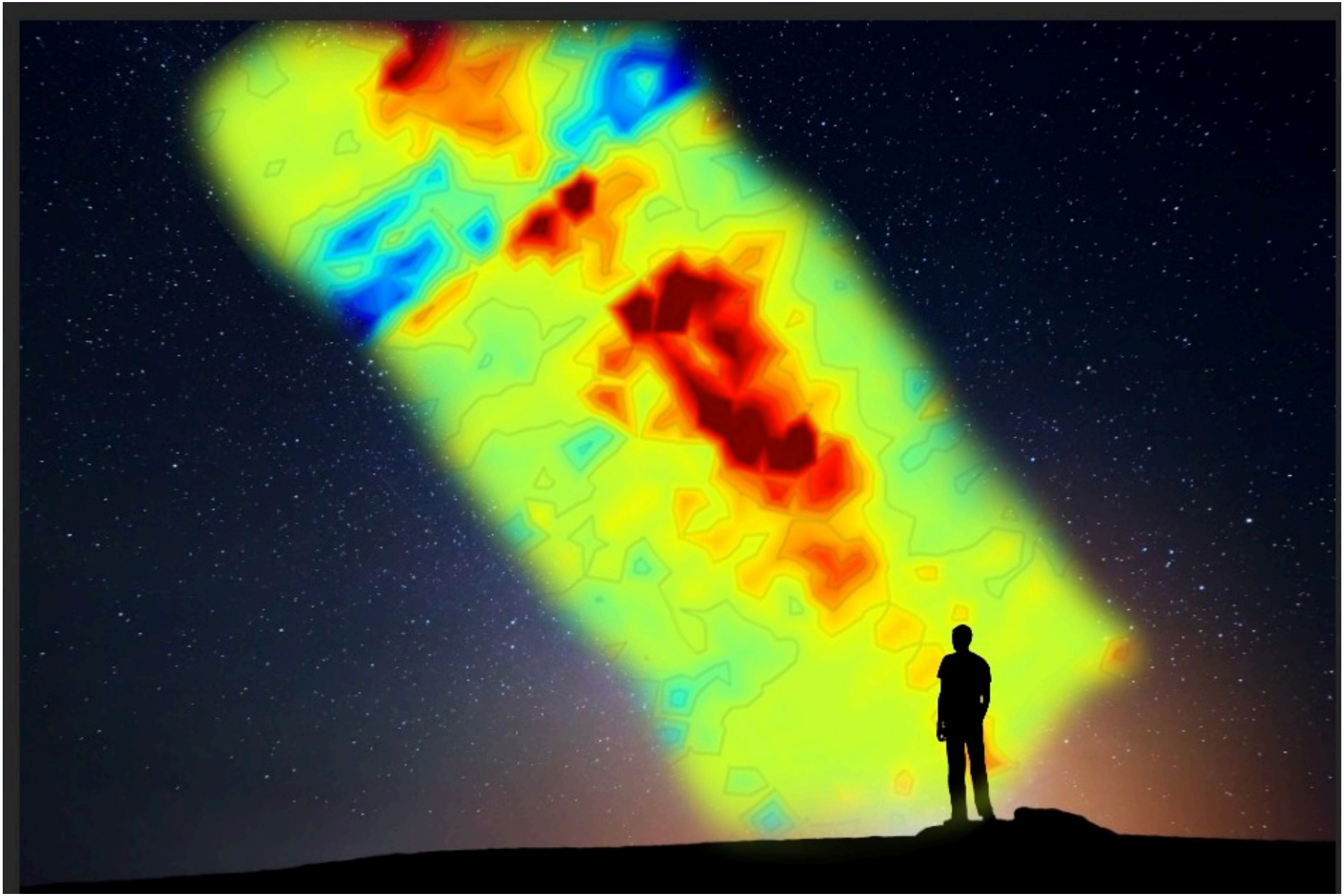
# The Big Milky Way Galaxy

A project to capture a huge part of the Milky Way galaxy and beyond with the J.R.T. (Job's Radio Telescope)

By Job Geheniau
jobgeheniau@gmail.com

On a dark clear night you can see the Milky Way, the Galaxy in which we live.

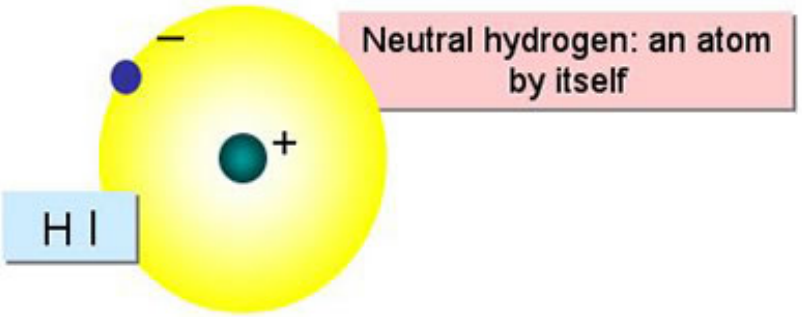
This document will explain how I got to this:



# We see the Milky Way in visible light.
# But it also emits Neutral Hydrogen.

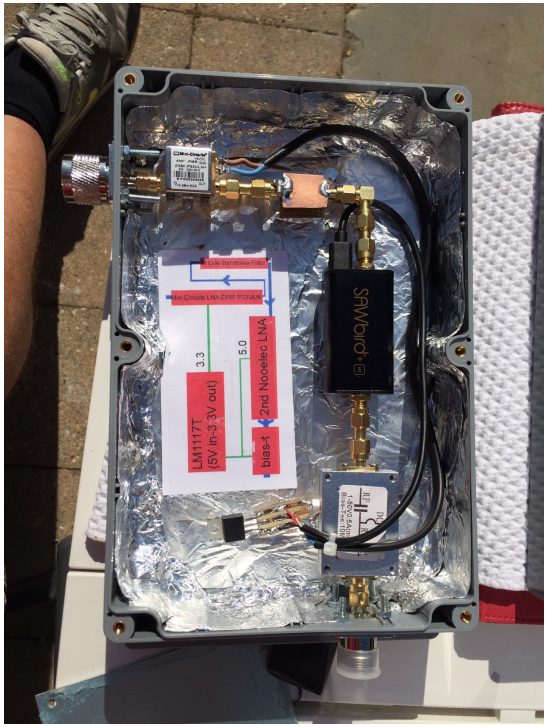
Neutral hydrogen is a normal hydrogen atom with one proton and one electron.

Neutral hydrogen is a normal, electrically neutral hydrogen atom with one proton and one electron. It is commonly referred to as HI (pronounced H-one), and is located throughout galaxies as HI clouds or external to galaxies as part of the intercloud gas . It is detected via the spin-flip transition at 21cm in the radio, and HI clouds were used to determine the structure of our Galaxy from our location within it.



