

Intro

과학기술 분야에서 흔히 접하는 데이터 형태로서 지점별로 데이터가 존재하는 경우가 꽤 많습니다. 예를 들어 기상관측 데이터를 얻는데 있어서 측정장비가 지점별로 산재해있는 경우입니다. 만약 N개의 지점들이 존재한다고 하면 각 지점마다 그 위치는 제각각일 것입니다. 그리고 이러한 위치정보는 경위도 좌표로 존재할 것입니다. 따라서 이러한 데이터는 N개의 측정값들, N개의 경도좌표값들 및 N개의 위도좌표값들로 존재하는 경우가 일반적이라고 볼 수 있습니다. 그런데 이렇게 불규칙하게 산재해있는 지점별 데이터들을 내삽하여 규칙 격자화된 데이터로 환산해야 하는 경우가 종종 있습니다. 물론 이러한 계산에 있어서, 원래 데이터값이 존재하지 않는 격자점들에 대해서도 값을 계산해야 하기 때문에 다양한 내삽 기법들이 존재합니다. IDL에서도 이러한 2차원 격자화 내삽 연산을 지원하는 내장함수들이 여러가지 있습니다. 그래서 몇몇 함수들을 이용하여 격자화를 수행하는 방법 및 그 결과를 살펴봅시다.

지점별 데이터 표출

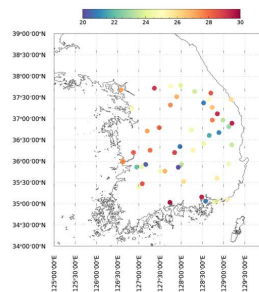
먼저 가상의 데이터를 다음과 같이 생성하고 지점별 표출부터 해봅시다. 여기서는 난수생성 기능을 사용하여 50개의 가상의 지점들에 대하여 관측값, 경도좌표, 위도좌표 값들에 해당되는 data, lons, lats라는 1차원 배열들을 생성하였습니다. 지점들은 주로 한반도 내에 분포하도록 하였고, 데이터 값의 범위는 20~30 정도가 되도록 하였습니다.

```
n = 50
seed = -3
lons = 126.5+RANDOMU(seed, n)*2.7
lats = 35+RANDOMU(seed, n)*2.8
data = 20+RANDOMU(seed, n)*10
HELP, lons, lats, data
PRINT, MIN(data), MAX(data)
```

그리고 MAP 함수 및 SCATTERPLOT 함수를 이용하여 다음과 같이 바탕지도를 그리고 지점별 값들을 특정 컬러테이블 기반으로 표출할 수 있습니다. 표출된 그림은 아래와 같습니다.

```
win = WINDOW(DIMENSIONS=[1200, 600], $
/NO_TOOLBAR)
m1 = MAP('geographic', LIMIT=[34,125,39,130], $
MARGIN=[0.12, 0.2, 0.05, 0.1], /CURRENT, $
```

```
LAYOUT=[2, 1, 1])
mc1 = MAPCONTINENTS(/HIRES)
ct = COLORTABLE(74, /REVERSE)
sp = SCATTERPLOT(lons, lats, $
MAGNITUDE=BYTSCAL(data, MIN=20, MAX=30), $
RGB_TABLE=ct, SYMBOL='circle', $
/SYM_FILLED, /OVERPLOT)
m1.MapGrid.LABEL_POSITION = 0
m1.MapGrid.LINESTYLE = 1
m1.MapGrid.HORIZON_LINESTYLE = 0
cb1 = COLORBAR(TARGET=sp, RANGE=[20, 30], $
/BORDER, POSITION=[0.14, 0.95, 0.4, 0.98])
```



격자화 결과 산출 (IDW 기법)

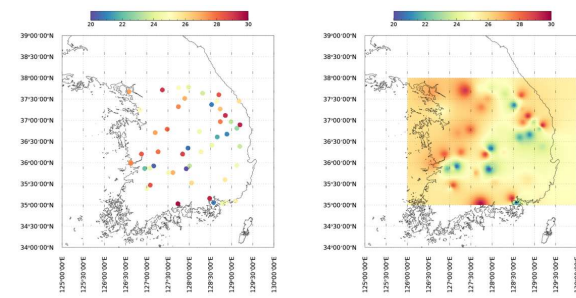
먼저 GRIDDATA 함수를 이용하여 IDW 기법(Inverse Distance Weighting)을 적용하여 결과를 얻어봅시다. 이 기법은 각 격자점의 값을 계산하는데 있어서 주변 지점들의 값들을 내삽에 반영할 때 거리가 가까울수록 가중치를 높게 그리고 멀수록 가중치를 낮게 주는 방식입니다. 다음과 같은 방식으로 사용합니다.

```
data_gridded = GRIDDATA(lons, lats, data, $
DELTA=[0.01, 0.01], DIMENSION=[401, 301], $
START=[126, 35], /DEGREES)
HELP, data_gridded
PRINT, MIN(data_gridded), MAX(data_gridded)
```

여기서는 결과로 얻게 될 격자 데이터가 경도 방향으로 동경 126~130도 위도 방향으로 북위 35~38도의 범위 내에 존재하고 격자 해상도는 0.01도가 되도록 설정하였습니다. 따라서 그 배열구조는 401x301의 형태가 됩니다. 이렇게 얻어진 격자 데이터를 그래픽창의 우측에서 맵과 이미지가 중첩된 형태로 다음과 같이 표출해봅시다. 그러면 아래 그림과 같이 결과가 표출됩니다.

```
m2 = MAP('geographic', $
LIMIT=[34, 125, 39, 130], $
```

