

```
In [1]: # -*- coding: utf-8 -*-  
  
import numpy as np  
  
from osgeo import gdal  
  
from osgeo import ogr  
  
from osgeo import osr  
  
from gdalconst import *  
  
import matplotlib.pyplot as plt  
  
import sys  
  
gdal.UseExceptions()  
  
dias_por_ano = range(1,366)  
anos = xrange(2010, 2016)  
  
Array_ages = []  
  
print (len(anos))
```

6

```
In [2]: # Definindo lista para o ano de 2014 entre os dias 98 e 365:
try:
    for y in anos:
        for x in dias_por_ano:
            filename = r'C:\Doutorado\1Trimestre\Geoprocessamento\LT_All_sat\MADT_eddy_ages\%04d\MADT_eddy_ages_dura_1
0_area_min_2_desv_05_%04d_dia%03d.tif' % (y, y, x)
            dataset = gdal.Open(filename, GA_ReadOnly)
            Array_ages.append(dataset.ReadAsArray())

        print("Arquivos abertos com sucesso!")

except RuntimeError, e:
    print e
    sys.exit("Erro na abertura de um arquivo")
    pass

np.array(Array_ages)
print(np.shape(Array_ages))
```

```
Arquivos abertos com sucesso!
Arquivos abertos com sucesso!
Arquivos abertos com sucesso!
Arquivos abertos com sucesso!
Arquivos abertos com sucesso!
Arquivos abertos com sucesso!
(2190L, 160L, 84L)
```

```
In [3]: Array_ages = np.asarray (Array_ages)
print np.min(Array_ages)
print np.mean(Array_ages)
print np.max(Array_ages)
```

```
-32768
-15591.6876624
66
```

```
In [4]: Array_2010 = Array_ages[0 : 365]
Array_2011 = Array_ages[365: 730]
Array_2012 = Array_ages[730 : 1095]
Array_2013 = Array_ages[1095 : 1460]
Array_2014 = Array_ages[1460 : 1825]
Array_2015 = Array_ages[1825 : 2190]
#Array_2016 = Array_ages[2190 : 2555]
```

```
In [5]: print(np.shape(Array_2010))
print(np.shape(Array_2011))
print(np.shape(Array_2012))
print(np.shape(Array_2013))
print(np.shape(Array_2014))
print(np.shape(Array_2015))
print ""

print type(Array_2010)
```

```
(365L, 160L, 84L)
(365L, 160L, 84L)
(365L, 160L, 84L)
(365L, 160L, 84L)
(365L, 160L, 84L)
(365L, 160L, 84L)
(365L, 160L, 84L)
```

```
<type 'numpy.ndarray'>
```

Duracao média anual dos vortices:

```
In [6]: Mean_Array_2010 = np.where(np.mean(Array_2010, axis = 0)<0, 0, np.mean(Array_2010, axis = 0))
Mean_Array_2011 = np.where(np.mean(Array_2011, axis = 0)<0, 0, np.mean(Array_2011, axis = 0))
Mean_Array_2012 = np.where(np.mean(Array_2012, axis = 0)<0, 0, np.mean(Array_2012, axis = 0))
Mean_Array_2013 = np.where(np.mean(Array_2013, axis = 0)<0, 0, np.mean(Array_2013, axis = 0))
Mean_Array_2014 = np.where(np.mean(Array_2014, axis = 0)<0, 0, np.mean(Array_2014, axis = 0))
Mean_Array_2015 = np.where(np.mean(Array_2015, axis = 0)<0, 0, np.mean(Array_2015, axis = 0))

if np.shape(Mean_Array_2010) == np.shape(Mean_Array_2011) == np.shape(Mean_Array_2012):
    print("Todos os anos apresentam mesma dimensão: {}".format(np.shape(Mean_Array_2010)))

    print ("Duração Máxima de todos os anos: {}".format(np.max(Array_ages)))
```

Todos os anos apresentam mesma dimensão: (160L, 84L)
Duração Máxima de todos os anos: 66

Duracao Média dos vortices:

```
In [7]: #Retirando valores abaixo de 0 das matrizes:

Array_2010_n = np.where (Array_2010>0.0,Array_2010,np.nan)
Array_2011_n = np.where (Array_2011>0.0,Array_2011,np.nan)
Array_2012_n = np.where (Array_2012>0.0,Array_2012,np.nan)
Array_2013_n = np.where (Array_2013>0.0,Array_2013,np.nan)
Array_2014_n = np.where (Array_2014>0.0,Array_2014,np.nan)
Array_2015_n = np.where (Array_2015>0.0,Array_2015,np.nan)

np.shape(Array_2015_n)
```

Out[7]: (365L, 160L, 84L)

```
In [8]: # Correcao dos valores netativos para 2010
print (np.shape(np.nanmean(Array_2010_n, axis =1)))

Array_2010_n = (Array_2010_n +32768)

Array_2010_n = np.where((Array_2010_n - 32768)<0, 0, (Array_2010_n - 32768))

print ("mínimo do Array 2010 normalizado corrigido 2010: %d" % np.nanmin(Array_2010_n))

print ("max do Array 2010 normalizado corrigido 2010: %d" % np.nanmax(Array_2010_n))

(365L, 84L)
mínimo do Array 2010 normalizado corrigido 2010: 1
max do Array 2010 normalizado corrigido 2010: 33

C:\Users\Philipe Leal\Anaconda2\lib\site-packages\numpy\lib\nanfunctions.py:675: RuntimeWarning: Mean of empty slice
  warnings.warn("Mean of empty slice", RuntimeWarning)
C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel\__main__.py:8: RuntimeWarning: invalid value encountered
  in less
```