# The detection of neutral Hydrogen we do with a Radio Telescope tuned at 1.420 Ghz

My FULLY REMOTE controlled Radio Telescope is 1.5 meter
with 2 LNA's and a filter.
Capture Software is partly developed by myself in Python



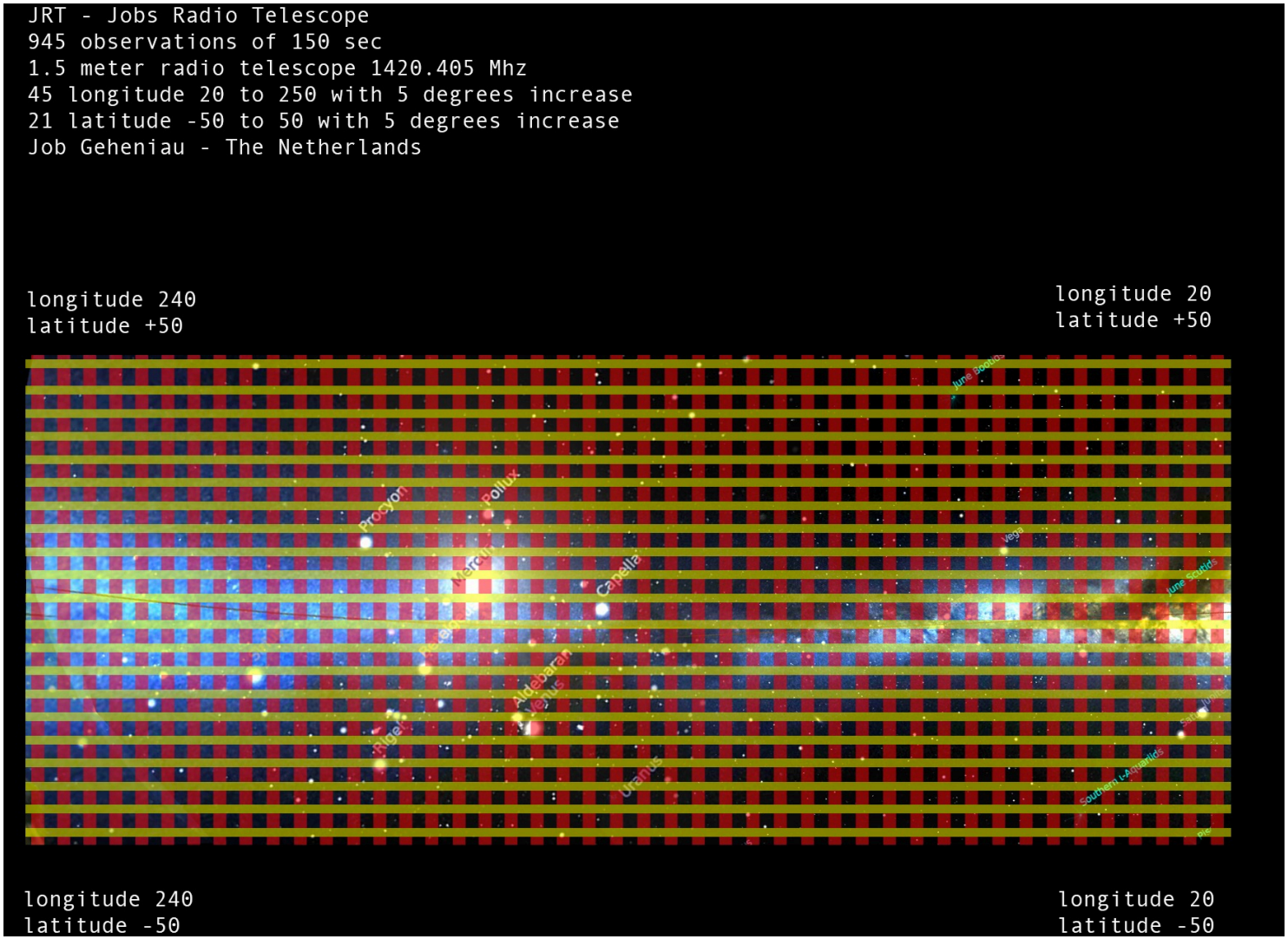
# The idea was to capture a big part of the Galaxy, almost fully automated.

So first I had to rewrite the software. That means …… learn Python

```
# Fill Excel column with all the max values


import sys
import os
import datetime
from time import sleep

os.system('cls' if os.name == 'nt' else 'clear')

print('\n*****************************************************************************************')
sleep(0.3)
print('TTTTTTT  OOOOOOO  TTTTTTT  AAAAAAA  L')
print('   T     O     O     T     A     A  L')
print('   T     O     O     T     AAAAAAA  L')
print('   T     O     O     T     A     A  L')
print('   T     OOOOOOO     T     A     A  LLLLLLL')
print('')
print('Find total of all values in nrad.tct files and place them in Excel .csv file')
print('')
print('developer Job Geheniau')
print('')
print('')
print('Version 1.0 june 2020')
sleep(0.3)
print('*****************************************************************************************\n')


print('[*] Please enter your desired observation parameters...\n')
sleep(0.5)

#Input observation parameters
observations = str(input('How many observations ?: '))

total = 0

# number of observations

frcycle = 1#
tocycle = int(observations)#8# BELANGRIJKE WAARDE


for i in range (0,(tocycle)) :
    with open('nrad-'+str(i)+'.tct', 'r') as inp, open('total'+str(i)+'.txt', 'w') as outp:
        for line in inp:
            try:
                num = float(line)
                total += num
                outp.write(line)
            except ValueError:
                print('{} is not a number!'.format(line))

    print('Total of all numbers nrad  '+str(i)+' {}'.format(total))
    f = open('total'+str(i)+'.txt', 'w')
    f.write('{}'.format(total))
    f.close()

    total = 0

import numpy as np
with open('nradtotal.csv', 'w') as f:
    for i in range (0,tocycle) :
        a = np.loadtxt("total"+str(i)+".txt")
        print i

        tem=str(a)
        temp=tem.replace('[', ' ')
        temp=temp.replace(']', ' ')
        f.write(temp+'\r\n')#CR, LF
f.close()
```