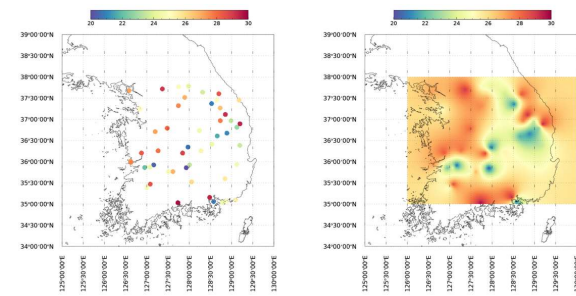
```
MARGIN=[0.12, 0.2, 0.05, 0.1], $
/CURRENT, LAYOUT=[2, 1, 2])
ct = COLORTABLE(74, /REVERSE)
im = IMAGE(data_gridded, RGB_TABLE=ct, $
MIN_VALUE=20, MAX_VALUE=30, $
IMAGE_LOCATION=[126, 35], $
IMAGE_DIMENSION=[4, 3], $
GRID_UNITS=2, /OVERPLOT)
mc2 = MAPCONTINENTS(/HIRES)
m2.MapGrid.LABEL_POSITION = 0
m2.MapGrid.LINESTYLE = 1
m2.MapGrid.HORIZON_LINESTYLE = 0
cb2 = COLORBAR(TARGET=im, /BORDER, $
POSITION=[0.64, 0.95, 0.9, 0.98])
```



격자화 결과 산출 (Kriging 기법)

이번에는 Kriging 기법을 사용하여 격자 데이터를 만들어 보겠습니다. 이 기법은 영향반경을 정의하여 격자점으로부터 그 거리 내에 있는 지점 데이터들만을 반영하는 방식이며, 영향반경 내 지점들의 거리에 따른 가중치의 변화 패턴을 정의하는 몇가지 함수들이 존재합니다. IDL에서는 GRIDDATA 함수 내에서도 이 기법을 지원하지만, 그 보다는 전용 함수인 KRIG2D 함수를 사용하는 것을 권장합니다. 격자화 데이터인 data_gridded를 산출하는 부분만 다음과 같은 내용으로 대체하여 다시 실행하면 됩니다. 그러면 결과는 아래 그림과 같습니다.

```
data_gridded = KRIG2D(data, lons, lats, $
EXPONENTIAL=[1.0, 0], GS=[0.01, 0.01], $
BOUND=[126, 35, 130, 38])
```



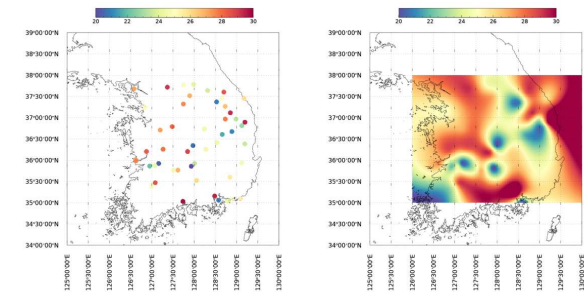
여기서는 거리에 따른 가중치의 변화 패턴에 있어서

Exponential 함수인 경우를 가정하였습니다. KRIG2D 함수에서는 이를 비롯하여 총 4종의 함수 패턴들이 지원되므로 도움말의 내용을 참조하여 선택하여 사용하시면 됩니다.

격자화 결과 산출 (TPS 기법)

이번에는 GRID_TPS 함수를 이용하여 TPS(Thin Plate Spline)라는 기법이 적용된 격자화 데이터를 산출해 보겠습니다. 해당 내용은 다음과 같습니다. 그러면 아래 그림과 같은 결과를 얻게 됩니다.

```
data_gridded = GRID_TPS(lons, lats, data, $
NGRID=[401, 301], START=[126, 35], $
DELTA=[0.01, 0.01])
```



기타의 방법들

앞서 소개한 3개의 경우들 외에도 IDL에서는 다양한 기법들을 적용하여 격자화 결과를 산출할 수 있습니다. GRIDDATA 함수 내에서도 총 10종의 기법들이 지원되며, 그 외에도 TRI_SURF, SPH_SCAT 등과 같은 함수들도 있습니다. 지면 관계상 여기서 다 소개할 수는 없지만 참조하시기 바랍니다. 다음은 위의 예제에서 TRI_SURF 및 SPH_SCAT 함수를 적용할 경우의 예문입니다.

```
data_gridded = TRI_SURF(data, lons, lats, $
BOUNDS=[126, 35, 130, 38], $
GS=[0.01, 0.01], /EXTRAPOLATE)
```

```
data_gridded = SPH_SCAT(lons, lats, data, $
BOUNDS=[126, 35, 130, 38], GS=[0.01, 0.01])
```

이와 같이 불규칙 분포 데이터를 규칙 격자화하는 작업에 있어서 어떤 기법을 적용할 것이며, 격자 해상도 및 범위를 어떻게 해야 할 것인지 등은 전적으로 연구자의 판단의 몫입니다. 그리고 격자화라는 작업 자체가 원래 데이터가 없는 부분들에 대하여 값을 인위적으로 계산하는 것이기 때문에, 그러한 값들을 어느 정도까지 신뢰할 수 있느냐에 대해서도 신중한 접근이 필요합니다.