

Tools for data analyses in Cosmology

- Aula 6 -

Camila Novaes

Observatório Nacional

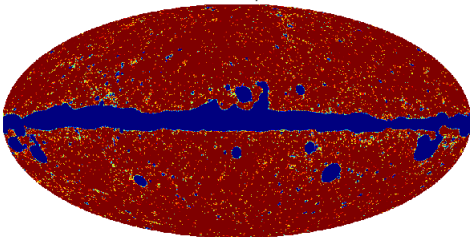
May 25, 2017

Healpy

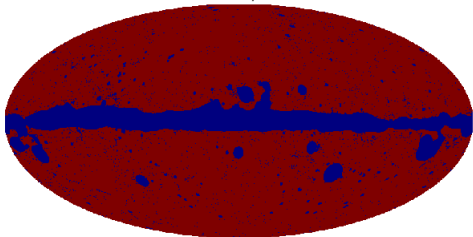
Previously on **healpy class** ...

- upgrading and degrading a mask -

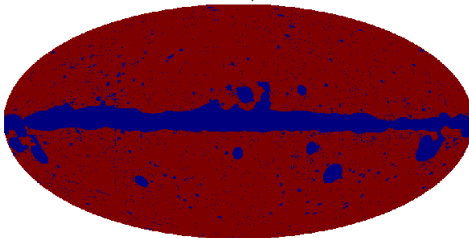
Confidence Mask / Nside = 128



Confidence Mask / Nside = 2048



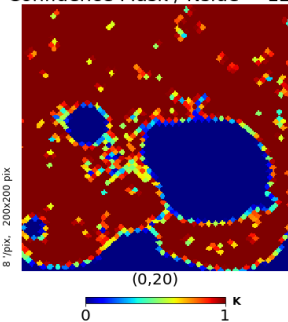
Confidence Mask / Nside = 1024



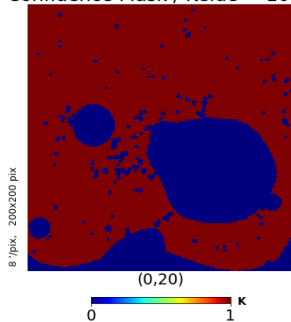
Previously on `healpy` class ...

- upgrading and degrading a mask -

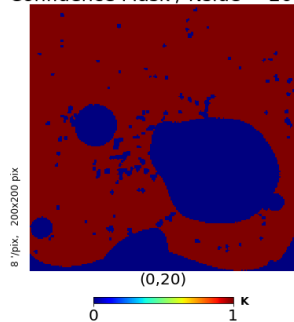
Confidence Mask / Nside = 128



Confidence Mask / Nside = 1024



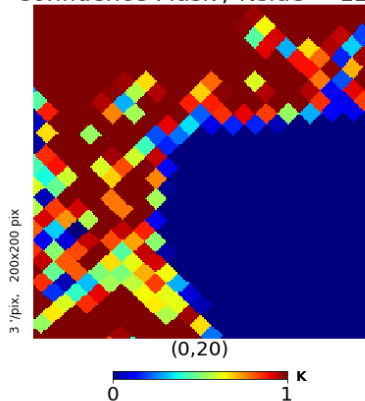
Confidence Mask / Nside = 2048



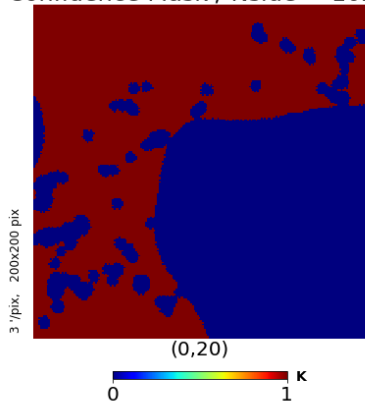
Previously on `healpy` class ...

- upgrading and degrading a mask -

Confidence Mask / Nside = 128



Confidence Mask / Nside = 1024



Masking pixels ...

