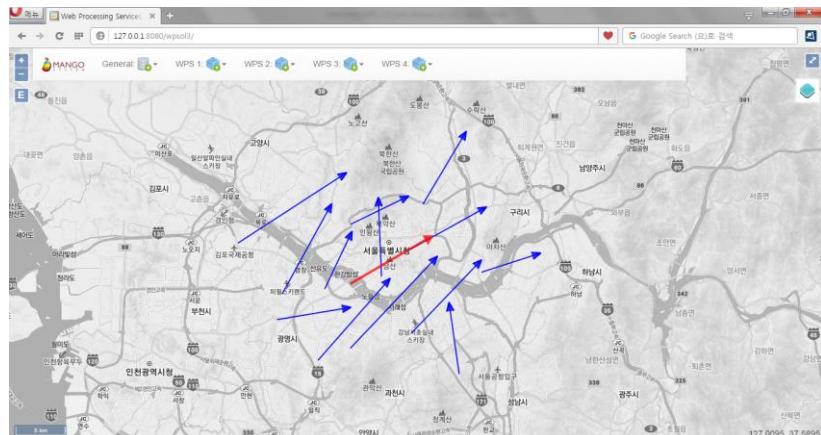
```

<wps:Input>
  <ows:Identifier>inputFeatures</ows:Identifier>
  <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
    <wps:Body>
      <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
      xmlns:foss="http://www.opengeospatial.net/foss">
        <wfs:Query typeName="foss:line"/>
      </wfs:GetFeature>
    </wps:Body>
  </wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>orientationOnly</ows:Identifier>
  <wps>Data>
    <wps:LiteralData>True</wps:LiteralData>
  </wps>Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of Linear Directional Mean analysis for line layers with length and directionality.



4.4.3. Point Pattern Analysis

Analyze the pattern of the point data.

4.4.3.1. Nearest Neighbor Statistic

Calculates the Nearest Neighbor Index based on the average distance from the feature closest to each feature in the feature layer.

■ Syntax

```
NearestNeighborIndex (SimpleFeatureCollection inputFeatures, DistanceMethod  
distanceMethod, Double area): NearestNeighborResult
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features.	Complex	✓
distanceMethod	Specifies how distances are calculated from each feature to neighboring features: Euclidean (default) or Manhattan.	Literal	-
area	A numeric value representing the study area.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Result Nearest Neighbor Index	Complex	✓

■ Constraints

- Calculates using Centroid of **inputFeatures**.
- Unless set the area parameter, use the Convex Hull Polygon area for Centroid in **inputFeatures**.
- The output is returned in XML format. If the returned Nearest Neighbor Ratio value is 1, Random, if it is larger than 1, it is distributed. If it is smaller than 1, it is concentrated.

■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:NearestNeighborIndex</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
            xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:apartment" />
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>distanceMethod</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>Euclidean</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

■ Response

This is the result of analyzing Average Nearest Neighbor for the distribution of apartments in Seoul and is returned in XML format. Since the Nearest Neighbor Ratio is less than 1, it can explain Cluster.

```
<?xml version="1.0" encoding="utf-8"?>
<NearestNeighborIndex>
    <TypeName>apartment</TypeName>
    <Observed_Point_Count>4052</Observed_Point_Count>
    <Study_Area>1.047557075141607E9</Study_Area>
    <Observed_Mean_Distance>200.00446</Observed_Mean_Distance>
    <Expected_Mean_Distance>254.22844</Expected_Mean_Distance>
    <Nearest_Neighbor_Ratio>0.786712</Nearest_Neighbor_Ratio>
    <Z_Score>-25.973484</Z_Score>
    <P_Value>0.0</P_Value>
    <Standard_Error>2.087667</Standard_Error>
</NearestNeighborIndex>
```

4.4.3.2. Quadrat Method

Analyze the point pattern using Quadrat analysis method.

■ Syntax

QuadratAnalysis (SimpleFeatureCollection inputFeatures, Double cellSize): QuadratResult

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The point features to be calculated.	Complex	✓
cellSize	The size of the grid cell.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	The Result of quadrat analysis	Complex	✓

■ Constraints

- Calculates using Centroid of **inputFeatures**.
- If you do not set the **cellSize** parameter, use the following formula to calculate the **cell size**. $\text{Math.sqrt} ((\text{BBOX area of inputFeatures} * 2) / \text{number of points})$.

■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:QuadratAnalysis</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
```

```
<wps:Body>
  <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
  xmlns:foss="http://www.opengeospatial.net/foss">
    <wfs:Query typeName="foss:gasstation" />
  </wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

This is the result of analyzing the Quadrate Method for the distribution of gas stations in Seoul and is returned in XML format.

```
<?xml version="1.0" encoding="utf-8"?>
<QuadratAnalysis>
  <TypeName>gasstation</TypeName>
  <FeatureCount>587</FeatureCount>
  <Area>1.4406602767217913E9</Area>
  <CellSize>2215.5254234488443</CellSize>
  <Columns>19</Columns>
  <Rows>16</Rows>
  <Number_of_Quadrats>304</Number_of_Quadrats>
  <Mean>1.930921052631579</Mean>
  <Variance>5.643254414819944</Variance>
  <Variance_Mean_Ratio>2.9225712812696134</Variance_Mean_Ratio>
  <Kolmogorov_Smirnov_Test>0.32209069225598863</Kolmogorov_Smirnov_Test>
  <Critical_Value_at_5percent>0.078001349515991</Critical_Value_at_5percent>
</QuadratAnalysis>
```

4.4.3.3. K-Nearest Neighbor Map

Creates line feature layers that connects the k-th nearest feature from all features in the feature layer.

■ Syntax

KNearestNeighborMap (SimpleFeatureCollection inputFeatures, Integer neighbor, Boolean convexHull): SimpleFeatureCollection