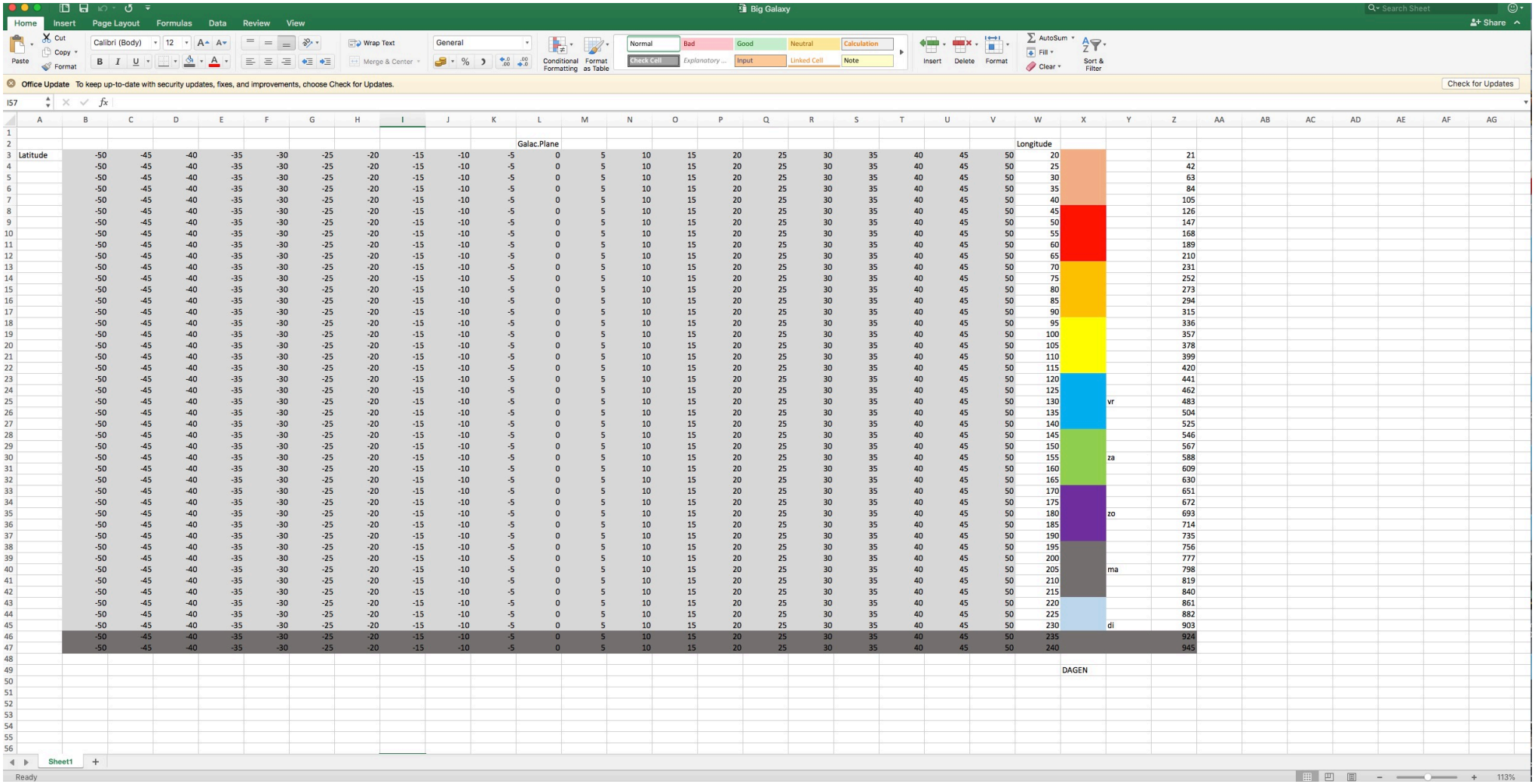
# The Matrix

Then determine with a matrix which part of the galaxy I wanted to record. The part which is visible on my location within the matrix is roughly between latitude -50 to 50 and longitude 20 to 230



# THAT IS 903 EXPOSURES OF 150 SECONDS EACH !

## So I made a longitude latitude Excel matrix



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | Galac.Plane | | | | | | | | | | | Longitude | | | |
| 3 | Latitude | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 20 | | | 21 |
| 4 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 25 | | | 42 |
| 5 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 30 | | | 63 |
| 6 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 35 | | | 84 |
| 7 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 40 | | | 105 |
| 8 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 45 | | | 126 |
| 9 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 50 | | | 147 |
| 10 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 55 | | | 168 |
| 11 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 60 | | | 189 |
| 12 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 65 | | | 210 |
| 13 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 70 | | | 231 |
| 14 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 75 | | | 252 |
| 15 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 80 | | | 273 |
| 16 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 85 | | | 294 |
| 17 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 90 | | | 315 |
| 18 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 95 | | | 336 |
| 19 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 100 | | | 357 |
| 20 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 105 | | | 378 |
| 21 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 110 | | | 399 |
| 22 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 115 | | | 420 |
| 23 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 120 | | | 441 |
| 24 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 125 | | | 462 |
| 25 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 130 | | vr | 483 |
| 26 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 135 | | | 504 |
| 27 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 140 | | | 525 |
| 28 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 145 | | | 546 |
| 29 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 150 | | | 567 |
| 30 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 155 | | za | 588 |
| 31 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 160 | | | 609 |
| 32 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 165 | | | 630 |
| 33 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 170 | | | 651 |
| 34 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 175 | | | 672 |
| 35 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 180 | | zo | 693 |
| 36 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 185 | | | 714 |
| 37 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 190 | | | 735 |
| 38 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 195 | | | 756 |
| 39 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 200 | | | 777 |
| 40 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 205 | | ma | 798 |
| 41 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 210 | | | 819 |
| 42 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 215 | | | 840 |
| 43 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 220 | | | 861 |
| 44 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 225 | | | 882 |
| 45 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 230 | | di | 903 |
| 46 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 235 | | | 924 |
| 47 | | -50 | -45 | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 240 | | | 945 |
| 48 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 49 | | | | | | | | | | | | | | | | | | | | | | | | DAGEN | | |