Treating the missing value

healpy.pixelfunc.UNSEEN

`healpy.pixelfunc.UNSEEN = -1.6375e+30`

Special value used for masked pixels

How to use:

```
In [118]: miss = hp.UNSEEN
```

```
In [119]: miss
```

```
Out[119]: -1.6375e+30
```

Treating the missing value

healpy.pixelfunc.mask_bad

```
healpy.pixelfunc.mask_bad(m, badval=-1.6375e+30, rtol=1e-05, atol=1e-08)
```

Returns a bool array with `True` where *m* is close to *badval*.

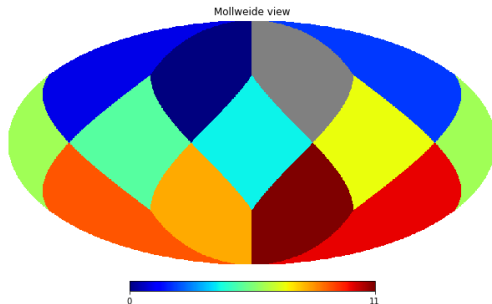
healpy.pixelfunc.mask_good

```
healpy.pixelfunc.mask_good(m, badval=-1.6375e+30, rtol=1e-05, atol=1e-08)
```


Treating the missing value

How to use:

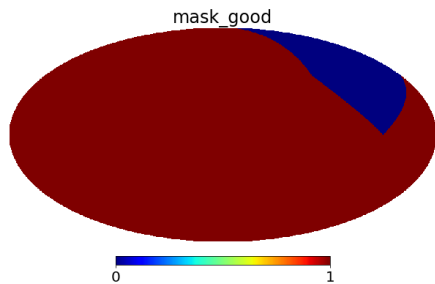
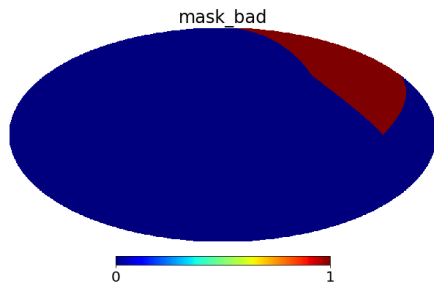
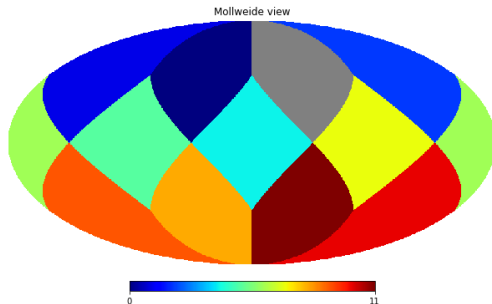
```
In [8]: import healpy as hp
...: import numpy as np
...: m = np.arange(12.)
...: m[2] = hp.UNSEEN
...:
...: m_bad = hp.mask_bad(m)
...:
...: m_good = hp.mask_good(m)
...:
...: print('m_bad ->', m_bad)
...: print('m_good ->', m_good)
...:
...: hp.mollview(m_bad, title='mask_bad')
...: hp.mollview(m_good, title='mask_good')
...:
```



Treating the missing value

How to use:

```
In [8]: import healpy as hp
...: import numpy as np
...: m = np.arange(12.)
...: m[2] = hp.UNSEEN
...:
...: m_bad = hp.mask_bad(m)
...:
...: m_good = hp.mask_good(m)
...:
...: print('m_bad ->', m_bad)
...: print('m_good ->', m_good)
...:
...: hp.mollview(m_bad, title='mask_bad')
...: hp.mollview(m_good, title='mask_good')
```

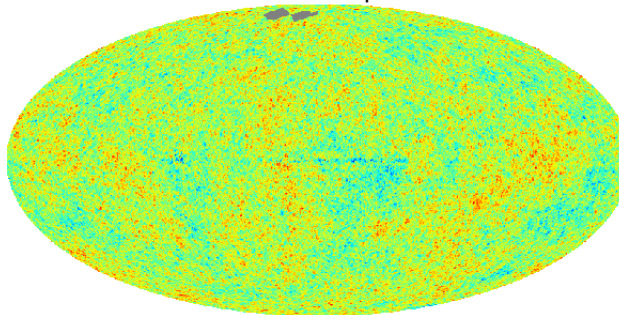


Treating the missing value

- what is it for? -

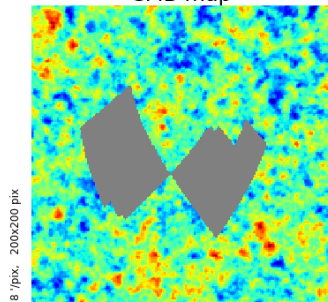
```
In [120]: mapa = hp.read_map('CMB_map_Ns1024.fits', field={0,1}, h=True)  
NSIDE = 1024  
ORDERING = RING in fits file  
INDXSCHM = IMPLICIT
```

