

Intro

특정 지역에 대한 2차원적인 데이터가 있을 때, 먼저 그 지역의 지도를 표시하고 그 위에 2차원 데이터를 중첩하는 작업을 해야 하는 경우가 많습니다. 2차원 데이터는 주로 이미지(Image) 또는 등위선(Contour)의 형태로 표시할 수 있습니다. 따라서 IDL에서 **지도/이미지** 또는 **지도/등위선**의 조합으로 **중첩 표시**하는 방법에 관하여 관련 예제들과 함께 각각 소개해보고자 합니다.

예제 데이터 준비

예제로 사용할 2차원 데이터는 다음과 같이 HANNING 함수를 이용하여 생성합니다. 이 데이터는 400x400의 구조를 갖는 실수형 배열이며, 경도상으로 동경 120~140도 그리고 위도상으로 북위 30~50도의 범위를 커버한다고 가정하였습니다. 이러한 **2차원 데이터 배열**을 다음과 같이 data라는 이름으로 생성합니다.

```
data = HANNING(400, 400)*100
add = FLTARR(400, 400)
add[200, 200] = HANNING(200, 200)*50
data = data + add
```

그리고 이러한 작업에서는 2차원 데이터의 격자들이 위치하는 경도값들 및 위도값들로 구성된 배열을 각각 만들어서 사용하는 것이 좋습니다. 따라서 경도 방향으로 동경 120도부터 139.95도까지 0.05의 간격으로 총 400개의 **경도값들로 구성된 배열** lons, 그리고 위도 방향으로 북위 30도부터 49.95도까지 0.05의 간격으로 총 400개의 **위도값들로 구성된 배열** lats를 다음과 같이 정의합니다.

```
lons = 120 + FINDGEN(400)*0.05
lats = 30 + FINDGEN(400)*0.05
```

이와 같이 lons, lats와 같은 배열들까지도 추가적으로 사용하는 것은 지도/이미지 중첩 표시에서는 필수적인 것은 아닙니다. 하지만 **지도/등위선 중첩 표시에서는 필수**입니다.

지도/이미지 중첩 표시

그러면 지도/이미지 중첩 표시부터 해봅시다. 먼저 2차원 데이터의 분포 영역에 맞는 경도 및 위도 범위의 바탕 지도를 표시하고, 그 위에 이미지를 중첩하는 방식으로 구현하면 됩니다. 일단 바탕 지도를 표시하는 과정은 다음과 같습니다. 여기서 지도의 투영법은 LCC(Lambert Conformal Conic) 투영법을 적용합니다.

```
win = WINDOW(DIMENSIONS=[600, 600], $
/NO_TOOLBAR)
limit = [30, 120, 50, 140]
m = MAP('Lambert Conformal Conic', $
LIMIT=limit, STANDARD_PAR1=30, $
STANDARD_PAR2=60, ASPECT_RATIO=0, $
POSITION=[0.08, 0.2, 0.94, 0.96], $
CLIP=0, /CURRENT)
```

여기서는 WINDOW 및 MAP 함수를 사용하여 지도의 기본 바탕만 먼저 구현하였습니다. 원래는 MAPCONTINENTS 함수도 바로 이어서 사용하여 대륙경계선까지 표시해야 하겠지만, 이 과정은 이미지의 중첩 이후에 진행해야 합니다. 그렇게 해야 **대륙경계선이 이미지에 덮어씌워지는 현상**을 방지할 수 있습니다. 이제 이미지를 중첩해야 하는데 두가지 방법이 있습니다. 먼저 **이미지 중첩의 첫번째 방법**은 다음과 같습니다.

```
im = IMAGE(data, RGB_TABLE=74, $
IMAGE_DIMENSIONS=[20, 20], $
IMAGE_LOCATION=[120, 30], $
GRID_UNITS=2, ASPECT_RATIO=0, $
/OVERPLOT)
```

첫번째 방법에서는 위와 같이 **IMAGE_LOCATION** 및 **IMAGE_DIMENSION 속성들을** 사용하여 이미지의 좌측 하단 꼭지점의 좌표 및 가로/세로 방향 크기를 경도/위도 단위로 명시하면 됩니다. 그리고 바로 이어서 대륙경계선의 표시 및 기타 자잘한 손질들 그리고 컬러바의 표시까지 다음과 같이 처리합니다.

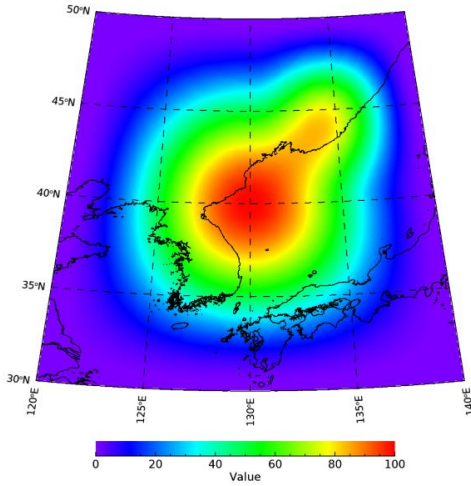
```
mc = MAPCONTINENTS(/HIRES)
m.MapGrid.LABEL_POSITION = 0
m.MapGrid.LINESTYLE = 2
m.MapGrid.HORIZON_LINESTYLE = 0
cb = COLORBAR(TARGET=im, $
TITLE='Value', FONT_SIZE=10, $
POSITION=[0.2, 0.07, 0.8, 0.10])
```

지면 관계상 여기서 **이미지 중첩의 두번째 방법**도 바로 소개하겠습니다. 앞서 IMAGE 함수가 사용되었던 내용만 다음과 같이 바꾸면 됩니다.

```
im = IMAGE(data, lons, lats, $
RGB_TABLE=34, GRID_UNITS=2, $
ASPECT_RATIO=0, /OVERPLOT)
```

이 방법에서는 위와 같이 IMAGE 함수에서 **2차원 데이터 배열인 data** 뿐 아니라 **lons, lats 배열들까지 두번째 및 세번째 인수로** 사용합니다. 그러면 2차원 배열의 격자별 경위도 위치를 lons, lats 배열들로부터 인지하게 됩니다. 앞서 첫번째 방법에서는 data 배열만 사용하고 lons, lats 배열들은 사용되지 않았음에 유의합니다.