# The write a new Python script

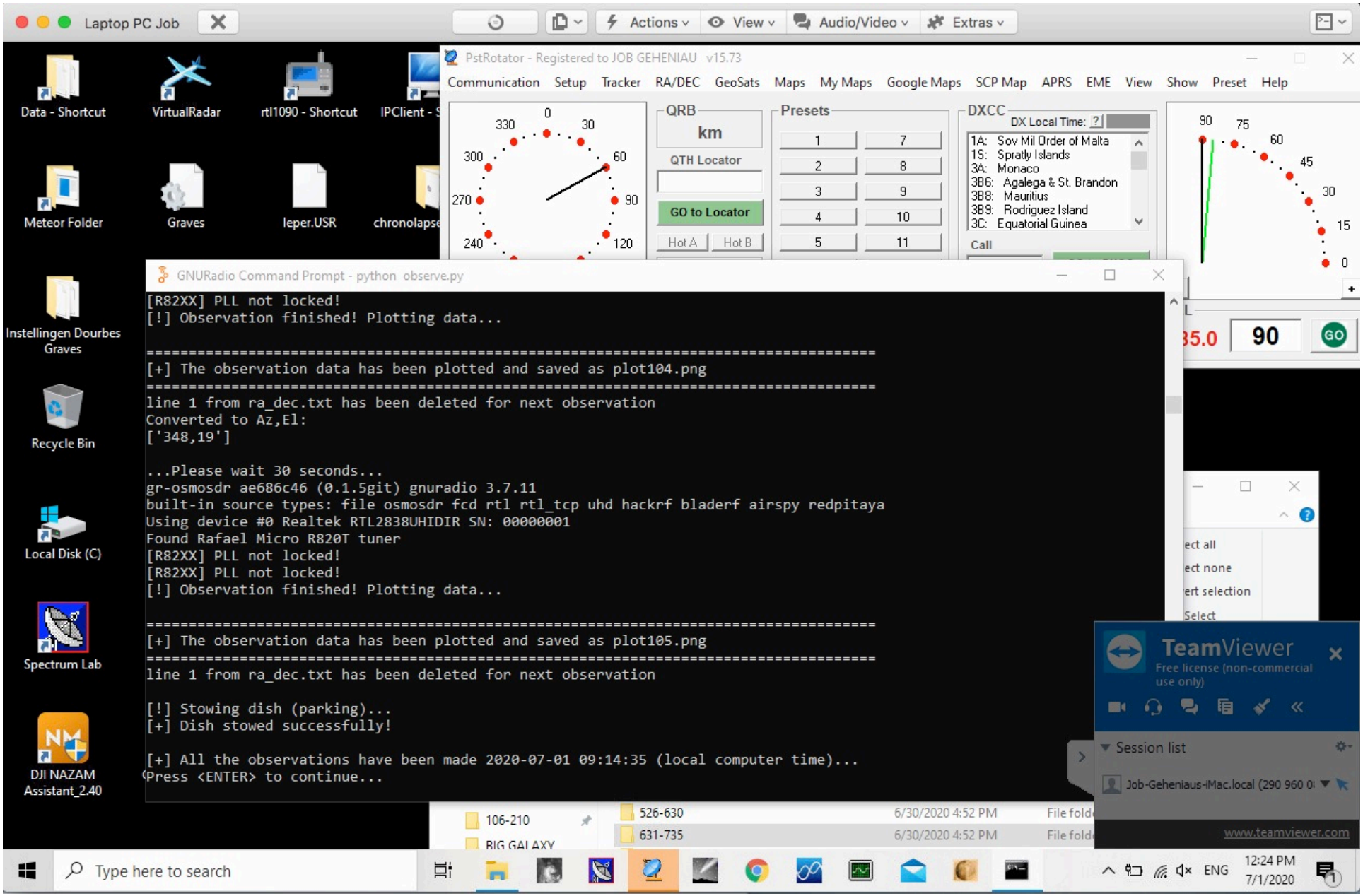
This one converts LONG LAT to RA DEC and then to ALT AZ When the observation is to be done.

```
import os
import sys
import datetime
from time import sleep
import socket

os.system('cls' if os.name == 'nt' else 'clear')

raw_input("Press Enter to continue and delete the old ra_dec.txt, ELSE press Control Z")

f = open("ra_dec.txt", "a")
sleep(1)
os.remove("ra_dec.txt")

print('\n*****************************************************************************************')
sleep(0.3)
print('CCCCCCC  V       V ')
print('C         V     V ')
print('C          V   V ')
print('C           V V    ')
print('CCCCCCC      V ')
print('')
print('CONVERT: convert Galactic Longitude Latitude to RA DEC (in xxhxxm0s)')
print('')
print('by Job Geheniau')
print('')
print('')
print('Version 1.0 june 2020')
sleep(0.3)
print('*****************************************************************************************\n')
sleep(1)
print('[*] Please enter your desired parameters...\n')
sleep(0.5)

#Input parameters

yea = str(input('Enter Epoch Year for coordinates : '))
year= str('J'+yea)
totalnumbers = int(input('How many long/lat coordinates do you want to enter: '))

for x in range(0, totalnumbers):

    long = int(input('Longitude [Degrees]: '))
    lat = int(input('Latitude [Degrees]: '))


    # INPUT LONGITUDE LATITUDE IN DEGREES

    c = SkyCoord(l=long*u.degree, b=lat*u.degree, frame='galactic')

    #print c.fk5
    #print c.transform_to(FK5(equinox='J2020'))


    ra_degree = float((str(c.transform_to(FK5(equinox=year)))[58:68])) # RA IN DEGREES FROM STRING
    dec_degree_to_hms = float((str(c.transform_to(FK5(equinox=year)))[71:78])) # DEC IN DEGREES FROM STRING

    # CONVERT RA DEGREES TO HOUR
    d = SkyCoord(ra=ra_degree*u.degree, dec=0*u.degree) #DEC is NOT important here only ra.

    #print d.ra.hms

    convert_ra_to_string = str(d.ra.hour)
    convert_ra_to_H = convert_ra_to_string.replace(".", "H")

    convert_dec_to_string = str(dec_degree_to_hms)
    convert_dec_to_H = convert_dec_to_string.replace(".", "H")



    # CONVERT EVERYTHING TO xxhxxm0s

    rafinal2=convert_ra_to_H[convert_ra_to_H.find("H")+1:].split()[0]

    rafinal1=int(rafinal2[0:2])

    rafinal3=(rafinal1*60)/100
    rafinal=str(rafinal3)
    #print convert_ra_to_H.split('H')[0]+'d'+rafinal[0:2]+'m0s'

    decfinal2=convert_dec_to_H[convert_dec_to_H.find("H")+1:].split()[0]

    decfinal1=int(decfinal2[0:2])
    decfinal3=(decfinal1*60)/100
    decfinal=str(decfinal3)
    #print convert_dec_to_H.split('H')[0]+'d'+decfinal[0:2]+'m0s'


    f = open("ra_dec.txt", "a")
    f.write((convert_ra_to_H.split('H')[0]+'h'+rafinal[0:2]+'m0s')+','+convert_dec_to_H.split('H')[0]+'d'+decfinal[0:2]+'m0s'+'\n')
    f.close()

print('**** Done ****')
```

# Time to run the software and have a sleep

And hopefully it has gathered the data the next morning

Every night I did 105 exposures with a total 8 nights and mornings



# Finally I had 903 Signal to Noise text files which contained the data for each exposure

| | | | |
|---|---|---|---|
| snr288.txt | 1 Jul 2020 at 07:44 | 15 KB | Plain Text |
| snr289.txt | 1 Jul 2020 at 07:47 | 15 KB | Plain Text |
| snr290.txt | 1 Jul 2020 at 07:51 | 15 KB | Plain Text |
| snr291.txt | 1 Jul 2020 at 07:54 | 15 KB | Plain Text |
| snr292.txt | 1 Jul 2020 at 07:57 | 15 KB | Plain Text |
| snr293.txt | 1 Jul 2020 at 08:01 | 16 KB | Plain Text |
| snr294.txt | 1 Jul 2020 at 08:04 | 15 KB | Plain Text |
| snr295.txt | 1 Jul 2020 at 08:07 | 16 KB | Plain Text |
| snr296.txt | 1 Jul 2020 at 08:11 | 16 KB | Plain Text |
| snr297.txt | 1 Jul 2020 at 08:14 | 15 KB | Plain Text |
| snr298.txt | 1 Jul 2020 at 08:17 | 15 KB | Plain Text |
| snr299.txt | 1 Jul 2020 at 08:21 | 15 KB | Plain Text |
| snr300.txt | 1 Jul 2020 at 08:24 | 15 KB | Plain Text |
| snr301.txt | 1 Jul 2020 at 08:27 | 16 KB | Plain Text |
| snr302.txt | 1 Jul 2020 at 08:31 | 15 KB | Plain Text |
| snr303.txt | 1 Jul 2020 at 08:34 | 15 KB | Plain Text |
| snr304.txt | 1 Jul 2020 at 08:37 | 15 KB | Plain Text |
| snr305.txt | 1 Jul 2020 at 08:41 | 15 KB | Plain Text |
| snr306.txt | 1 Jul 2020 at 08:44 | 15 KB | Plain Text |
| snr307.txt | 1 Jul 2020 at 08:47 | 15 KB | Plain Text |
| snr308.txt | 1 Jul 2020 at 08:51 | 15 KB | Plain Text |
| snr309.txt | 1 Jul 2020 at 08:54 | 15 KB | Plain Text |
| snr310.txt | 1 Jul 2020 at 08:57 | 15 KB | Plain Text |
| snr311.txt | 1 Jul 2020 at 09:01 | 15 KB | Plain Text |
| snr312.txt | 1 Jul 2020 at 09:04 | 16 KB | Plain Text |
| snr313.txt | 1 Jul 2020 at 09:07 | 16 KB | Plain Text |
| snr314.txt | 1 Jul 2020 at 09:11 | 15 KB | Plain Text |
| snr315.txt | 1 Jul 2020 at 09:14 | 16 KB | Plain Text |
| snr316.txt | 2 Jul 2020 at 05:50 | 16 KB | Plain Text |
| snr317.txt | 2 Jul 2020 at 05:53 | 16 KB | Plain Text |
| snr318.txt | 2 Jul 2020 at 05:56 | 16 KB | Plain Text |
| snr319.txt | 2 Jul 2020 at 06:00 | 15 KB | Plain Text |
| snr320.txt | 2 Jul 2020 at 06:03 | 15 KB | Plain Text |
| snr321.txt | 2 Jul 2020 at 06:06 | 16 KB | Plain Text |
| snr322.txt | 2 Jul 2020 at 06:09 | 16 KB | Plain Text |
| snr323.txt | 2 Jul 2020 at 06:13 | 16 KB | Plain Text |
| snr324.txt | 2 Jul 2020 at 06:16 | 16 KB | Plain Text |
| snr325.txt | 2 Jul 2020 at 06:19 | 15 KB | Plain Text |
| snr326.txt | 2 Jul 2020 at 06:23 | 15 KB | Plain Text |
| snr327.txt | 2 Jul 2020 at 06:26 | 15 KB | Plain Text |
| snr328.txt | 2 Jul 2020 at 06:29 | 15 KB | Plain Text |
| snr329.txt | 2 Jul 2020 at 06:32 | 15 KB | Plain Text |
| snr330.txt | 2 Jul 2020 at 06:36 | 15 KB | Plain Text |
| snr331.txt | 2 Jul 2020 at 06:39 | 15 KB | Plain Text |
| snr332.txt | 2 Jul 2020 at 06:42 | 15 KB | Plain Text |
| snr333.txt | 2 Jul 2020 at 06:45 | 15 KB | Plain Text |
| snr334.txt | 2 Jul 2020 at 06:49 | 15 KB | Plain Text |
| snr335.txt | 2 Jul 2020 at 06:52 | 15 KB | Plain Text |
| snr336.txt | 2 Jul 2020 at 06:55 | 15 KB | Plain Text |
| snr337.txt | 2 Jul 2020 at 06:58 | 15 KB | Plain Text |