CMB map



-0.000605052 K 0.000480358

CMB map



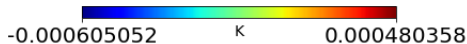
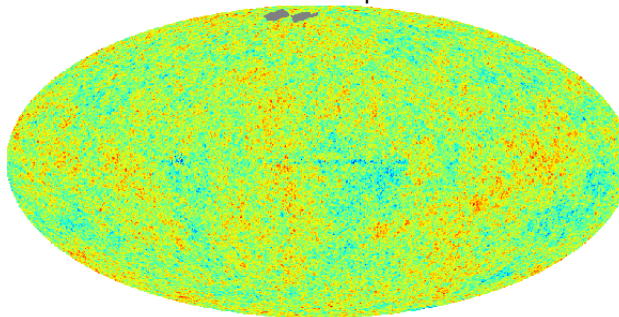
-0.000327 K 0.000439

Treating the missing value

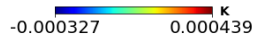
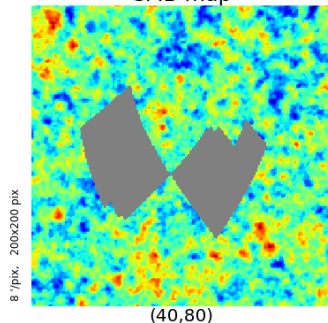
Exercise:

- Read the CMB map and the corresponding confidence mask.
- Remove from this mask (**pixel values = 0**) the missing pixels in the CMB map.
- Visualize the result.

CMB map

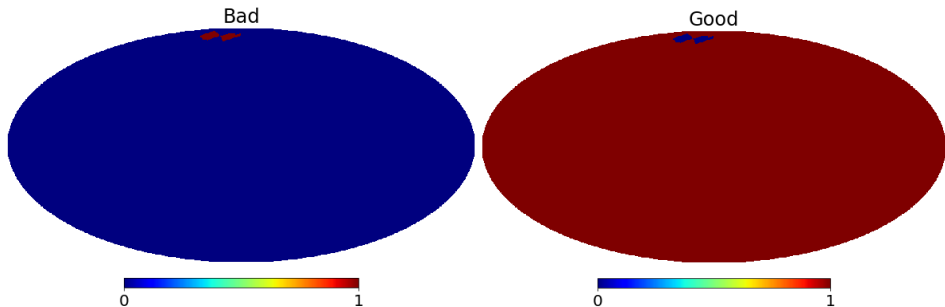


CMB map



Treating the missing value

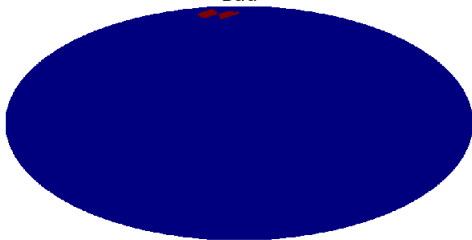
Results:



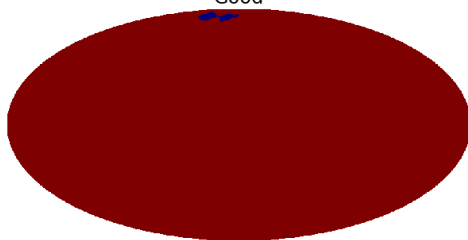
Treating the missing value

Results:

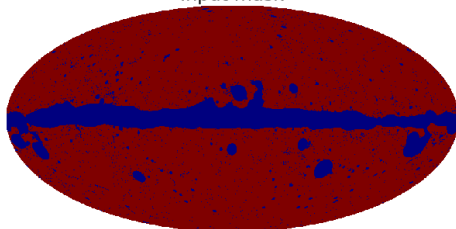
Bad



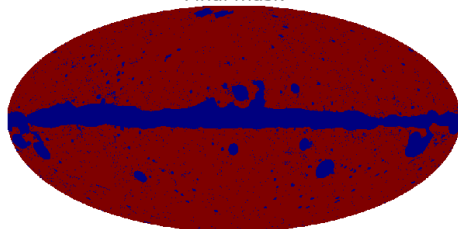
Good



Input mask



Final mask



Treating the missing value

There is an alternative way of dealing with UNSEEN pixel based on the numpy MaskedArray class, `ma()` loads a map as a masked array:

healpy.pixelfunc.ma

```
healpy.pixelfunc.ma(m, badval=-1.6375e+30, rtol=1e-05, atol=1e-08, copy=True)
```

Treating the missing value

How to use:

```
In [29]: cmb_masked = hp.ma(cmb)
```

```
In [30]: cmb
```

```
Out[30]:  
array([ -1.26071362e-04,  -8.35976316e-05,  -6.17667538e-05, ...,  
        8.17280015e-05,   8.86737107e-05,   7.60578187e-05])
```

```
In [31]: cmb_masked
```

```
Out[31]:  
masked_array(data = [-0.00012607136159203947 -8.359763160115108e-05  
-6.176675378810614e-05 ...,  
 8.17280015326105e-05 8.867371070664376e-05 7.60578186600469e-05],  
             mask = [False False False ..., False False False],  
             fill_value = -1.6375e+30)
```

Visualize it!

Treating the missing value

How to use:

```
In [32]: cmb_masked.mask = np.logical_not(mask)
```

```
In [33]: cmb_masked
```

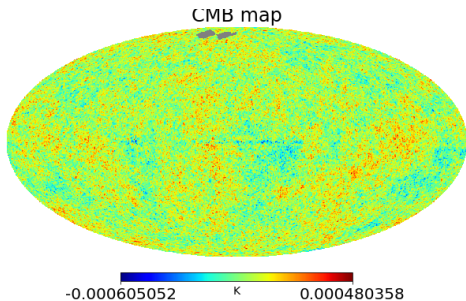
```
Out[33]:
```

```
masked_array(data = [-0.00012607136159203947 -8.359763160115108e-05  
-6.176675378810614e-05 ...,  
 8.17280015326105e-05 8.867371070664376e-05 7.60578186600469e-05],  
             mask = [False False False ..., False False False],  
             fill_value = -1.6375e+30)
```

Visualize it!

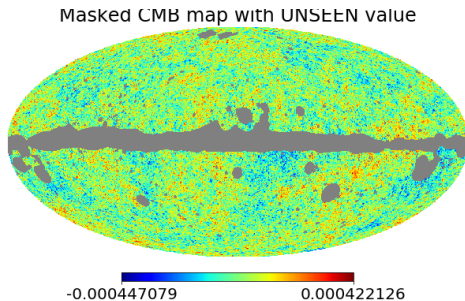
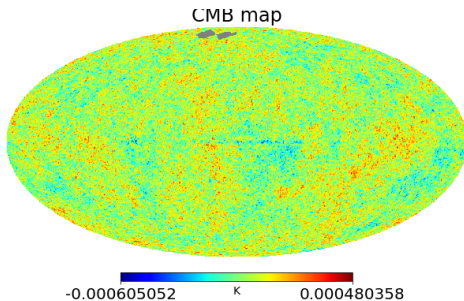
Treating the missing value

Result:



Treating the missing value

Result:



Treating the missing value

How to use:

```
In [34]: cmb_masked0 = cmb_masked.filled()
```

```
In [35]: cmb_masked0
```

```
Out[35]:  
array([ -1.26071362e-04,  -8.35976316e-05,  -6.17667538e-05,  ...,  
        8.17280015e-05,   8.86737107e-05,   7.60578187e-05])
```

Filling a masked array fills in the UNSEEN value and return a standard array!