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features that can be point, line, polygon.	Complex	✓
neighbor	Number of Neighbors. Default is 1.	Literal	✓
convexHull	Add convex hull boundary to the output features. Default is True.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- The **Neighbor** parameter must be greater than or equal to 1 and defaults to 1.
- The Output layer is the line feature type.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
  xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
  xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
  xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
  xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
  http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:KNearestNeighborMap</ows:Identifier>
  <wps:DataInputs>
```

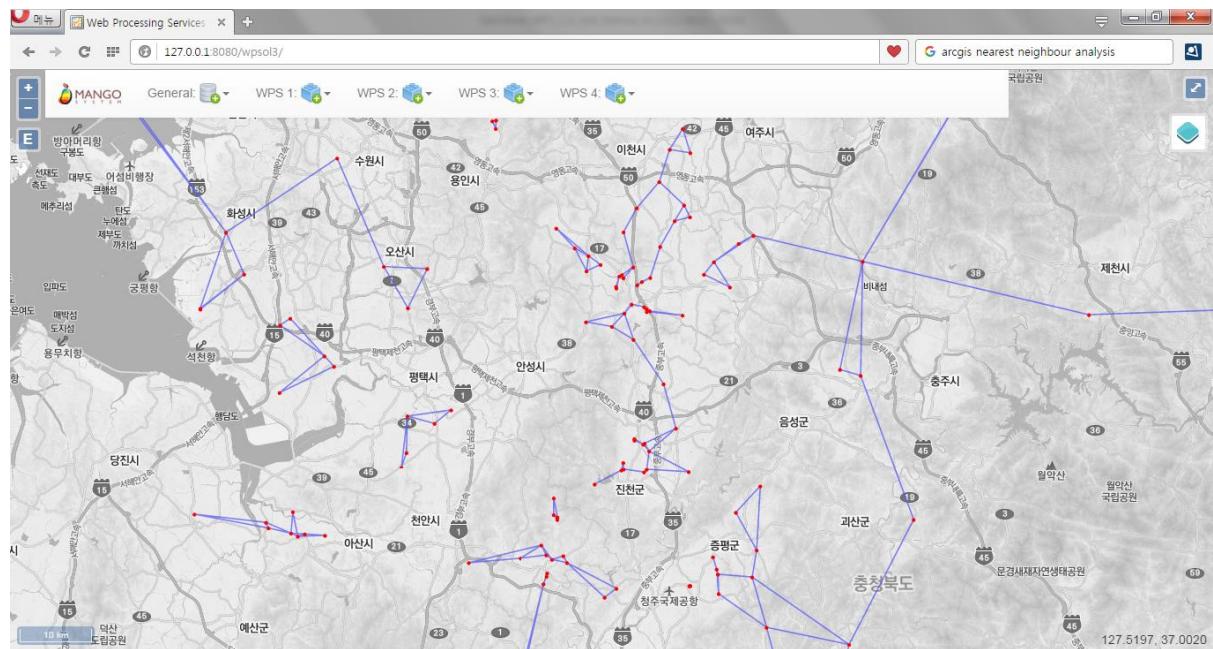
```

<wps:Input>
  <ows:Identifier>inputFeatures</ows:Identifier>
  <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
    <wps:Body>
      <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
      xmlns:foss="http://www.opengeospatial.net/foss">
        <wfs:Query typeName="foss:checkins"/>
      </wfs:GetFeature>
    </wps:Body>
  </wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>neighbor</ows:Identifier>
  <wps>Data>
    <wps:LiteralData>2</wps:LiteralData>
  </wps>Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>convexHull</ows:Identifier>
  <wps>Data>
    <wps:LiteralData>False</wps:LiteralData>
  </wps>Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of the K-Nearest Neighbor with the Neighbor parameter set to 2.



4.4.3.4. K-Means Clustering

All features in the feature layer are grouped into K clusters using the K-Means Clustering algorithm.

■ Syntax

```
KMeansClustering (SimpleFeatureCollection inputFeatures, String targetField, Integer  
numberOfClusters): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	Input features to be clustered.	Complex	✓
targetField	The numeric cluster id field to be calculated. Default is cluster	Literal	✓
numberOfClusters	The number of clusters to be grouped. Default is 5.	Literal	✓

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- Unless set the **targetField** parameter, use default cluster field,
- Unless set the **numberOfClusters** parameter, use default value of 5.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"  
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"  
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"  
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"  
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0  
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">  
    <ows:Identifier>statistics:KMeansClustering</ows:Identifier>  
    <wps:DataInputs>
```

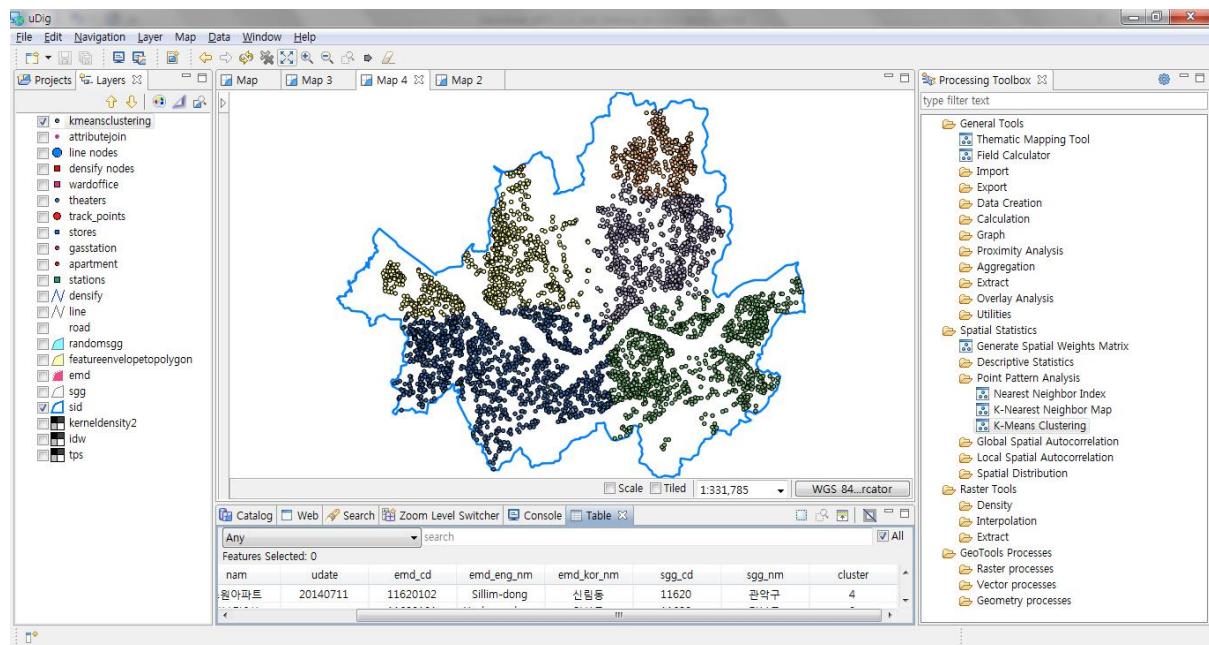
```

<wps:Input>
  <ows:Identifier>inputFeatures</ows:Identifier>
  <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
    <wps:Body>
      <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
      xmlns:foss="http://www.opengeospatial.net/foss">
        <wfs:Query typeName="foss:apartment"/>
      </wfs:GetFeature>
    </wps:Body>
  </wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>targetField</ows:Identifier>
  <wps>Data>
    <wps:LiteralData>cluster</wps:LiteralData>
  </wps>Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>numberOfClusters</ows:Identifier>
  <wps>Data>
    <wps:LiteralData>5</wps:LiteralData>
  </wps>Data>
</wps:Input>
</wps:Inputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response

The following figure shows the result of point layers Clusters with the set 5.



4.4.4. Global Spatial Auto-Correlation

It consists of processes for analyzing global spatial autocorrelation.

4.4.4.1. Join Count Statistic

Measure the global spatial autocorrelation of binary data (such as 1 or Black, 0 or White) based on the field values of the feature layer.

■ Syntax

```
JoinCount (SimpleFeatureCollection inputFeatures, Filter blackExpression, ContiguityType contiguityType): JoinCountProcessResult
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features for which join count statistics will be performed.	Complex	✓
blackExpression	Black Expression for 1 or True (for Black) value ex) [pop] > 1500.	Complex	✓
contiguityType	Contiguity Type(Queen, Rook, Bishops). Default is Queen.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Join Count Statistics.	Complex	✓

■ Constraints

- Both field and combination formula of fields can be the **blackExpression** parameter.
- Output is returned in XML format.