In [9]: *# Correcao dos valores netativos para 2011*

```
print (np.shape(np.nanmean(Array_2011_n, axis =1)))
```

```
Array_2011_n = (Array_2011_n +32768)
```

```
Array_2011_n = np.where((Array_2011_n - 32768)<0, 0, (Array_2011_n - 32768))
```

```
print ("mínimo do Array 2011 normalizado corrigido 2011: %d" % np.nanmin(Array_2011_n))
```

```
print ("max do Array 2011 normalizado corrigido 2011: %d" % np.nanmax(Array_2011_n))
```

```
(365L, 84L)
```

```
mínimo do Array 2011 normalizado corrigido 2011: 1
```

```
max do Array 2011 normalizado corrigido 2011: 44
```

```
C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel\__main__.py:10: RuntimeWarning: invalid value encountered in less
```

In [10]: *# Correcao dos valores netativos para 2012*

```
print (np.shape(np.nanmean(Array_2012_n, axis =1)))
```

```
Array_2012_n = (Array_2012_n +32768)
```

```
Array_2012_n = np.where((Array_2012_n - 32768)<0, 0, (Array_2012_n - 32768))
```

```
print ("mínimo do Array 2012 normalizado corrigido 2012: %d" % np.nanmin(Array_2012_n))
```

```
print ("max do Array 2012 normalizado corrigido 2012: %d" % np.nanmax(Array_2012_n))
```

```
(365L, 84L)
```

```
C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel\__main__.py:10: RuntimeWarning: invalid value encountered in less
```

```
mínimo do Array 2012 normalizado corrigido 2012: 1
```

```
max do Array 2012 normalizado corrigido 2012: 36
```

In [11]: *# Correcao dos valores netativos para 2013*

```
print (np.shape(np.nanmean(Array_2013_n, axis =1)))
```

```
Array_2013_n = (Array_2013_n +32768)
```

```
Array_2013_n = np.where((Array_2013_n - 32768)<0, 0, (Array_2013_n - 32768))
```

```
print ("mínimo do Array 2013 normalizado corrigido 2013: %d" % np.nanmin(Array_2013_n))
```

```
print ("max do Array 2013 normalizado corrigido 2013: %d" % np.nanmax(Array_2013_n))
```

```
(365L, 84L)
```

```
C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel\__main__.py:10: RuntimeWarning: invalid value encountered in less
```

```
mínimo do Array 2013 normalizado corrigido 2013: 1
```

```
max do Array 2013 normalizado corrigido 2013: 51
```


In [12]: *# Correcao dos valores netativos para 2014*

```
print (np.shape(np.nanmean(Array_2014_n, axis =1)))
```

```
Array_2014_n = (Array_2014_n +32768)
```

```
Array_2014_n = np.where((Array_2014_n - 32768)<0, 0, (Array_2014_n - 32768))
```

```
print ("mínimo do Array 2014 normalizado corrigido 2014: %d" % np.nanmin(Array_2014_n))
```

```
print ("max do Array 2014 normalizado corrigido 2014: %d" % np.nanmax(Array_2014_n))
```

```
C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel\__main__.py:9: RuntimeWarning: invalid value encountered  
in less
```

```
(365L, 84L)
```

```
mínimo do Array 2014 normalizado corrigido 2014: 1
```

```
max do Array 2014 normalizado corrigido 2014: 63
```

In [13]: *# Correcao dos valores netativos para 2015*

```
print (np.shape(np.nanmean(Array_2015_n, axis =1)))
```

```
Array_2015_n = (Array_2015_n +32768)
```

```
Array_2015_n = np.where((Array_2015_n - 32768)<0, 0, (Array_2015_n - 32768))
```

```
print ("mínimo do Array 2015 normalizado corrigido 2015: %d" % np.nanmin(Array_2015_n))
```

```
print ("max do Array 2015 normalizado corrigido 2015: %d" % np.nanmax(Array_2015_n))
```

```
(365L, 84L)
```

```
C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel\__main__.py:10: RuntimeWarning: invalid value encountered in less
```

```
mínimo do Array 2015 normalizado corrigido 2015: 1
```

```
max do Array 2015 normalizado corrigido 2015: 66
```

In [14]: *#Geracao da lista da duracao de vortices acumulados. Notar que ainda falta dividir os respectivos valores pelo numero de vortices de cada dia.*

```
Lista_Duracao_paisagem_diario = [np.nansum((np.nansum(Array_2010_n, axis =1)), axis = 1),
                                np.nansum((np.nansum(Array_2011_n, axis =1)), axis = 1),
                                np.nansum((np.nansum(Array_2012_n, axis =1)), axis = 1),
                                np.nansum((np.nansum(Array_2013_n, axis =1)), axis = 1),
                                np.nansum((np.nansum(Array_2014_n, axis =1)), axis = 1),
                                np.nansum((np.nansum(Array_2015_n, axis =1)), axis = 1)]

np.asarray(Lista_Duracao_paisagem_diario)

print ("Dimensoes da Lista de Duracao diaria: {}".format(np.shape(Lista_Duracao_paisagem_diario)))
print ""
print ("Duração de vórtices totais: {}".format(Lista_Duracao_paisagem_diario))
```