Map data manipulation

healpy.pixelfunc.remove_monopole

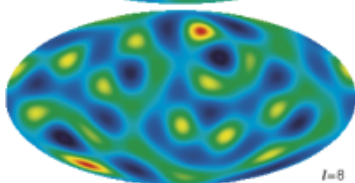
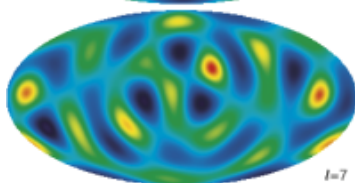
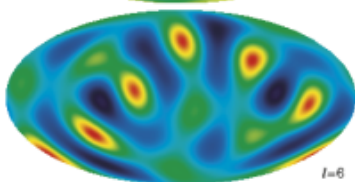
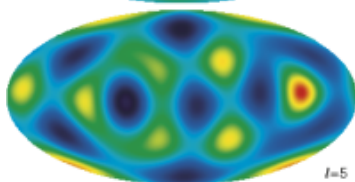
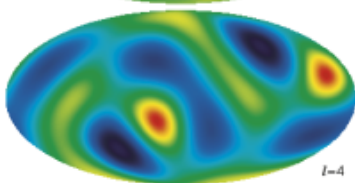
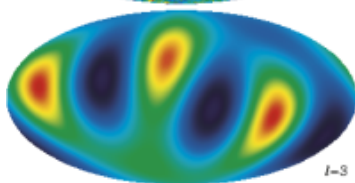
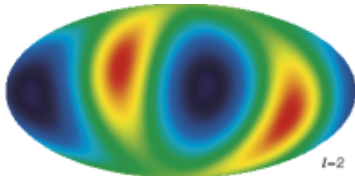
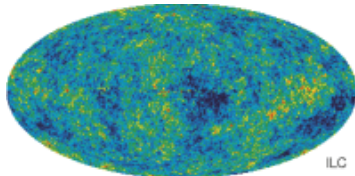
```
healpy.pixelfunc.remove_monopole(m, nest=False, bad=-1.6375e+30, gal_cut=0, fitval=False, copy=True, verbose=True)
```

Fit and subtract the monopole from the given map *m*.

healpy.pixelfunc.remove_dipole

```
healpy.pixelfunc.remove_dipole(m, nest=False, bad=-1.6375e+30, gal_cut=0, fitval=False, copy=True, verbose=True)
```

Fit and subtract the dipole and the monopole from the given map *m*.



Map data manipulation

How to use:

```
In [37]: mapa = hp.read_map('LFI_CompMap_Foregrounds-smica_1024_R2.00.fits')
NSIDE = 1024
ORDERING = NESTED in fits file
INDXSCHM = IMPLICIT
/home/camila/anaconda3_4p3p1/lib/python3.6/site-packages/healpy/fitsfunc.py:339:
UserWarning: No INDXSCHM keyword in header file : assume IMPLICIT
    "assume {}".format(schm))
Ordering converted to RING

In [38]: print('<mapa> =', np.mean(mapa))
<mapa> = 5.0560580581e-06

In [39]: mapa2 = hp.remove_monopole(mapa)
monopole: 5.05606e-06

In [40]: print('<mapa2> =', np.mean(mapa2))
<mapa2> = -1.22113916562e-20
```

Visualize it!

Map data manipulation

How to use:

```
In [37]: mapa = hp.read_map('LFI_CompMap_Foregrounds-smica_1024_R2.00.fits')
NSIDE = 1024
ORDERING = NESTED in fits file
INDXSCHM = IMPLICIT
/home/camila/anaconda3_4p3p1/lib/python3.6/site-packages/healpy/fitsfunc.py:339:
UserWarning: No INDXSCHM keyword in header file : assume IMPLICIT
    "assume {}".format(schm))
Ordering converted to RING

In [38]: print('<mapa> =', np.mean(mapa))
<mapa> = 5.0560580581e-06

In [39]: mapa2 = hp.remove_monopole(mapa)
monopole: 5.05606e-06

In [40]: print('<mapa2> =', np.mean(mapa2))
<mapa2> = -1.22113916562e-20
```

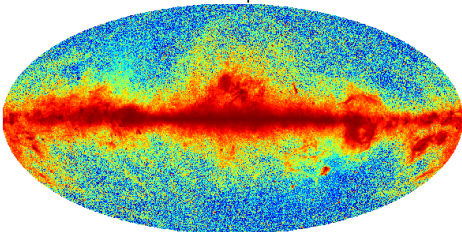
Visualize it!