■ Request Examples

```
<?xml version="1.0" encoding="utf-8"?>
<wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
```

```

xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:JoinCount</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3">
            <wfs:Query typeName="foss:sgg" />
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>blackExpression</ows:Identifier>
      <wps:Data>
        <wps:ComplexData mimeType="text/plain; subtype=cql"><![CDATA[pop_den >
18890]]></wps:ComplexData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>contiguityType</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>Queen</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

■ Response

This is the result of analyzing the Join Count statistic by the method of the Queen, and it returns in the XML format.

```
<?xml version="1.0" encoding="utf-8"?>
<JoinCountStatistics>
    <TypeName>sgg</TypeName>
    <ContiguityType>Queen</ContiguityType>
    <FeatureCount>25</FeatureCount>
    <BlackCount>11</BlackCount>
    <WhiteCount>14</WhiteCount>
    <NumberOfJoins>56</NumberOfJoins>
    <ObservedBB>11</ObservedBB>
    <ObservedWW>18</ObservedWW>
    <ObservedBW>27</ObservedBW>
    <ExpectedBB>10.8416</ExpectedBB>
    <ExpectedWW>17.561600000000002</ExpectedWW>
    <ExpectedBW>27.5968</ExpectedBW>
    <StdDevBB>5.487588556005269</StdDevBB>
    <StdDevWW>6.831669500202715</StdDevWW>
    <StdDevBW>3.9479960283667954</StdDevBW>
    <ZScoreBB>0.028865137825731742</ZScoreBB>
    <ZScoreWW>0.06417172259094052</ZScoreWW>
    <ZScoreBW>-0.15116529898002093</ZScoreBW>
</JoinCountStatistics>
```

4.4.4.2. Moran's I

Measures the global Moran's / spatial autocorrelation based on the location and attribute values of the feature layer.

■ Syntax

```
GlobalMoransI (SimpleFeatureCollection inputFeatures, String inputField, SpatialConcept spatialConcept, DistanceMethod distanceMethod, StandardizationMethod standardization, Double searchDistance): MoransI
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features for which spatial autocorrelation will be calculated.	Complex	✓
inputField	The numeric field used in assessing spatial autocorrelation.	Literal	✓
spatialConcept	Specifies how spatial relationships among features are conceptualized. Default is InverseDistance	Literal	-
distanceMethod	Specifies how distances are calculated from each feature to neighboring features. Default is Euclidean	Literal	-
standardization	Row standardization. Default is None	Literal	-
searchDistance	Specifies a cutoff distance for Inverse Distance and Fixed Distance options.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output XML.	Complex	✓

■ Constraints

- Output is returned in XML format.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
```

```
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:GlobalMoransI</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:korea_sgg"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>inputField</ows:Identifier>
<wps:Data>
<wps:LiteralData>a3_2005</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>spatialConcept</ows:Identifier>
<wps:Data>
<wps:LiteralData>InverseDistance</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>distanceMethod</ows:Identifier>
<wps:Data>
<wps:LiteralData>Euclidean</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>standardization</ows:Identifier>
<wps:Data>
<wps:LiteralData>Row</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
```

```
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

```
<?xml version="1.0" encoding="utf-8"?>
<MoransI>
  <TypeName>korea_sgg</TypeName>
  <PropertyName>a3_2005</PropertyName>
  <Observed_Index>0.070175</Observed_Index>
  <Expected_Index>-0.004292</Expected_Index>
  <Variance>0.000203</Variance>
  <Z_Score>5.230945</Z_Score>
  <P_Value>0</P_Value>
  <Conceptualization>InverseDistance</Conceptualization>
  <DistanceMethod>Euclidean</DistanceMethod>
  <RowStandardization>Row</RowStandardization>
  <DistanceThreshold>191807.950591</DistanceThreshold>
</MoransI>
```

4.4.4.3. Geary's *c*

Measures the global Geary's *c* spatial autocorrelation based on the location and attribute values of the feature layer.

■ Syntax

```
GlobalGearysC (SimpleFeatureCollection inputFeatures, String inputField, SpatialConcept spatialConcept, DistanceMethod distanceMethod, StandardizationMethod standardization, Double searchDistance): GearysC
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features for which spatial autocorrelation will be calculated.	Complex	✓
inputField	The numeric field used in assessing spatial autocorrelation.	Literal	✓
spatialConcept	Specifies how spatial relationships among features are conceptualized. Default is InverseDistance	Literal	-
distanceMethod	Specifies how distances are calculated from each feature to neighboring features. Default is Euclidean	Literal	-
standardization	Row standardization. Default is None	Literal	-
searchDistance	Specifies a cutoff distance for Inverse Distance and Fixed Distance options.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output XML.	Complex	✓

■ Constraints

- Output is returned in XML format.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
```

```
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:GlobalGearysC</ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3">
<xmlns:foss="http://www.opengeospatial.net/foss">
  <wfs:Query typeName="foss:korea_sgg"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>inputField</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>a3_2005</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>spatialConcept</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>InverseDistance</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>distanceMethod</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>Euclidean</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>standardization</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>Row</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
```

```
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

```
<?xml version="1.0" encoding="utf-8"?>
<GlobalGearysC>
  <TypeName>korea_sgg</TypeName>
  <PropertyName>a3_2005</PropertyName>
  <Observed_Index>0.908981</Observed_Index>
  <Expected_Index>1</Expected_Index>
  <Variance>0.00029</Variance>
  <Z_Score>-5.341097</Z_Score>
  <P_Value>0</P_Value>
  <Conceptualization>InverseDistance</Conceptualization>
  <DistanceMethod>Euclidean</DistanceMethod>
  <RowStandardization>Row</RowStandardization>
  <DistanceThreshold>191807.950591</DistanceThreshold>
</GlobalGearysC>
```

4.4.4. Getis-Ord's General G

Measures the global Getis-Ord General *G* spatial autocorrelation based on the location and attribute values of the feature layer.

■ Syntax