Dimensoes da Lista de Duracao diaria: (6L, 365L)

```

Duração de vórtices totais: [array([ 10.,  14.,  21.,  32.,  40.,  60.,  70.,  72.,  99.,
 110., 121.,  87.,   6.,  13.,  21.,  31.,  47.,  63.,
   65.,  98., 101., 112., 112., 122., 106., 114., 107.,
 146., 136., 180., 171., 180., 168.,   0.,   0.,   0.,
   0.,   0.,   2.,   4.,   6.,   8.,  10.,  12.,  14.,
  16.,  18.,  43.,  59.,  81., 103., 113., 147., 158.,
 169., 205., 194., 218., 128.,  60.,  78.,  84.,  96.,
 108., 110., 110., 108., 120., 196.,  45.,   0.,   8.,
  16.,  38.,  48.,  79.,  98., 136., 166., 187., 140.,
  72.,  41.,  43.,  50.,  39.,  45.,  51.,  38.,  51.,
  46.,  80.,  88., 112.,  91.,  10.,  12.,  14.,  16.,
  18.,  22.,  28.,  33.,  38.,  43.,  50.,  59.,  33.,
  30.,  35.,  40.,  73.,  91.,  54.,  74.,  78.,  86.,
  50.,  56.,  41.,  36.,  49.,  30.,  35.,  30.,  54.,
  62.,  70.,  68.,  86.,  76., 105.,  90., 128., 132.,
 134., 138., 142., 140.,  95.,  70.,  79.,  88.,  97.,
 115.,  80.,  65.,  48.,  52.,  42.,  45.,  64.,   0.,
   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   2.,
   4.,   9.,  12.,  10.,  20.,  18.,  22.,  29.,  38.,
  24.,  34.,  41.,  48.,  55.,  62.,  69.,  56.,  61.,
  36.,  39.,  42.,  75.,  64.,  85.,  90., 114., 120.,
 168., 181., 179., 195.,  32.,  30.,  46.,  89., 114.,
 110., 145., 180., 200., 220., 238., 204., 176.,  34.,
  20.,  30.,  42.,  48.,  58.,  53.,  72.,  41.,  48.,
  61.,  69.,  85.,  85.,  42.,  46.,  26.,   2.,   4.,
   6.,   8.,  10.,  18.,  14.,  16.,  18.,  20.,  22.,
  36.,  39.,  42.,  49.,  74., 103.,  86., 109., 110.,
 162., 114., 136., 180., 171., 195., 213., 242., 239.,
 291., 273., 253., 162., 164., 174., 141., 158., 150.,
 151., 323., 213.,  18.,  24.,  30.,  46.,  64.,  75.,
  91.,  88., 102., 148., 159., 123., 124., 126.,  98.,
  48.,  47.,  36.,  12.,  18.,  24.,  40.,  54.,  63.,
  73.,  87., 102., 124., 152., 164., 198., 198., 146.,
 133., 154., 152., 186., 212., 261., 222., 162., 138.,
 110., 119.,  70.,  80.,  92.,  99., 106.,  45.,  50.,
  59.,  68.,  77., 104., 100.,  24.,  28.,  32.,  36.,
  52.,  39.,  51.,  59.,  67.,  81.,  88.,  98.,  72.,
  90.,  88., 108., 104.,  84.,  30.,   0.,   0.,   0.,
   0.,   0.,   0.,   0.,   0.,   0.,   4.,  18.,  30.,