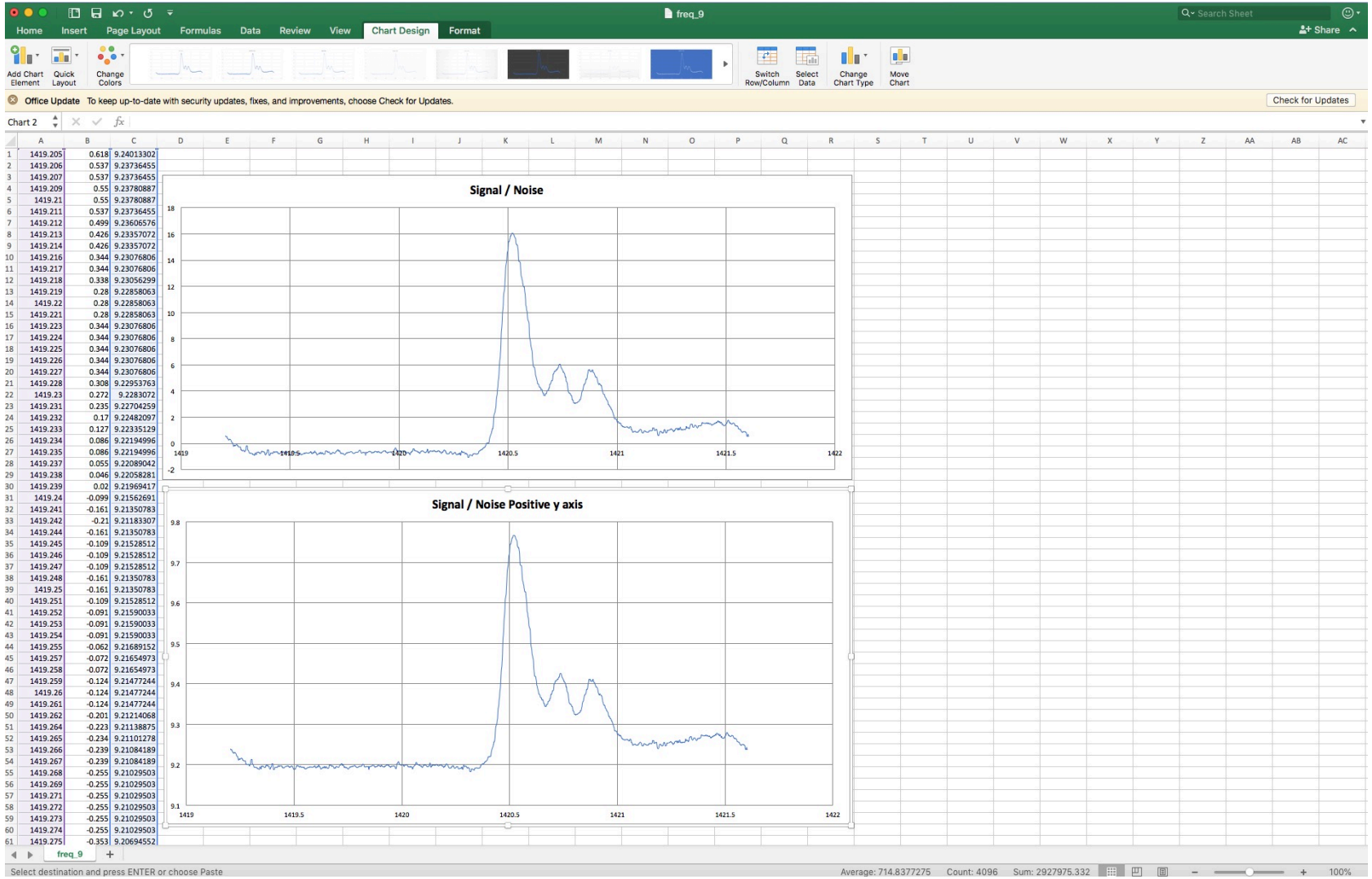
# Time to Process the Data !

I have processed all the data in different ways.

1) With a Pyhon script which generates the graphics
2) Manual processing in Excel with a 2 color mode (Black White)
3) Manual processing in Excel with a 3 color mode (the higher values more visible)
4) A ‘Mollweide’ Cartographic representation of the 3 Graphs

# First I get rid of negative data by transforming it with Python

Here you see 1 exposure the original and the new one without negative values.
The peaks are the TYPICAL Hydrogen where I am looking for in all the exposures….