그리고 첫번째 방법을 사용하면 두번째 방법을 사용하는 표출 결과는 동일합니다. 그 모습은 다음 그림과 같습니다.

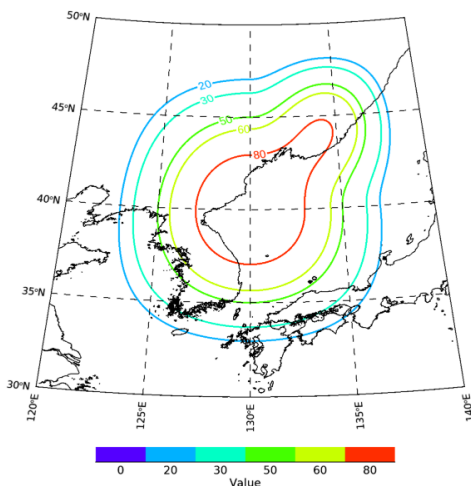


지도/등위선 중첩 표출

이번에는 지도/등위선 중첩 표출을 해봅시다. 이를 위하여 CONTOUR 함수를 사용해야 합니다. 앞서 제시된 지도 표출과 관련된 내용은 그대로 사용하면 됩니다. 다만 IMAGE 함수가 사용되었던 부분만 다음과 같은 내용으로 대체하면 됩니다.

```
conv = [0, 20, 30, 50, 60, 80]
cn = CONTOUR(data, lons, lats, $
    RGB_TABLE=34, C_VALUE=conv, $
    C_LABEL_SHOW=1, C_THICK=2, $
    GRID_UNITS=2, /OVERPLOT)
```

이렇게 했을 때 표출된 결과는 다음 그림과 같습니다.



여기서 CONTOUR 함수 내에 투입된 인수들을 보면, 앞서 이미지 중첩의 두번째 방법에서와 같이 data, lons, lats 배열들이 모두 사용되었음에 주목해야 합니다.

지도와 등위선을 중첩하는 작업에서 CONTOUR 함수를 사용하는 방법은 오직 이 방법 뿐입니다. 즉 앞서 이미지 중첩의 첫번째 방법에서 IMAGE 함수 내에 오직 data 배열만 투입했던 것과 같은 방법은 지도/등위선 중첩에서는 사용할 수 없다는 뜻입니다. 따라서 지도/등위선 중첩 작업을 위해서는 데이터 배열 뿐 아니라 경도 및 위도 격자값들로 구성된 배열들까지도 만들어서 사용하는 것이 필수임을 유의해야 합니다.

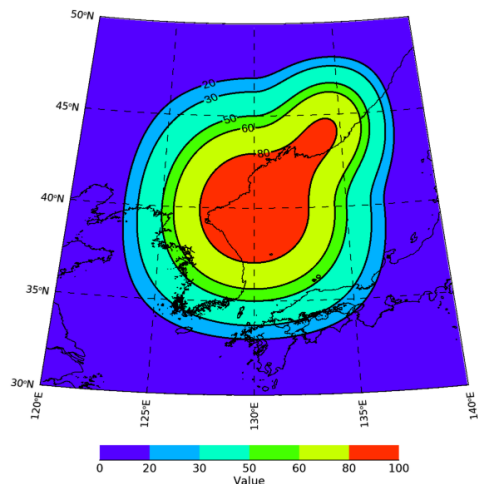
그리고 앞선 그림은 등위선을 선의 형태로만 표시하면서 컬러테이블을 적용한 경우입니다. 만약 컬러테이블의 색상으로 선과 선 사이가 채워지는 형태로 표출하고자 한다면, 다음과 같이 CONTOUR 함수를 두 번 사용하는 것이 좋습니다. 첫번째 CONTOUR 함수는 색상이 채워진 등위선(Filled Contour)을 표출하고, 두번째 CONTOUR 함수는 그 위에 선 및 라벨 문자들만 표시하는 역할을 합니다.

```
conv = [0, 20, 30, 50, 60, 80]
cn = CONTOUR(data, lons, lats,
    RGB_TABLE=34, C_VALUE=conv, $
    /FILL, GRID_UNITS=2, /OVERPLOT)
cno = CONTOUR(data, lons, lats, $
    COLOR='black', C_VALUE=conv, $
    C_LABEL_SHOW=1, C_THICK=2, $
    GRID_UNITS=2, /OVERPLOT)
```

그리고 등위선에 대한 컬러바의 표출을 위하여 COLORBAR 함수를 사용할 때 다음과 같이 TAPER 속성을 0으로 설정해주는 것이 좋습니다. 그렇지 않으면 컬러바의 양 옆이 뾰족한 형태가 되어버리기 때문입니다.

```
cb = COLORBAR(TARGET=cn, $
    TITLE='Value', FONT_SIZE=10, $
    POSITION=[0.2, 0.07, 0.8, 0.1], $
    TAPER=0)
```

표출된 모습은 다음과 같습니다.



오늘 소개된 내용에 관한 좀 더 자세한 내용은 아래 링크에서도 보실 수 있습니다.

<https://blog.daum.net/swrush/582>