```

```

36., 40., 42., 51., 62., 81., 88., 98., 96.,
105., 128., 153., 164., 153.]), array([ 162., 171., 160., 149., 162., 113., 120., 85., 65.,
78., 112., 153., 153., 159., 108., 130., 126., 135.,
144., 104., 62., 50., 60., 15., 20., 42., 40.,
63., 60., 55., 74., 69., 62., 68., 90., 114.,
138., 131., 102., 69., 86., 107., 120., 139., 118.,
130., 96., 95., 102., 90., 102., 92., 144., 125.,
153., 176., 213., 139., 134., 115., 111., 134., 147.,
112., 82., 102., 76., 76., 165., 88., 112., 156.,
128., 136., 165., 240., 230., 240., 250., 286., 327.,
234., 192., 174., 154., 160., 173., 195., 205., 179.,
234., 280., 328., 378., 481., 484., 239., 88., 0.,
0., 0., 0., 2., 6., 22., 40., 54., 80.,
100., 130., 166., 159., 161., 177., 183., 154., 143.,
66., 86., 99., 75., 14., 22., 32., 50., 81.,
89., 119., 136., 169., 242., 265., 292., 309., 321.,
261., 189., 207., 225., 240., 293., 236., 213., 258.,
42., 48., 16., 20., 24., 35., 40., 18., 20.,
0., 0., 0., 0., 0., 0., 0., 0., 0.,
0., 0., 0., 0., 0., 0., 0., 0., 0.,
0., 3., 10., 15., 34., 34., 36., 57., 52.,
38., 49., 76., 85., 112., 132., 133., 128., 172.,
214., 248., 210., 198., 167., 173., 141., 138., 104.,
42., 46., 50., 58., 68., 74., 21., 40., 36.,
40., 44., 38., 43., 51., 57., 52., 69., 75.,
108., 116., 124., 169., 163., 96., 66., 25., 30.,
14., 16., 18., 20., 22., 24., 26., 2., 4.,
6., 8., 10., 12., 16., 23., 40., 55., 76.,
72., 93., 101., 118., 125., 121., 102., 39., 18.,
14., 24., 29., 24., 28., 8., 10., 12., 14.,
16., 18., 20., 22., 24., 0., 0., 2., 12.,
21., 28., 35., 36., 42., 66., 76., 59., 78.,
53., 48., 33., 34., 54., 74., 57., 76., 87.,
80., 50., 57., 54., 38., 48., 53., 92., 90.,
116., 55., 60., 104., 70., 107., 116., 176., 188.,
181., 152., 161., 82., 27., 20., 22., 0., 0.,
0., 0., 0., 0., 0., 0., 0., 0., 0.,
0., 0., 0., 0., 0., 2., 6., 15., 20.,
30., 42., 49., 48., 54., 60., 46., 32., 15.,
20., 35., 36., 58., 65., 84., 94., 93., 105.,
136., 109., 136., 162., 145., 141., 146., 158., 155.,
160., 153., 144., 173., 204.]), array([ 216., 184., 171., 180., 71., 24., 18., 20., 22.,

```

```

0., 0., 2., 4., 6., 8., 10., 12., 28.,
32., 36., 60., 66., 62., 30., 6., 8., 10.,
12., 14., 16., 18., 20., 24., 28., 6., 8.,
10., 12., 14., 16., 18., 20., 22., 26., 4.,
12., 16., 25., 30., 42., 64., 72., 80., 91.,
84., 58., 77., 77., 39., 45., 47., 63., 81.,
75., 72., 81., 77., 84., 100., 98., 107., 78.,
72., 82., 59., 18., 28., 48., 54., 60., 70.,
136., 158., 158., 164., 170., 95., 123., 151., 172.,
187., 211., 174., 192., 272., 287., 270., 250., 224.,
209., 212., 262., 208., 195., 95., 33., 0., 0.,
2., 4., 9., 24., 35., 42., 52., 54., 51.,
62., 65., 60., 53., 118., 133., 182., 206., 219.,
231., 225., 205., 132., 94., 64., 76., 90., 116.,
148., 186., 195., 261., 318., 274., 233., 54., 58.,
66., 74., 82., 90., 98., 106., 114., 135., 129.,
139., 149., 146., 179., 148., 178., 186., 172., 194.,
130., 136., 158., 142., 153., 169., 153., 162., 126.,
123., 144., 168., 116., 51., 26., 31., 42., 39.,
38., 55., 29., 38., 49., 42., 49., 64., 72.,
80., 99., 108., 104., 98., 105., 112., 102., 92.,
99., 6., 8., 10., 12., 14., 16., 18., 20.,
22., 24., 0., 3., 8., 15., 20., 25., 36.,
49., 66., 67., 66., 45., 19., 30., 42., 48.,
67., 72., 75., 64., 71., 68., 33., 36., 39.,
72., 94., 73., 80., 87., 50., 21., 16., 18.,
20., 22., 24., 28., 32., 39., 44., 49., 48.,
52., 18., 49., 92., 119., 169., 186., 259., 256.,
257., 195., 247., 277., 271., 301., 270., 65., 69.,
42., 64., 79., 90., 102., 73., 83., 83., 80.,
99., 108., 117., 126., 135., 160., 170., 180., 190.,
160., 105., 0., 2., 8., 15., 28., 50., 48.,
63., 64., 54., 40., 36., 8., 12., 20., 25.,
30., 21., 16., 20., 24., 28., 15., 19., 30.,
43., 55., 66., 71., 68., 78., 102., 114., 140.,
119., 116., 105., 32., 36., 38., 48., 56., 33.,
48., 54., 64., 96., 114., 123., 120., 141., 136.,
97., 112., 138., 100., 107., 120., 128., 136., 180.,
190., 229., 204., 251., 248., 258., 203., 113., 123.,
24., 35., 42., 49., 56.]), array([ 74., 83., 80., 80., 46., 21., 24., 18., 20.,
22., 0., 0., 0., 0., 0., 0., 0., 0.,
2., 6., 15., 12., 17., 22., 44., 48., 48.,