# Add all the data PER Sample

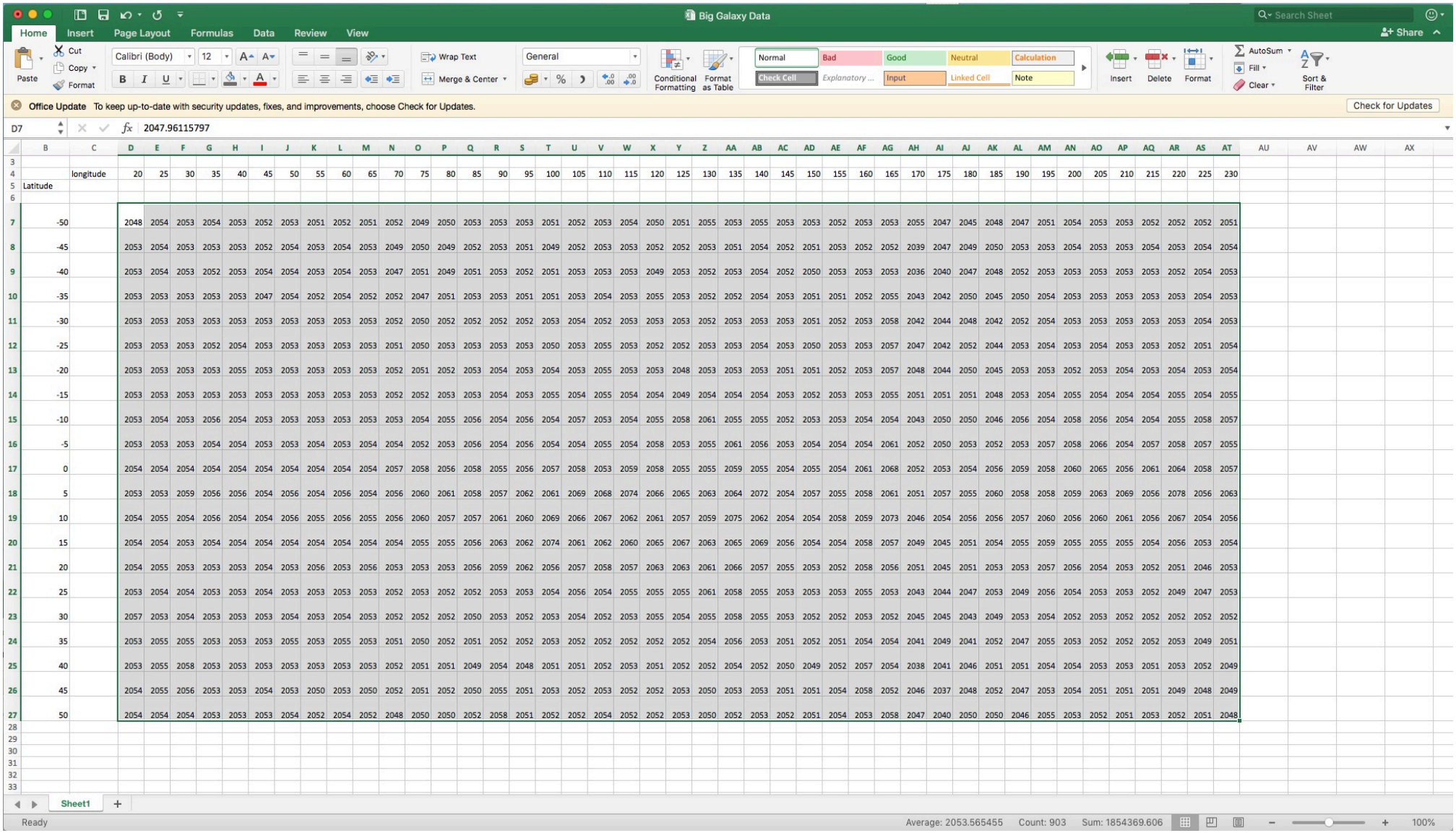
Because there is more Hydrogen than 1 peak I add all the data to 1 number.
So Finally I get 903 different numbers which represent the TOTAL of the data per exposure.

fill_total.py

```
import sys
import os
import datetime
from time import sleep

os.system('cls' if os.name == 'nt' else 'clear')

print('\n*****************************************************************************************')
sleep(0.3)
print('TTTTTTT  OOOOOOO  TTTTTTT  AAAAAAA  L')
print('   T     O     O     T     A     A  L')
print('   T     O     O     T     AAAAAAA  L')
print('   T     O     O     T     A     A  L')
print('   T     OOOOOOO     T     A     A  LLLLLLL')
print('')
print('Find total of all values in nrad.tct files and place them in Excel .csv file')
print('')
print('developer Job Geheniau')
print('')
print('')
print('Version 1.0 june 2020')
sleep(0.3)
print('*****************************************************************************************\n')


print('[*] Please enter your desired observation parameters...\n')
sleep(0.5)

#Input observation parameters
observations = str(input('How many observations ?: '))

total = 0

# number of observations

frcycle = 1#
tocycle = int(observations)#8# BELANGRIJKE WAARDE


for i in range (0,(tocycle)) :
    with open('nrad-'+str(i)+'.tct', 'r') as inp, open('total'+str(i)+'.txt', 'w') as outp:
        for line in inp:
            try:
                num = float(line)
                total += num
                outp.write(line)
            except ValueError:
                print('{} is not a number!'.format(line))

    print('Total of all numbers nrad  '+str(i)+' {}'.format(total))
    f = open('total'+str(i)+'.txt', 'w')
    f.write('{}'.format(total))
    f.close()

    total = 0

import numpy as np
with open('nradtotal.csv', 'w') as f:
    for i in range (0,tocycle) :
        a = np.loadtxt("total"+str(i)+".txt")
        print i

        tem=str(a)
        temp=tem.replace('[', ' ')
        temp=temp.replace(']', ' ')
        f.write(temp+'\r\n')#CR, LF
f.close()
print"done"
```

# Manual Data Processing in Excel

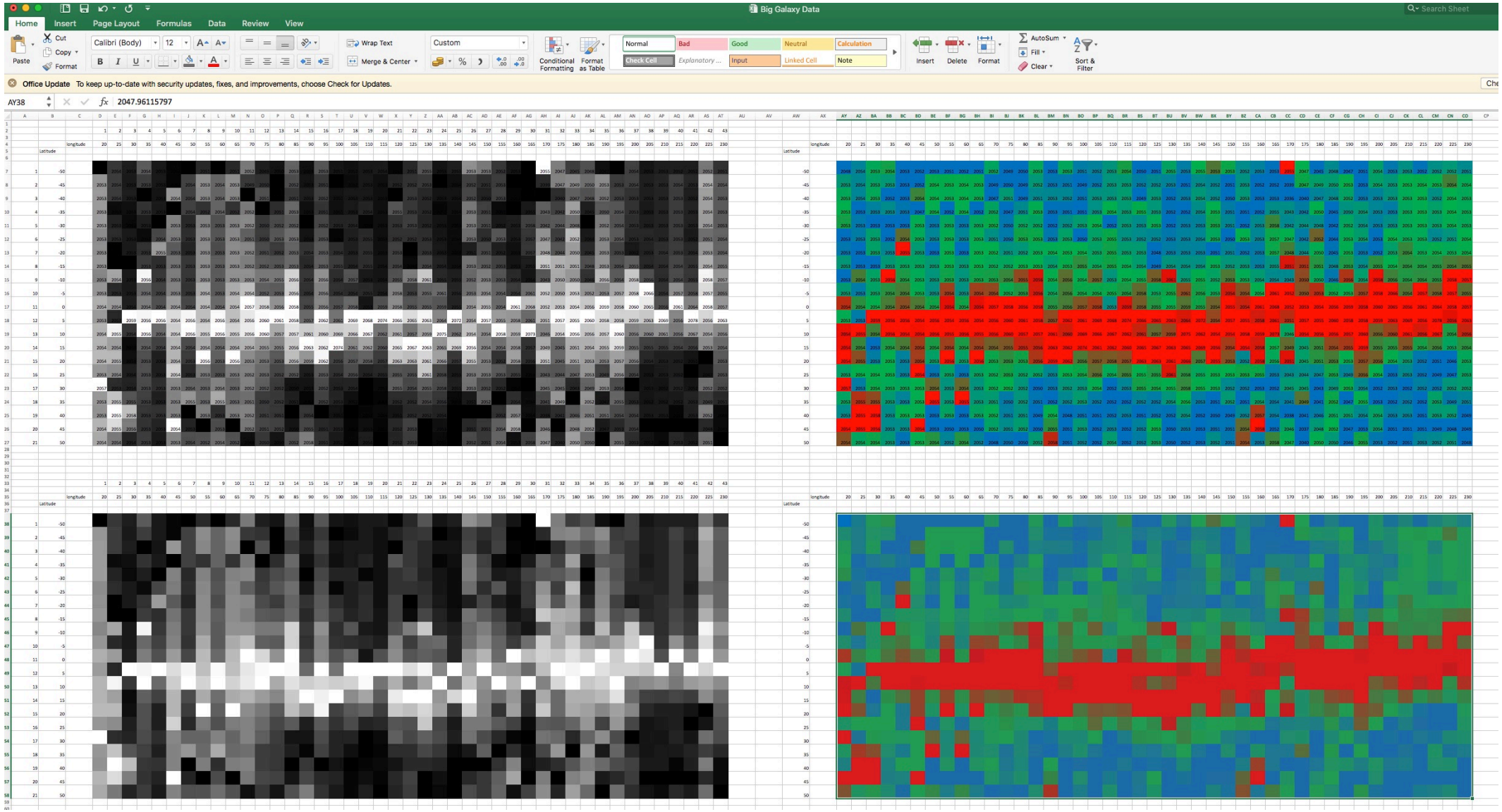
We have 21 latitude exposures per longitude.
We have 43 longitude exposures
We put those 903 numbers in an Excel Sheet