```
GlobalGStatistics (SimpleFeatureCollection inputFeatures, String inputField,  
SpatialConcept spatialConcept, DistanceMethod distanceMethod, StandardizationMethod  
standardization, Double searchDistance): GeneralG
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features for which spatial autocorrelation will be calculated.	Complex	✓
inputField	The numeric field used in assessing spatial autocorrelation.	Literal	✓
spatialConcept	Specifies how spatial relationships among features are conceptualized. Default is InverseDistance	Literal	-
distanceMethod	Specifies how distances are calculated from each feature to neighboring features. Default is Euclidean	Literal	-
standardization	Row standardization. Default is None	Literal	-
searchDistance	Specifies a cutoff distance for Inverse Distance and Fixed Distance options.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output XML.	Complex	✓

■ Constraints

- Output is returned in XML format.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
```

```
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:GlobalGStatistics</ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3">
<xmlns:foss="http://www.opengeospatial.net/foss">
  <wfs:Query typeName="foss:korea_sgg"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>inputField</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>a3_2005</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>spatialConcept</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>InverseDistance</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>distanceMethod</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>Euclidean</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>standardization</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>Row</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps:DataInputs>
```

```
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

```
<?xml version="1.0" encoding="utf-8"?>
<GStatistics>
  <TypeName>korea_sgg</TypeName>
  <PropertyName>a3_2005</PropertyName>
  <Observed_Index>0.004492</Observed_Index>
  <Expected_Index>0.004292</Expected_Index>
  <Variance>0</Variance>
  <Z_Score>4.275913</Z_Score>
  <P_Value>0.000019</P_Value>
  <Conceptualization>InverseDistance</Conceptualization>
  <DistanceMethod>Euclidean</DistanceMethod>
  <RowStandardization>Row</RowStandardization>
  <DistanceThreshold>191807.950591</DistanceThreshold>
</GStatistics>
```

4.4.4.5. Lee's S

Measures the global Lee's *S* spatial autocorrelation based on the location and attribute values of the feature layer.

■ Syntax

```
GlobalLeesS (SimpleFeatureCollection inputFeatures, String inputField, SpatialConcept spatialConcept, DistanceMethod distanceMethod, StandardizationMethod standardization, Double searchDistance): LeesS
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features for which spatial autocorrelation will be calculated.	Complex	✓
inputField	The numeric field used in assessing spatial autocorrelation.	Literal	✓
spatialConcept	Specifies how spatial relationships among features are conceptualized. Default is InverseDistance	Literal	-
distanceMethod	Specifies how distances are calculated from each feature to neighboring features. Default is Euclidean	Literal	-
standardization	Row standardization. Default is None	Literal	-
searchDistance	Specifies a cutoff distance for Inverse Distance and Fixed Distance options.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output XML.	Complex	✓

■ Constraints

- Output is returned in XML format.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
```

```
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:GlobalLeesS</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3">
<wfs:Query typeName="foss:korea_sgg"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>inputField</ows:Identifier>
<wps:Data>
<wps:LiteralData>a3_2005</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>spatialConcept</ows:Identifier>
<wps:Data>
<wps:LiteralData>InverseDistance</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>distanceMethod</ows:Identifier>
<wps:Data>
<wps:LiteralData>Euclidean</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>standardization</ows:Identifier>
<wps:Data>
<wps:LiteralData>Row</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
```

```
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

```
<?xml version="1.0" encoding="utf-8"?>
<GlobalLeesS>
  <TypeName>korea_sgg</TypeName>
  <PropertyName>a3_2005</PropertyName>
  <Observed_Index>0.065413</Observed_Index>
  <Expected_Index>0.090566</Expected_Index>
  <Variance>0</Variance>
  <Z_Score>0</Z_Score>
  <P_Value>1</P_Value>
  <Conceptualization>InverseDistance</Conceptualization>
  <DistanceMethod>Euclidean</DistanceMethod>
  <RowStandardization>Row</RowStandardization>
  <DistanceThreshold>191807.950591</DistanceThreshold>
</GlobalLeesS>
```

4.4.4.6. Lee's L

Measures the global Lee's L spatial autocorrelation based on the location and two attribute values of the feature layer.

■ Syntax

```
GlobalLeesL (SimpleFeatureCollection inputFeatures, String xField, String yField,  
SpatialConcept spatialConcept, DistanceMethod distanceMethod, StandardizationMethod  
standardization, Double searchDistance): LeesL
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features for which spatial autocorrelation will be calculated.	Complex	✓
xField	The numeric x field used in assessing spatial autocorrelation.	Literal	✓
yField	The numeric y field used in assessing spatial autocorrelation.	Literal	✓
spatialConcept	Specifies how spatial relationships among features are conceptualized. Default is InverseDistance	Literal	-
distanceMethod	Specifies how distances are calculated from each feature to neighboring features. Default is Euclidean	Literal	-
standardization	Row standardization. Default is None	Literal	-
searchDistance	Specifies a cutoff distance for Inverse Distance and Fixed Distance options.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output XML.	Complex	✓

■ Constraints

- Output is returned in XML format.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:GlobalLeesL</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
        xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:korea_sgg"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>inputField</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>a3_2005</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>spatialConcept</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>InverseDistance</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>distanceMethod</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>Euclidean</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>standardization</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>Row</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>

```

```
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response

```
<?xml version="1.0" encoding="utf-8"?>
<GlobalLeesL>
  <TypeName>korea_sgg</TypeName>
  <PropertyName>a + b</PropertyName>
  <Observed_Index>0.42206509</Observed_Index>
  <Expected_Index>0.090566</Expected_Index>
  <Variance>0</Variance>
  <Z_Score>0</Z_Score>
  <P_Value>1</P_Value>
  <Conceptualization>0.42206509</Conceptualization>
  <DistanceMethod>Euclidean</DistanceMethod>
  <RowStandardization>Row</RowStandardization>
  <DistanceThreshold>0</DistanceThreshold>
</GlobalLeesL>
```

4.4.5. Local Spatial Auto-Correlation

It consists of processes for analyzing local spatial autocorrelation.

4.4.5.1. Local Moran's I

Using the Anselin's local Moran's / statistics from the feature layers, identify statistically significant hot spot, cold spot and spatial outliers.