```

```

58., 90., 112., 143., 108., 83., 52., 58., 64.,
53., 60., 70., 78., 65., 50., 62., 76., 83.,
102., 48., 30., 47., 71., 80., 75., 79., 90.,
101., 70., 36., 42., 30., 34., 16., 24., 38.,
39., 52., 65., 66., 84., 112., 117., 100., 90.,
88., 32., 15., 22., 33., 55., 79., 105., 110.,
139., 106., 122., 115., 139., 181., 185., 194., 196.,
133., 129., 150., 165., 196., 226., 180., 181., 133.,
120., 147., 71., 16., 18., 30., 22., 0., 0.,
0., 0., 0., 0., 2., 4., 12., 20., 25.,
30., 35., 48., 36., 30., 3., 6., 9., 12.,
15., 18., 28., 24., 42., 46., 61., 41., 52.,
63., 88., 101., 123., 145., 129., 129., 0., 4.,
14., 12., 32., 45., 54., 56., 48., 54., 70.,
66., 50., 64., 63., 36., 50., 60., 63., 80.,
99., 100., 110., 108., 104., 84., 90., 99., 74.,
90., 24., 35., 39., 45., 43., 44., 53., 47.,
50., 59., 52., 66., 87., 116., 133., 195., 236.,
206., 188., 120., 148., 136., 106., 98., 60., 72.,
57., 24., 18., 22., 8., 16., 23., 34., 48.,
50., 59., 69., 79., 89., 99., 99., 110., 163.,
177., 181., 203., 226., 246., 208., 222., 234., 252.,
239., 255., 234., 248., 265., 238., 220., 225., 276.,
165., 239., 194., 239., 212., 221., 230., 217., 204.,
210., 173., 134., 138., 142., 172., 179., 235., 249.,
178., 175., 88., 35., 26., 67., 108., 100., 141.,
149., 165., 132., 154., 116., 88., 132., 165., 218.,
283., 328., 315., 318., 370., 328., 304., 320., 340.,
418., 391., 416., 415., 180., 174., 174., 192., 198.,
204., 174., 179., 184., 228., 234., 199., 204., 252.,
258., 219., 180., 90., 94., 100., 111., 16., 30.,
42., 42., 58., 67., 82., 86., 99., 111., 112.,
116., 110., 76., 72., 85., 102., 76., 116., 84.,
96., 122., 136., 159., 147., 155., 69., 54., 30.,
44., 57., 71., 83., 105., 97., 132., 124., 90.,
90., 121., 144., 169., 154., 135., 144., 119., 126.,
133., 120., 126., 110., 138., 122., 185., 162., 261.,
314., 306., 324., 321., 313.]), array([ 328., 298., 320., 360., 346., 217., 151., 286., 183.,
201., 234., 303., 236., 346., 366., 552., 582., 667.,
730., 764., 677., 635., 600., 644., 740., 786., 733.,
679., 586., 460., 112., 145., 154., 190., 202., 243.,
294., 243., 218., 224., 198., 141., 140., 117., 126.,

```

```

152., 120., 117., 150., 94., 88., 67., 77., 69.,
 72., 103., 91., 129., 167., 198., 217., 244., 242.,
248., 285., 345., 340., 362., 347., 343., 381., 351.,
298., 172., 154., 231., 347., 394., 206., 262., 276.,
300., 236., 253., 305., 333., 381., 360., 380., 380.,
410., 425., 448., 388., 270., 280., 290., 270., 285.,
485., 525., 597., 415., 425., 377., 329., 283., 289.,
194., 88., 93., 98., 18., 30., 33., 36., 54.,
 48., 62., 68., 89., 110., 88., 94., 100., 104.,
117., 128., 139., 124., 167., 192., 192., 128., 99.,
145., 159., 173., 201., 228., 228., 280., 338., 341.,
277., 260., 232., 239., 268., 283., 283., 305., 310.,
267., 301., 370., 427., 424., 442., 460., 478., 452.,
383., 178., 90., 2., 4., 8., 12., 19., 24.,
 34., 48., 50., 68., 69., 60., 55., 66., 56.,
 48., 36., 20., 2., 4., 9., 12., 10., 14.,
 20., 26., 32., 38., 44., 26., 30., 34., 38.,
 47., 58., 44., 44., 52., 48., 54., 64., 33.,
 38., 21., 16., 18., 20., 22., 24., 26., 28.,
 30., 48., 55., 46., 58., 76., 88., 110., 90.,
112., 118., 138., 107., 96., 92., 48., 39., 42.,
 32., 40., 49., 24., 40., 48., 70., 80., 90.,
102., 83., 115., 110., 89., 83., 78., 54., 80.,
 70., 84., 64., 78., 81., 98., 107., 131., 129.,
122., 90., 85., 85., 55., 62., 69., 94., 137.,
172., 176., 225., 218., 221., 208., 208., 210., 199.,
126., 148., 132., 141., 152., 163., 174., 81., 67.,
 32., 51., 64., 74., 68., 69., 74., 58., 66.,
 84., 72., 79., 86., 93., 128., 139., 166., 181.,
198., 179., 154., 187., 201., 231., 247., 211., 226.,
222., 305., 358., 318., 405., 388., 414., 477., 366.,
453., 424., 446., 524., 489., 546., 481., 530., 479.,
488., 457., 401., 311., 226., 241., 222., 217., 208.,
190., 161., 231., 220., 265., 318., 329., 340., 349.,
390., 396., 414., 415., 436., 469., 432., 489., 462.,
310., 366., 330., 393., 430.]), array([ 444., 486., 630., 580., 585., 380., 380., 335., 301.,
419., 447., 271., 328., 322., 346., 355., 232., 105.,
 90., 94., 152., 158., 135., 90., 124., 128., 165.,
102., 105., 111., 117., 161., 246., 135., 135., 140.,
145., 150., 110., 92., 96., 4., 9., 16., 20.,
 30., 35., 32., 36., 20., 0., 0., 0., 2.,
 6., 9., 8., 15., 18., 23., 20., 33., 32.,