What happen if you subtract them? [sub = mapa - mapa2]

Map data manipulation

Result:

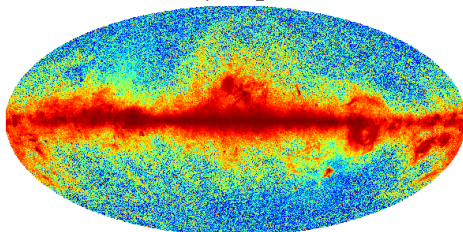
all multipoles



-0.000615499

0.185788

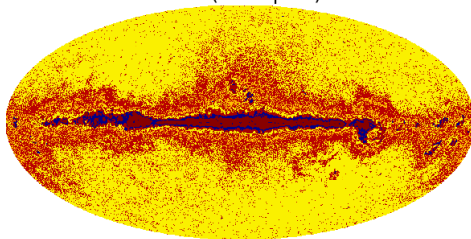
$l \geq 1$



-0.000620555

0.185783

$l = 0$ (monopole)



5.05606e-06

5.05606e-06

Treating the missing value

Exercise:

- Repeat the exercise removing **dipole + monopole**.
- Visualize the result.
- Verify what happen if you **cut** $\pm 20^\circ$ of the Galaxy region (use `gal_cut=20`).

Map data manipulation

Result:

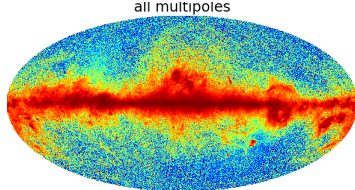
```
In [55]: mapa3 = hp.remove_dipole(mapa)
monopole: 5.05606e-06  dipole: lon: 1.28489, lat: 0.000777875,
amp: 0.000399906
```

```
In [55]:
```

```
In [56]: mapa3 = hp.remove_dipole(mapa, gal_cut= 20)
monopole: -0.000304967  dipole: lon: 0.276195, lat: 3.98544,
amp: 2.53603e-05
```

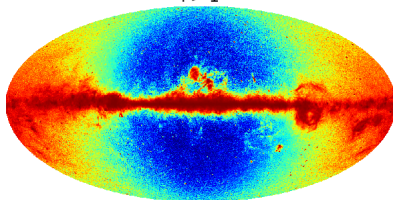
```
In [57]: |
```

Result:



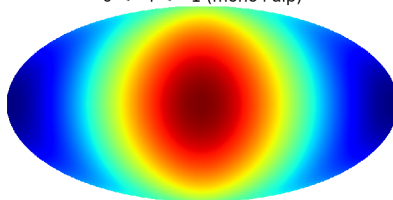
-0.000615499 0.185788

$l > 1$

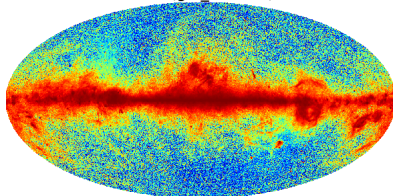


-0.000865407 0.185395
 $l > 1$ (gal_cut=20)

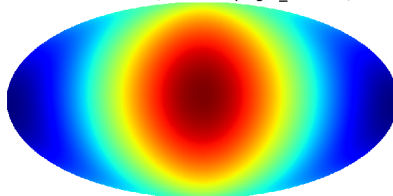
$0 \leq l \leq 1$ (mono+dip)



-0.000394847 0.000404958
 $0 \leq l \leq 1$ (mono+dip, gal_cut=20)



-0.000301964 0.186069



-0.000330327 -0.000279606

Map data manipulation

How to use:

```
In [30]: mapa3 = hp.remove_dipole(mapa, fitval = True)
monopole: 5.05606e-06  dipole: lon: 1.28489, lat: 0.000777875, amp: 0.000399906
```

```
In [31]: len(mapa3)
Out[31]: 3
```

```
In [32]: mapa3[0]
Out[32]:
masked_array(data = [-0.00038791 -0.00038744 -0.00038133 ..., -0.00032639
-0.00028807
-0.00043686],
             mask = False,
             fill_value = -1.6375e+30)
```

```
In [33]: mapa3[1]
Out[33]: 5.0560580580996753e-06
```

```
In [34]: mapa3[2]
Out[34]: array([ 3.99804999e-04,  8.96738494e-06,  5.42931005e-09])
```

```
In [35]: |
```