■ Syntax

```
LocalMoransI (SimpleFeatureCollection inputFeatures, String inputField, SpatialConcept spatialConcept, DistanceMethod distanceMethod, StandardizationMethod standardization, Double searchDistance): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features for which spatial autocorrelation will be calculated.	Complex	✓
inputField	The numeric field used in assessing spatial autocorrelation.	Literal	✓
spatialConcept	Specifies how spatial relationships among features are conceptualized. Default is InverseDistance	Literal	-
distanceMethod	Specifies how distances are calculated from each feature to neighboring features. Default is Euclidean	Literal	-
standardization	Row standardization. Default is None	Literal	-
searchDistance	Specifies a cutoff distance for Inverse Distance and Fixed Distance options.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

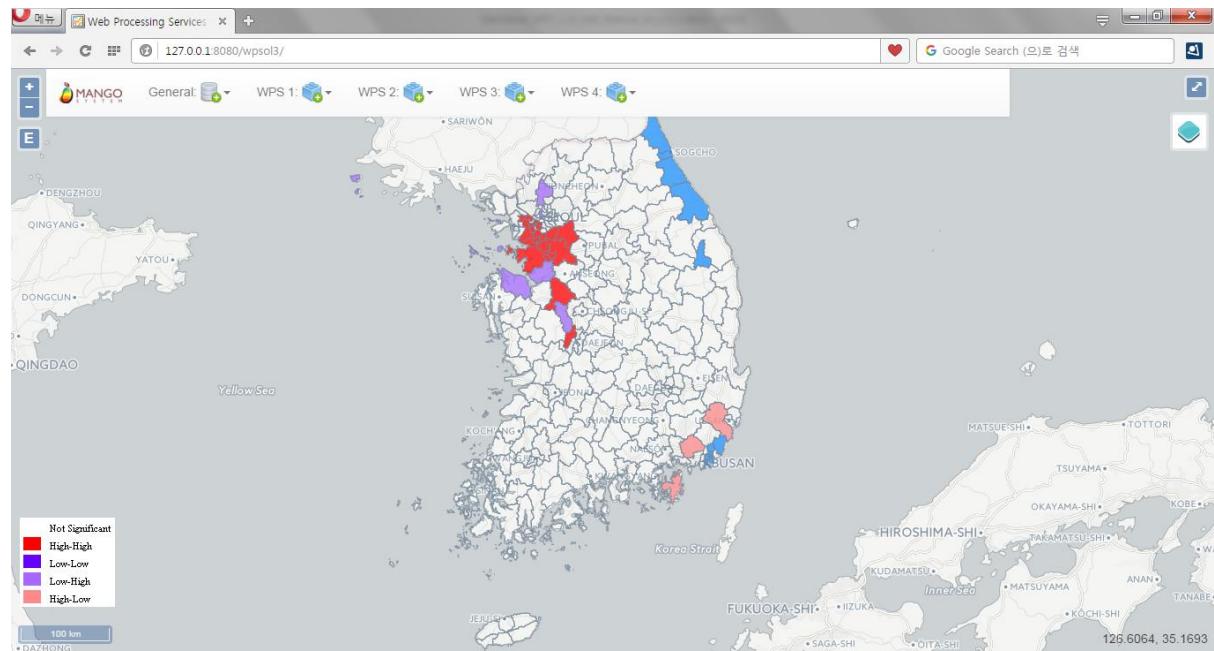
- The Output layer contains all the fields of **inputFeatures**, with the field of LMIndex, LMZScore, LMIPValue, LMIZValue, LMiwzValue, and COType added.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:LocalMoransI</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
        xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:korea_sgg"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>inputField</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>a3_2005</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>spatialConcept</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>InverseDistance</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>distanceMethod</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>Euclidean</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>standardization</ows:Identifier>
      <wps:Data>
```

```
<wps:LiteralData>Row</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
<ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response



4.4.5.2. Local G (G_i^*)

Identify statistically significant hot spots and cold spots using the local Getis-Ord G_i^* statistic from the feature layer.

■ Syntax

```
LocalGStatistics (SimpleFeatureCollection inputFeatures, String inputField, SpatialConcept spatialConcept, DistanceMethod distanceMethod, StandardizationMethod standardization, Double searchDistance): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features for which spatial autocorrelation will be calculated.	Complex	✓
inputField	The numeric field used in assessing spatial autocorrelation.	Literal	✓
spatialConcept	Specifies how spatial relationships among features are conceptualized. Default is InverseDistance	Literal	-
distanceMethod	Specifies how distances are calculated from each feature to neighboring features. Default is Euclidean	Literal	-
standardization	Row standardization. Default is None	Literal	-
searchDistance	Specifies a cutoff distance for Inverse Distance and Fixed Distance options.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- The Output layer contains all the fields of **inputFeatures**, with the fields of GiZScore, GiMean, GiVar, and GiPValue added.

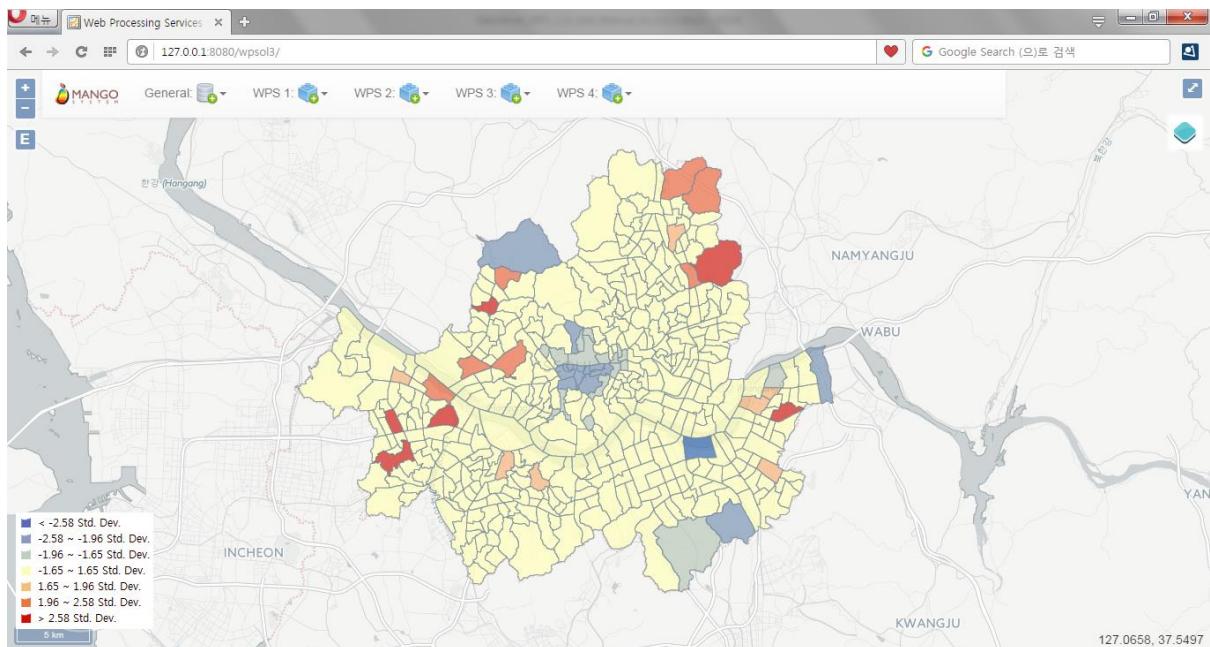
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:LocalGStatistics</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:emd"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>inputField</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>pts</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>spatialConcept</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>InverseDistance</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>distanceMethod</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>Euclidean</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>standardization</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>Row</wps:LiteralData>
      </wps:Data>
    </wps:Input>
```