```



```

64., 90., 94., 109., 71., 67., 92., 89., 101.,
120., 124., 135., 137., 81., 58., 40., 0., 0.,
0., 0., 0., 0., 2., 4., 8., 14., 24.,
30., 29., 34., 39., 46., 22., 28., 38., 40.,
58., 36., 34., 62., 80., 91., 42., 32., 12.,
16., 20., 24., 28., 32., 42., 49., 58., 37.,
44., 56., 58., 65., 79., 79., 95., 88., 58.,
2., 4., 6., 8., 10., 12., 14., 16., 18.,
20., 24., 30., 37., 44., 46., 52., 58., 64.,
70., 36., 40., 20., 22., 24., 26., 28., 0.,
0., 2., 4., 12., 18., 31., 34., 46., 56.,
84., 111., 115., 88., 97., 96., 60., 40., 48.,
62., 75., 84., 86., 94., 102., 100., 109., 102.,
90., 72., 75., 78., 32., 32., 34., 49., 58.,
74., 56., 64., 64., 49., 54., 59., 64., 71.,
92., 130., 138., 165., 174., 202., 212., 199., 208.,
217., 202., 210., 218., 170., 146., 151., 60., 62.,
0., 0., 0., 2., 4., 9., 14., 19., 24.,
29., 47., 69., 83., 102., 99., 109., 119., 108.,
106., 102., 48., 53., 58., 65., 64., 78., 84.,
132., 144., 134., 132., 161., 174., 162., 130., 81.,
66., 99., 108., 113., 130., 160., 198., 101., 42.,
50., 89., 96., 110., 135., 147., 171., 172., 172.,
197., 216., 161., 98., 120., 90., 124., 98., 117.,
165., 188., 230., 244., 255., 248., 188., 218., 180.,
63., 54., 25., 30., 28., 32., 18., 20., 3.,
8., 13., 14., 22., 27., 32., 37., 42., 38.,
42., 70., 39., 44., 34., 38., 46., 15., 18.,
21., 16., 18., 20., 2., 4., 6., 8., 10.,
12., 14., 16., 18., 22., 4., 6., 8., 10.,
12., 14., 16., 18., 20., 22., 26., 30., 36.,
42., 48., 26., 33., 32., 55., 58., 57., 71.,
83., 95., 113., 109., 109., 140., 126., 84., 52.,
56., 30., 34., 14., 23.]]]

```

```
In [15]: #Polarização da Array Total

Array_polarity = []

try:
    for y in anos:
        for x in dias_por_ano:
            filename = r'C:\Doutorado\1Trimestre\Geoprocessamento\LT_All_sat\MADT_eddy_polarities\%04d\MADT_eddy_polarities_dura_10_area_min_2_desv_05_%04d_dia%03d.tif' % (y, y, x)
            dataset = gdal.Open(filename, GA_ReadOnly)
            Array_polarity.append(dataset.ReadAsArray())

            print("Arquivos abertos com sucesso!")

except RuntimeError, e:
    print e
    sys.exit("Erro na abertura de um arquivo")
    pass

np.array(Array_polarity)
print(np.shape(Array_polarity))
```

```
Arquivos abertos com sucesso!
Arquivos abertos com sucesso!
Arquivos abertos com sucesso!
Arquivos abertos com sucesso!
Arquivos abertos com sucesso!
Arquivos abertos com sucesso!
(2190L, 160L, 84L)
```

```
In [16]: Array_Age_polarizado = np.multiply(Array_polarity, Array_ages)
```

```
print np.shape(Array_Age_polarizado)
print ""
print (Array_Age_polarizado[0])
print ""
print np.min(Array_Age_polarizado[15])
print ""
print np.max(Array_Age_polarizado[15])
```

```
(2190L, 160L, 84L)
```

```
[[0 0 0 ..., 0 0 0]
 [0 0 0 ..., 0 0 0]
 [0 0 0 ..., 0 0 0]
 ...,
 [0 0 0 ..., 0 0 0]
 [0 0 0 ..., 0 0 0]
 [0 0 0 ..., 0 0 0]]
```

```
-5
```

```
4
```

```
In [17]: Array_2010_polarizado = Array_Age_polarizado[0 : 365]
Array_2011_polarizado = Array_Age_polarizado[365: 730]
Array_2012_polarizado = Array_Age_polarizado[730 : 1095]
Array_2013_polarizado = Array_Age_polarizado[1095 : 1460]
Array_2014_polarizado = Array_Age_polarizado[1460 : 1825]
Array_2015_polarizado = Array_Age_polarizado[1825 :2190]
```

```
print (np.shape(Array_2015_polarizado))
```

```
(365L, 160L, 84L)
```