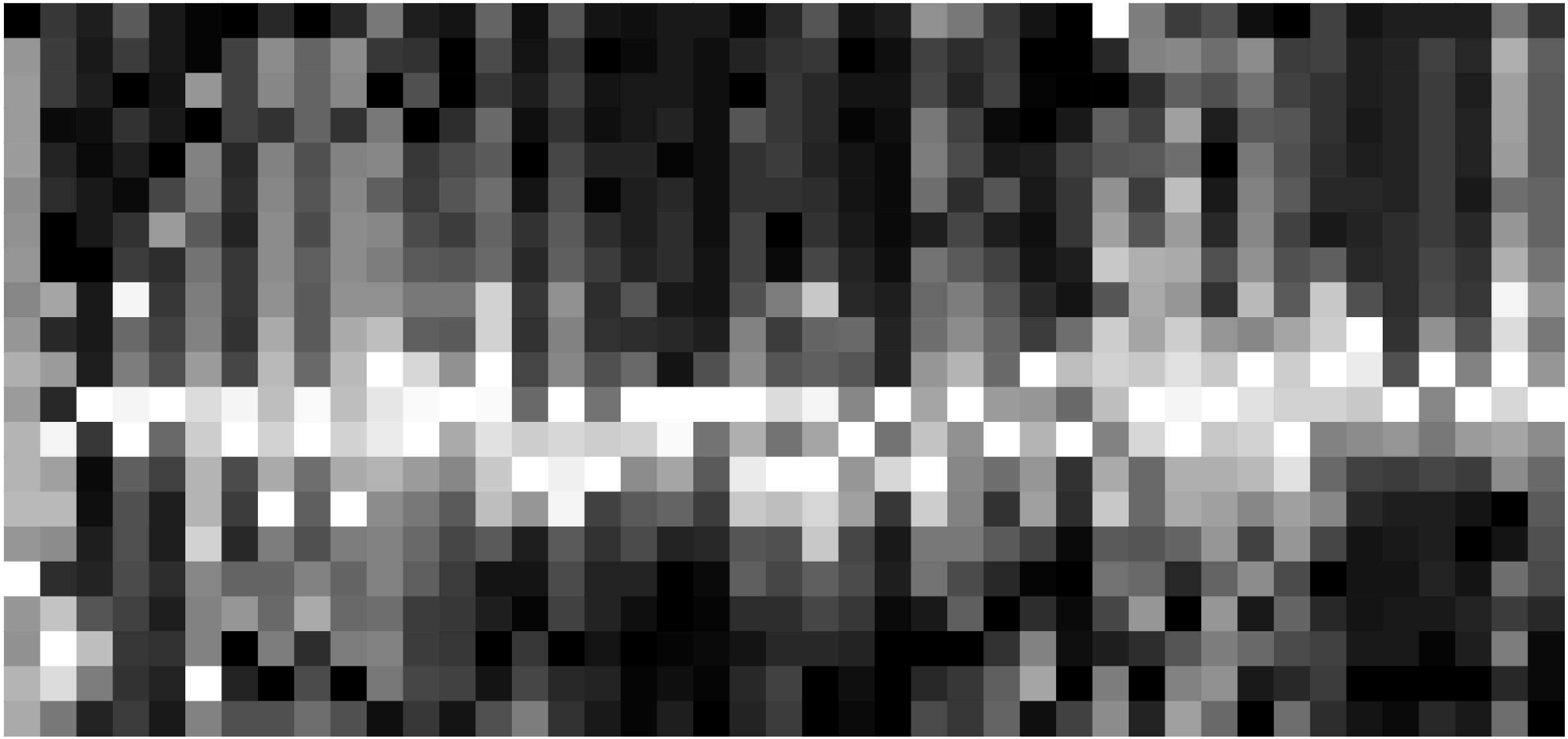
| | longitude | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 75 | 80 | 85 | 90 | 95 | 100 | 105 | 110 | 115 | 120 | 125 | 130 | 135 | 140 | 145 | 150 | 155 | 160 | 165 | 170 | 175 | 180 | 185 | 190 | 195 | 200 | 205 | 210 | 215 | 220 | 225 | 230 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Latitude | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -50 | | 2048 | 2054 | 2053 | 2054 | 2053 | 2052 | 2053 | 2051 | 2052 | 2051 | 2052 | 2049 | 2050 | 2053 | 2053 | 2053 | 2051 | 2052 | 2053 | 2054 | 2050 | 2051 | 2055 | 2053 | 2055 | 2053 | 2053 | 2052 | 2053 | 2053 | 2055 | 2047 | 2045 | 2048 | 2047 | 2051 | 2054 | 2053 | 2053 | 2052 | 2052 | 2052 | 2051 |
| -45 | | 2053 | 2054 | 2053 | 2053 | 2053 | 2052 | 2054 | 2053 | 2054 | 2053 | 2049 | 2050 | 2049 | 2052 | 2053 | 2051 | 2049 | 2052 | 2053 | 2053 | 2052 | 2052 | 2053 | 2051 | 2054 | 2052 | 2051 | 2053 | 2052 | 2052 | 2039 | 2047 | 2049 | 2050 | 2053 | 2053 | 2054 | 2053 | 2053 | 2054 | 2053 | 2054 | 2054 |
| -40 | | 2053 | 2054 | 2053 | 2052 | 2053 | 2054 | 2054 | 2053 | 2054 | 2053 | 2047 | 2051 | 2049 | 2051 | 2053 | 2052 | 2051 | 2053 | 2053 | 2053 | 2049 | 2053 | 2052 | 2053 | 2054 | 2052 | 2050 | 2053 | 2053 | 2053 | 2036 | 2040 | 2047 | 2048 | 2052 | 2053 | 2053 | 2053 | 2053 | 2053 | 2052 | 2054 | 2053 |
| -35 | | 2053 | 2053 | 2053 | 2053 | 2053 | 2047 | 2054 | 2052 | 2054 | 2052 | 2052 | 2047 | 2051 | 2053 | 2053 | 2051 | 2051 | 2053 | 2054 | 2053 | 2055 | 2053 | 2052 | 2052 | 2054 | 2053 | 2051 | 2051 | 2052 | 2055 | 2043 | 2042 | 2050 | 2045 | 2050 | 2054 | 2053 | 2053 | 2053 | 2053 | 2053 | 2054 | 2053 |
| -30 | | 2053 | 2053 | 2053 | 2053 | 2053 | 2053 | 2053 | 2053 | 2053 | 2053 | 2052 | 2050 | 2052 | 2052 | 2052 | 2052 | 2053 | 2054 | 2052 | 2053 | 2053 | 2053 | 2052 | 2053 | 2053 | 2053 | 2051 | 2052 | 2053 | 2058 | 2042 | 2044 | 2048 | 2042 | 2052 | 2054 | 2053 | 2053 | 2053 | 2053 | 2053 | 2054 | 2053 |
| -25 | | 2053 | 2053 | 2053 | 2052 | 2054 | 2053 | 2053 | 2053 | 2053 | 2053 | 2051 | 2050 | 2053 | 2053 | 2053 | 2053 | 2050 | 2053 | 2055 | 2053 | 2052 | 2052 | 2053 | 2053 | 2054 | 2053 | 2050 | 2053 | 2053 | 2057 | 2047 | 2042 | 2052 | 2044 | 2053 | 2054 | 2053 | 2054 | 2053 | 2053 | 2052 | 2051 | 2054 |
| -20 | | 2053 | 2053 | 2053 | 2053 | 2055 | 2053 | 2053 | 2053 | 2053 | 2053 | 2052 | 2051 | 2052 | 2053 | 2054 | 2053 | 2054 | 2053 | 2055 | 2053 | 2053 | 2048 | 2053 | 2053 | 2053 | 2051 | 2051 | 2052 | 2053 | 2057 | 2048 | 2044 | 2050 | 2045 | 2053 | 2053 | 2052 | 2053 | 2054 | 2053 | 2054 | 2053 | 2054 |
| -15 | | 2053 | 2053 | 2053 | 2053 | 2053 | 2053 | 2053 | 2053 | 2053 | 2053 | 2052 | 2052 | 2053 | 2053 | 2054 | 2053 | 2055 | 2054 | 2055 | 2054 | 2054 | 2049 | 2054 | 2054 | 2054 | 2053 | 2052 | 2053 | 2053 | 2055 | 2051 | 2051 | 2051 | 2048 | 2053 | 2054 | 2055 | 2054 | 2054 | 2054 | 2055 | 2054 | 2055 |