```
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response



4.4.5.3. Local Geary's *c*

Calculates the local Geary's *c* statistic from the feature layer

■ Syntax

```
LocalGearysC (SimpleFeatureCollection inputFeatures, String inputField, SpatialConcept spatialConcept, DistanceMethod distanceMethod, StandardizationMethod standardization, Double searchDistance): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features for which spatial autocorrelation will be calculated.	Complex	✓
inputField	The numeric field used in assessing spatial autocorrelation.	Literal	✓
spatialConcept	Specifies how spatial relationships among features are conceptualized. Default is InverseDistance	Literal	-
distanceMethod	Specifies how distances are calculated from each feature to neighboring features. Default is Euclidean	Literal	-
standardization	Row standardization. Default is None	Literal	-
searchDistance	Specifies a cutoff distance for Inverse Distance and Fixed Distance options.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- The Output layer contains all the fields of **inputFeatures**, with the fields of LGcIndex, LGcZScore, and LGcPValue added.

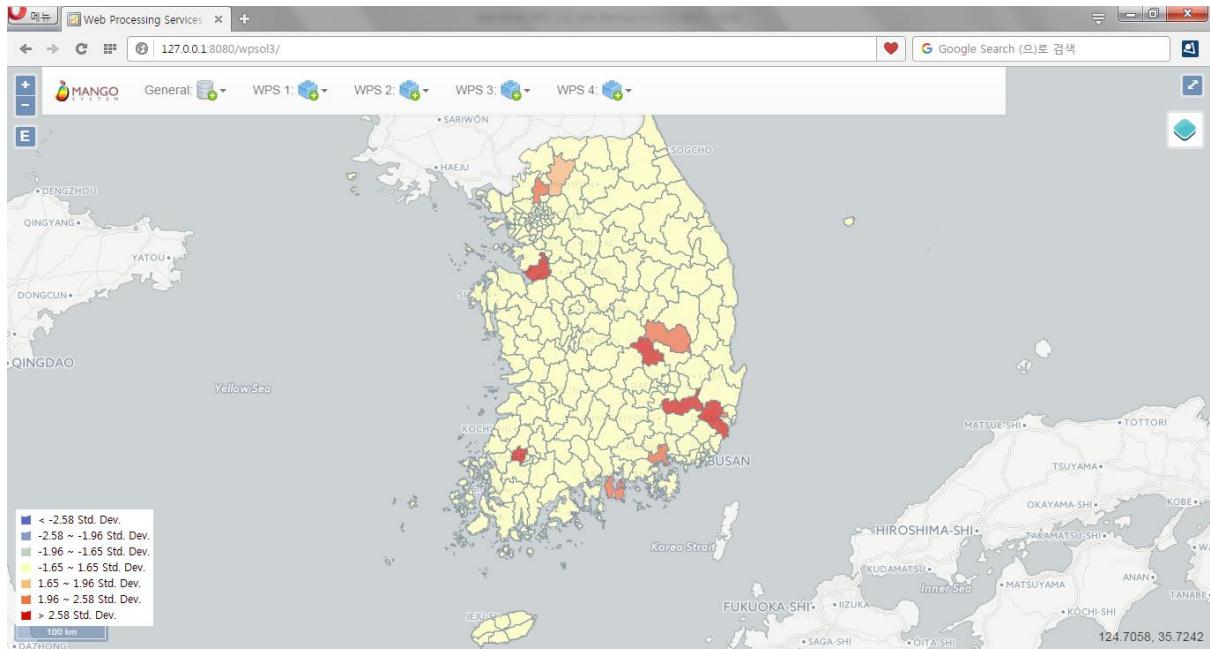
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
```

```
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:LocalGearysC</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>inputFeatures</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
<wps:Body>
<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
<wfs:Query typeName="foss:korea_sgg"/>
</wfs:GetFeature>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
<ows:Identifier>inputField</ows:Identifier>
<wps:Data>
<wps:LiteralData>a3_2005</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>spatialConcept</ows:Identifier>
<wps:Data>
<wps:LiteralData>InverseDistance</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>distanceMethod</ows:Identifier>
<wps:Data>
<wps:LiteralData>Euclidean</wps:LiteralData>
</wps:Data>
</wps:Input>
<wps:Input>
<ows:Identifier>standardization</ows:Identifier>
<wps:Data>
<wps:LiteralData>Row</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
```

```
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response



4.4.5.4. Lee's Si

Calculates the local Lee's *S*, statistics from the feature layer.

■ Syntax

```
LocalLeesS (SimpleFeatureCollection inputFeatures, String inputField, SpatialConcept spatialConcept, DistanceMethod distanceMethod, StandardizationMethod standardization, Double searchDistance): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features for which spatial autocorrelation will be calculated.	Complex	✓
inputField	The numeric field used in assessing spatial autocorrelation.	Literal	✓
spatialConcept	Specifies how spatial relationships among features are conceptualized. Default is InverseDistance	Literal	-
distanceMethod	Specifies how distances are calculated from each feature to neighboring features. Default is Euclidean	Literal	-
standardization	Row standardization. Default is None	Literal	-
searchDistance	Specifies a cutoff distance for Inverse Distance and Fixed Distance options.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- The Output layer contains all the fields of **inputFeatures**, with the fields of LLsIndex, LLsZScore and LLsPValue added.