Para Vórtices Ciclônicos

```
In [18]: Array_2010_ciclonico = np.where(Array_2010_polarizado<0, Array_2010_polarizado, np.nan)
Array_2011_ciclonico = np.where(Array_2011_polarizado<0, Array_2011_polarizado, np.nan)
Array_2012_ciclonico = np.where(Array_2012_polarizado<0, Array_2012_polarizado, np.nan)
Array_2013_ciclonico = np.where(Array_2013_polarizado<0, Array_2013_polarizado, np.nan)
Array_2014_ciclonico = np.where(Array_2014_polarizado<0, Array_2014_polarizado, np.nan)
Array_2015_ciclonico = np.where(Array_2015_polarizado<0, Array_2015_polarizado, np.nan)
```

```
In [19]: print np.shape(Array_2010_ciclonico)
(365L, 160L, 84L)
```

```
In [20]: Lista_Duracao_Ciclones_2010 = []
Lista_Duracao_Ciclones_2011 = []
Lista_Duracao_Ciclones_2012 = []
Lista_Duracao_Ciclones_2013 = []
Lista_Duracao_Ciclones_2014 = []
Lista_Duracao_Ciclones_2015 = []

try:
    try:
        for x in (dias_por_ano):
            Ciclones = np.nanmean(np.nansum(Array_2010_ciclonico[x<=365]))
            Lista_Duracao_Ciclones_2010.append(Ciclones)

            print("Contagem de Ciclones de 2010 concluidos com sucesso!")

    except RuntimeError, e:
        print e
        sys.exit("Erro na Contagem de Ciclones de 2010")
        pass

    try:
        for x in dias_por_ano:
            Ciclones = np.nanmean(Array_2011_ciclonico[x<=365])
            Lista_Duracao_Ciclones_2011.append(Ciclones)

            print("Contagem de Ciclones de 2011 concluidos com sucesso!")
```

```
except RuntimeError, e:
    print e
    sys.exit("Erro na Contagem de Ciclones de 2011")
    pass

try:
    for x in dias_por_ano:
        Ciclones = np.nanmean(Array_2012_ciclonico[x<=365])
        Lista_Duracao_Ciclones_2012.append(Ciclones)

    print("Contagem de Ciclones de 2012 concluidos com sucesso!")

except RuntimeError, e:
    print e
    sys.exit("Erro na Contagem de Ciclones de 2012")
    pass

try:
    for x in dias_por_ano:
        Ciclones = np.nanmean(Array_2013_ciclonico[x<=365])
        Lista_Duracao_Ciclones_2013.append(Ciclones)

    print("Contagem de Ciclones de 2013 concluidos com sucesso!")

except RuntimeError, e:
    print e
    sys.exit("Erro na Contagem de Ciclones de 2013")
    pass

try:
    for x in dias_por_ano:
        Ciclones = np.nanmean(Array_2014_ciclonico[x<=365])
        Lista_Duracao_Ciclones_2014.append(Ciclones)
```

```
    print("Contagem de Ciclones de 2014 concluidos com sucesso!")

except RuntimeError, e:
    print e
    sys.exit("Erro na Contagem de Ciclones de 2014")
    pass

try:
    for x in dias_por_ano:
        Ciclones = np.nanmean(Array_2015_ciclonico[x<=365])
        Lista_Duracao_Ciclones_2015.append(Ciclones)

    print("Contagem de Ciclones de 2015 concluidos com sucesso!")

except RuntimeError, e:
    print e
    sys.exit("Erro na Contagem de Ciclones de 2015")
    pass

print("Arquivos abertos com sucesso!")

except RuntimeError, e:
    print e
    sys.exit("Erro na Contagem de Ciclones de alguma data de algum ano")
    pass

print np.array(Lista_Duracao_Ciclones_2010)
print ""
print np.array(Lista_Duracao_Ciclones_2011)
print ""
print np.array(Lista_Duracao_Ciclones_2012)
print ""
print np.array(Lista_Duracao_Ciclones_2013)
print ""
print np.array(Lista_Duracao_Ciclones_2014)
```

```
print ""  
print np.array(Lista_Duracao_Ciclones_2015)
```



```
Contagem de Ciclones de 2010 concluidos com sucesso!  
Contagem de Ciclones de 2011 concluidos com sucesso!  
Contagem de Ciclones de 2012 concluidos com sucesso!
```

```
C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel\__main__.py:11: VisibleDeprecationWarning: using a boolean instead of an integer will result in an error in the future  
C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel\__main__.py:25: VisibleDeprecationWarning: using a boolean instead of an integer will result in an error in the future  
C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel\__main__.py:39: VisibleDeprecationWarning: using a boolean instead of an integer will result in an error in the future  
C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel\__main__.py:54: VisibleDeprecationWarning: using a boolean instead of an integer will result in an error in the future  
C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel\__main__.py:70: VisibleDeprecationWarning: using a boolean instead of an integer will result in an error in the future  
C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel\__main__.py:86: VisibleDeprecationWarning: using a boolean instead of an integer will result in an error in the future
```