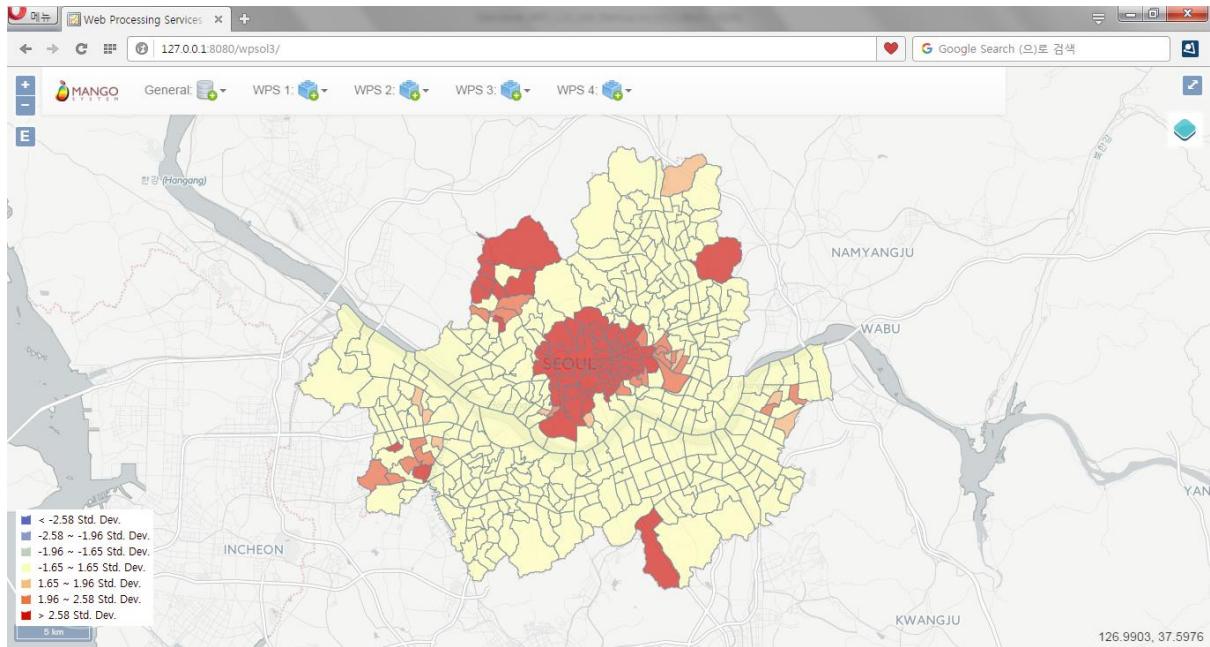
■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
```

```
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifier>statistics:LocalLeesS</ows:Identifier>
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>inputFeatures</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3">
<xmlns:foss="http://www.opengeospatial.net/foss">
  <wfs:Query typeName="foss:emd"/>
</wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>inputField</ows:Identifier>
    <wps:Data>
      <wps:LiteralData>pts</wps:LiteralData>
    </wps:Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>spatialConcept</ows:Identifier>
    <wps:Data>
      <wps:LiteralData>InverseDistance</wps:LiteralData>
    </wps:Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>distanceMethod</ows:Identifier>
    <wps:Data>
      <wps:LiteralData>Euclidean</wps:LiteralData>
    </wps:Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>standardization</ows:Identifier>
    <wps:Data>
      <wps:LiteralData>Row</wps:LiteralData>
    </wps:Data>
  </wps:Input>
</wps:DataInputs>
```

```
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

■ Response



4.4.5.5. Lee's Li

Calculates the local Lee's L_i statistic from the feature layer and two attribute values.

■ Syntax

```
LocalLeesL (SimpleFeatureCollection inputFeatures, String xField, String yField,  
SpatialConcept spatialConcept, DistanceMethod distanceMethod, StandardizationMethod  
standardization, Double searchDistance): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features for which spatial autocorrelation will be calculated.	Complex	✓
xField	The numeric x field used in assessing spatial autocorrelation.	Literal	✓
yField	The numeric y field used in assessing spatial autocorrelation.	Literal	✓
spatialConcept	Specifies how spatial relationships among features are conceptualized. Default is InverseDistance	Literal	-
distanceMethod	Specifies how distances are calculated from each feature to neighboring features. Default is Euclidean	Literal	-
standardization	Row standardization. Default is None	Literal	-
searchDistance	Specifies a cutoff distance for Inverse Distance and Fixed Distance options.	Literal	-

■ Process Outputs

Identifier	Description	Type	Required
result	Output features.	Complex	✓

■ Constraints

- The Output layer contains all the fields of **inputFeatures**, with the fields of LLIndex, LLIScore, and LLIPValue fields added.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>statistics:LocalLeesL</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>inputFeatures</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.1.0" outputFormat="GML3"
xmlns:foss="http://www.opengeospatial.net/foss">
            <wfs:Query typeName="foss:hexa2009"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>xField</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>a2009</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>yField</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>b2009</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>spatialConcept</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>ContiguityEdgesNodes</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>distanceMethod</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>Euclidean</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps:DataInputs>

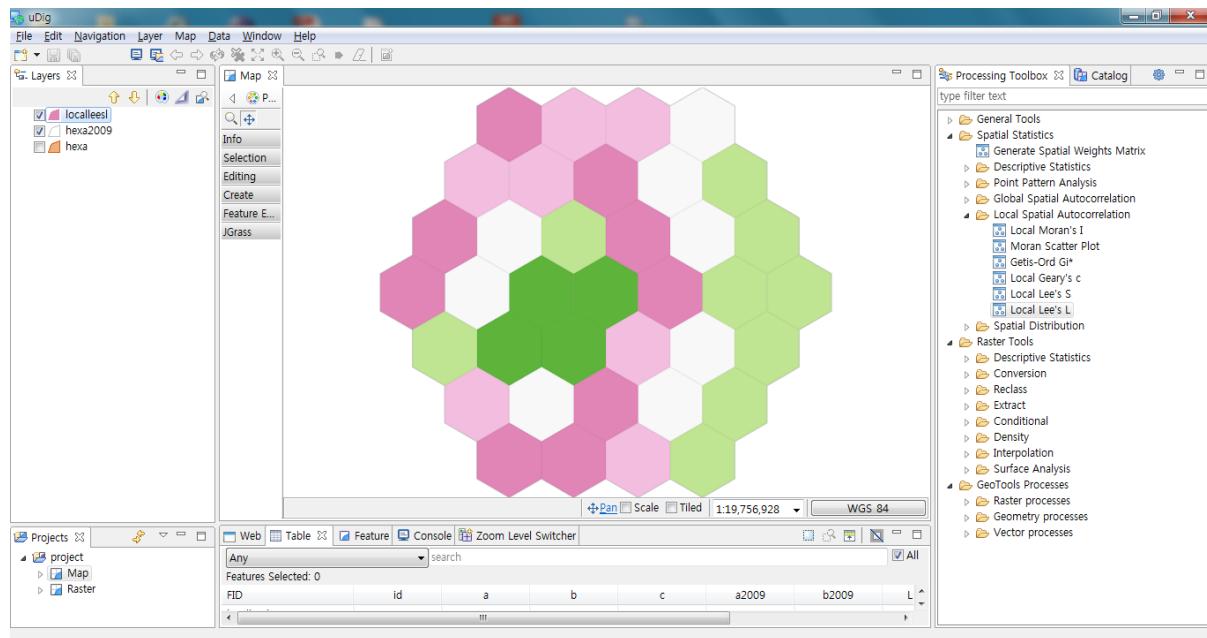
```

```

</wps:Input>
<wps:Input>
  <ows:Identifier>standardization</ows:Identifier>
<wps:Data>
  <wps:LiteralData>Row</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.1">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

■ Response



4.4.6. Global Spatial Modeling

It consists of processes that analyze global spatial modeling and spatial relationships.

4.4.6.1. Ordinary Least Squares (OLS)

Performs global Ordinary Least Squares(OLS) linear regression.

■ Syntax

```
OrdinaryLeastSquares (SimpleFeatureCollection inputFeatures, String dependentVariable,  
String explanatoryVariables): SimpleFeatureCollection
```

■ Parameters

■ Data Inputs

Identifier	Description	Type	Required
inputFeatures	The features containing the dependent and independent variables for analysis.	Complex	✓
dependentVariable	The numeric field containing values for what you are trying to model.	Literal	✓
explanatoryVariables	The comma separated fields representing explanatory variables in your regression model.	Literal	✓

■ Process Outputs

Identifier	Description	Type	Required
olsFeatures	The output features to receive dependent variable estimates and residuals.	Complex	
report	Output OLS results.	Complex	✓

■ Constraints

- The olsFeatures layer contains all the fields of **inputFeatures**, with the fields of Estimated, Residual, StdResid, and StdResid2 added.
- Output is returned in XML format.

■ Request Examples

```
<?xml version="1.0" encoding="UTF-8"?> <wps:Execute version="1.0.0" service="WPS"
```

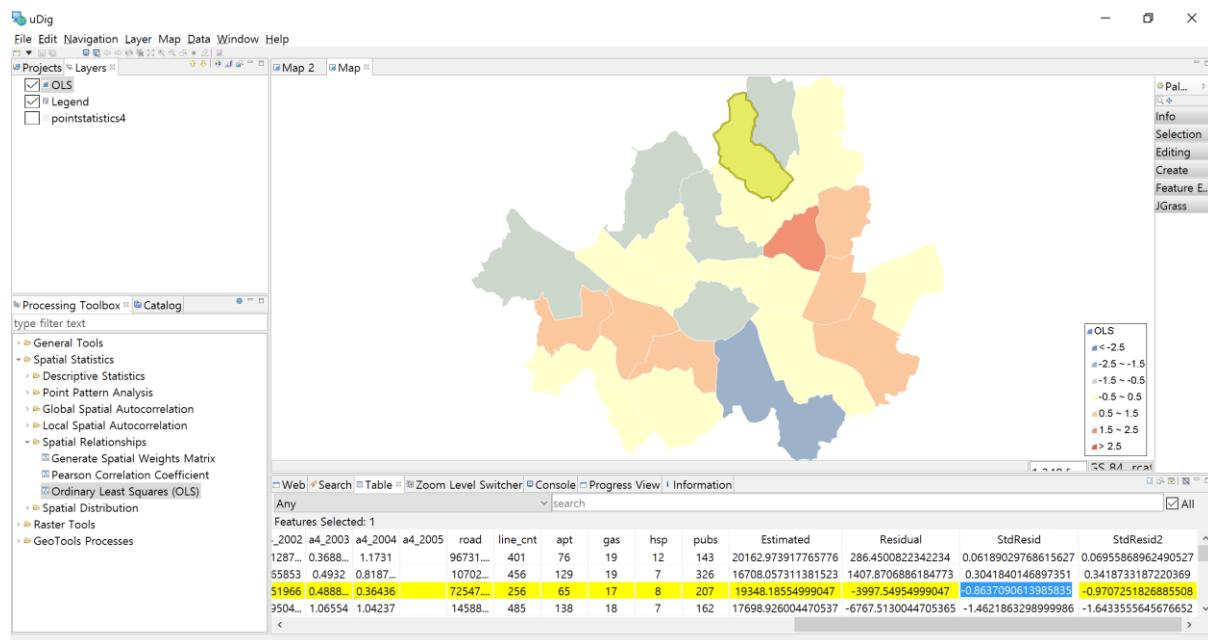
```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:
>
<wps:Data> </wps:Execute>

```

■ Response

● Map



● XML

```

<?xml version="1.0" encoding="utf-8"?>
<OrdinaryLeastSquares>
  <Diagnostics>
    <R>0.6101560399292024</R>
    <RSquared>0.3722903930620865</RSquared>
    <AdjustedRSquared>0.3290000753422304</AdjustedRSquared>
    <StandardError>12.933848555707566</StandardError>
    <NumberOfObservations>63</NumberOfObservations>
    <AIC>508.1174991231418</AIC>
    <AICc>509.6174991231418</AICc>
  </Diagnostics>
  <Variance>
    <Regression>
      <DegreesOfFreedom>4</DegreesOfFreedom>
      <SumOfSquare>5754.486696189348</SumOfSquare>
      <SquareMean>1438.621674047337</SquareMean>
    </Regression>
  </Variance>
</OrdinaryLeastSquares>

```

```
<FStatistic>8.599853562435902</FStatistic>
<FProbability>1.609736408525464E-5</FProbability>
</Regression>
<Residual>
    <DegreesOfFreedom>58</DegreesOfFreedom>
    <SumOfSquare>9702.497430794765</SumOfSquare>
    <SquareMean>167.2844384619787</SquareMean>
</Residual>
<Sum>
    <DegreesOfFreedom>62</DegreesOfFreedom>
    <SumOfSquare>15456.984126984113</SumOfSquare>
</Sum>
</Variance>
<Variables>
    <Item>
        <Variable>Intercept</Variable>
        <Coefficient>1.8149624761790997</Coefficient>
        <StdError>2.256350184412793</StdError>
        <TStatistic>0.8043797849806887</TStatistic>
        <Probability>0.42445922884684884</Probability>
    </Item>
    <Item>
        <Variable>etc</Variable>
        <Coefficient>6.922393697035605</Coefficient>
        <StdError>1.217664912349127</StdError>
        <TStatistic>5.684974270697246</TStatistic>
        <Probability>5.13680658710916E-7</Probability>
    </Item>
    <Item>
        <Variable>ccc</Variable>
        <Coefficient>-0.2585330201573057</Coefficient>
        <StdError>0.4672538188264545</StdError>
        <TStatistic>-0.5533031721530541</TStatistic>
        <Probability>0.5821837017619861</Probability>
    </Item>
    <Item>
        <Variable>mlbflb</Variable>
        <Coefficient>-0.04321817419824479</Coefficient>
        <StdError>2.571603704239341</StdError>
        <TStatistic>-0.01680592313932304</TStatistic>
        <Probability>0.9866481737384326</Probability>
    </Item>
    <Item>
```

```
<Variable>airport</Variable>
<Coefficient>-2.3644517041718496</Coefficient>
<StdError>2.64338529816144</StdError>
<TStatistic>-0.8944786466870351</TStatistic>
<Probability>0.3747591210649666</Probability>
</Item>
</Variables>
</OrdinaryLeastSquares>
```