Contagem de Ciclones de 2013 concluidos com sucesso!
Contagem de Ciclones de 2014 concluidos com sucesso!
Contagem de Ciclones de 2015 concluidos com sucesso!
Arquivos abertos com sucesso!

```
[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.]
```

```
[-19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19.
-19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19.
-19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19.
-19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19.
-19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19.
-19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19.
-19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19.
-19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19.
-19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19.
-19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19.
-19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19.
-19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19.
-19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19. -19.]
```



```
In [21]: Lista_Duracao_de_Ciclones = [Lista_Duracao_Ciclones_2010,  
                                     Lista_Duracao_Ciclones_2011,  
                                     Lista_Duracao_Ciclones_2012,  
                                     Lista_Duracao_Ciclones_2013,  
                                     Lista_Duracao_Ciclones_2014,  
                                     Lista_Duracao_Ciclones_2015]  
  
print Lista_Duracao_de_Ciclones[0]  
  
print Lista_Duracao_de_Ciclones[1]  
  
print Lista_Duracao_de_Ciclones[2]  
  
print Lista_Duracao_de_Ciclones[3]  
  
print Lista_Duracao_de_Ciclones[4]  
  
print Lista_Duracao_de_Ciclones[5]
```



```
In [24]: Lista_Duracao_Anticiclone_2010 = []
Lista_Duracao_Anticiclone_2011 = []
Lista_Duracao_Anticiclone_2012 = []
Lista_Duracao_Anticiclone_2013 = []
Lista_Duracao_Anticiclone_2014 = []
Lista_Duracao_Anticiclone_2015 = []

try:
    try:
        for x in dias_por_ano:
            Anticiclone = np.nanmean(Array_2010_anticiclonico[x<=365])
            Lista_Duracao_Anticiclone_2010.append(Anticiclone)

        print("Contagem de Anticiclone de 2010 concluidos com sucesso!")

    except RuntimeError, e:
        print e
        sys.exit("Erro na Contagem de Anticiclone de 2010")
        pass
```

```
try:
    for x in dias_por_ano:
        Anticiclone = np.nanmean(Array_2011_anticiclonico[x<=365])
        Lista_Duracao_Anticiclone_2011.append(Anticiclone)

    print("Contagem de Anticiclone de 2011 concluidos com sucesso!")

except RuntimeError, e:
    print e
    sys.exit("Erro na Contagem de Anticiclone de 2011")
    pass

try:
    for x in dias_por_ano:
        Anticiclone = np.nanmean(Array_2012_anticiclonico[x<=365])
        Lista_Duracao_Anticiclone_2012.append(Anticiclone)

    print("Contagem de Anticiclone de 2012 concluidos com sucesso!")

except RuntimeError, e:
    print e
    sys.exit("Erro na Contagem de Anticiclone de 2012")
    pass

try:
    for x in dias_por_ano:
        Anticiclone = np.nanmean(Array_2013_anticiclonico[x<=365])
        Lista_Duracao_Anticiclone_2013.append(Anticiclone)

    print("Contagem de Anticiclone de 2013 concluidos com sucesso!")

except RuntimeError, e:
    print e
    sys.exit("Erro na Contagem de Anticiclone de 2013")
    pass
```

```
try:
    for x in dias_por_ano:
        Anticiclone = np.nanmean(Array_2014_anticiclonico[x<=365])
        Lista_Duracao_Anticiclone_2014.append(Anticiclone)

    print("Contagem de Anticiclone de 2014 concluidos com sucesso!")

except RuntimeError, e:
    print e
    sys.exit("Erro na Contagem de Anticiclone de 2014")
    pass

try:
    for x in xrange(359):
        Anticiclone = np.nanmean(Array_2015_anticiclonico[x<=365])
        Lista_Duracao_Anticiclone_2015.append(Anticiclone)

    print("Contagem de Anticiclone de 2015 concluidos com sucesso!")

except RuntimeError, e:
    print e
    sys.exit("Erro na Contagem de Anticiclone de 2015")
    pass

print("Arquivos abertos com sucesso!")

except RuntimeError, e:
    print e
    sys.exit("Erro na Contagem de Anticiclone de alguma data de algum ano")
    pass
```

Contagem de Anticiclone de 2010 concluidos com sucesso!

C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel__main__.py:11: VisibleDeprecationWarning: using a boolean instead of an integer will result in an error in the future

C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel__main__.py:25: VisibleDeprecationWarning: using a boolean instead of an integer will result in an error in the future

C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel__main__.py:39: VisibleDeprecationWarning: using a boolean instead of an integer will result in an error in the future

C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel__main__.py:54: VisibleDeprecationWarning: using a boolean instead of an integer will result in an error in the future

Contagem de Anticiclone de 2011 concluidos com sucesso!

Contagem de Anticiclone de 2012 concluidos com sucesso!

Contagem de Anticiclone de 2013 concluidos com sucesso!

Contagem de Anticiclone de 2014 concluidos com sucesso!

Contagem de Anticiclone de 2015 concluidos com sucesso!

Arquivos abertos com sucesso!

C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel__main__.py:70: VisibleDeprecationWarning: using a boolean instead of an integer will result in an error in the future

C:\Users\Philipe Leal\Anaconda2\lib\site-packages\ipykernel__main__.py:86: VisibleDeprecationWarning: using a boolean instead of an integer will result in an error in the future

```
In [25]: Lista_scatter_diario = [Lista_Duracao_Anticiclone_2010,
                               Lista_Duracao_Anticiclone_2011,
                               Lista_Duracao_Anticiclone_2012,
                               Lista_Duracao_Anticiclone_2013,
                               Lista_Duracao_Anticiclone_2014,
                               Lista_Duracao_Anticiclone_2015]
```

```
In [26]: print Lista_scatter_diario[0]
         print Lista_scatter_diario[1]
         print Lista_scatter_diario[2]
         print Lista_scatter_diario[3]
         print Lista_scatter_diario[4]
         print Lista_scatter_diario[5]
```