| -10 | | 2053 | 2054 | 2053 | 2056 | 2054 | 2053 | 2053 | 2053 | 2053 | 2053 | 2053 | 2054 | 2055 | 2056 | 2054 | 2056 | 2054 | 2057 | 2053 | 2054 | 2055 | 2058 | 2061 | 2055 | 2055 | 2052 | 2053 | 2053 | 2054 | 2054 | 2043 | 2050 | 2050 | 2046 | 2056 | 2054 | 2058 | 2056 | 2054 | 2054 | 2055 | 2058 | 2057 |
| -5 | | 2053 | 2053 | 2053 | 2054 | 2054 | 2053 | 2053 | 2054 | 2053 | 2054 | 2054 | 2052 | 2053 | 2056 | 2054 | 2056 | 2054 | 2054 | 2055 | 2054 | 2058 | 2053 | 2055 | 2061 | 2056 | 2053 | 2054 | 2054 | 2054 | 2061 | 2052 | 2050 | 2053 | 2052 | 2053 | 2057 | 2058 | 2066 | 2054 | 2057 | 2058 | 2057 | 2055 |
| 0 | | 2054 | 2054 | 2054 | 2054 | 2054 | 2054 | 2054 | 2054 | 2054 | 2054 | 2057 | 2058 | 2056 | 2058 | 2055 | 2056 | 2057 | 2058 | 2053 | 2059 | 2058 | 2055 | 2055 | 2059 | 2055 | 2054 | 2055 | 2054 | 2061 | 2068 | 2052 | 2053 | 2054 | 2056 | 2059 | 2058 | 2060 | 2065 | 2056 | 2061 | 2064 | 2058 | 2057 |
| 5 | | 2053 | 2053 | 2059 | 2056 | 2056 | 2054 | 2056 | 2054 | 2056 | 2054 | 2056 | 2060 | 2061 | 2058 | 2057 | 2062 | 2061 | 2069 | 2068 | 2074 | 2066 | 2065 | 2063 | 2064 | 2072 | 2054 | 2057 | 2055 | 2058 | 2061 | 2051 | 2057 | 2055 | 2060 | 2058 | 2058 | 2059 | 2063 | 2069 | 2056 | 2078 | 2056 | 2063 |
| 10 | | 2054 | 2055 | 2054 | 2056 | 2054 | 2054 | 2056 | 2055 | 2056 | 2055 | 2056 | 2060 | 2057 | 2057 | 2061 | 2060 | 2069 | 2066 | 2067 | 2062 | 2061 | 2057 | 2059 | 2075 | 2062 | 2054 | 2054 | 2058 | 2059 | 2073 | 2046 | 2054 | 2056 | 2056 | 2057 | 2060 | 2056 | 2060 | 2061 | 2056 | 2067 | 2054 | 2056 |
| 15 | | 2054 | 2054 | 2053 | 2054 | 2054 | 2054 | 2054 | 2054 | 2054 | 2054 | 2054 | 2055 | 2055 | 2056 | 2063 | 2062 | 2074 | 2061 | 2062 | 2060 | 2065 | 2067 | 2063 | 2065 | 2069 | 2056 | 2054 | 2054 | 2058 | 2057 | 2049 | 2045 | 2051 | 2054 | 2055 | 2059 | 2055 | 2055 | 2055 | 2054 | 2056 | 2053 | 2054 |
| 20 | | 2054 | 2055 | 2053 | 2053 | 2053 | 2054 | 2053 | 2056 | 2053 | 2056 | 2053 | 2053 | 2053 | 2056 | 2059 | 2062 | 2056 | 2057 | 2058 | 2057 | 2063 | 2063 | 2061 | 2066 | 2057 | 2055 | 2053 | 2052 | 2058 | 2056 | 2051 | 2045 | 2051 | 2053 | 2053 | 2057 | 2056 | 2054 | 2053 | 2052 | 2051 | 2046 | 2053 |
| 25 | | 2053 | 2054 | 2054 | 2053 | 2053 | 2054 | 2053 | 2053 | 2053 | 2053 | 2052 | 2053 | 2052 | 2052 | 2053 | 2053 | 2054 | 2056 | 2054 | 2055 | 2055 | 2055 | 2061 | 2058 | 2055 | 2053 | 2053 | 2053 | 2055 | 2053 | 2043 | 2044 | 2047 | 2053 | 2049 | 2056 | 2054 | 2053 | 2053 | 2052 | 2049 | 2047 | 2053 |
| 30 | | 2057 | 2053 | 2054 | 2053 | 2053 | 2053 | 2054 | 2053 | 2054 | 2053 | 2052 | 2052 | 2052 | 2050 | 2053 | 2052 | 2053 | 2054 | 2052 | 2053 | 2055 | 2054 | 2055 | 2058 | 2055 | 2053 | 2052 | 2052 | 2053 | 2052 | 2045 | 2045 | 2043 | 2049 | 2053 | 2054 | 2052 | 2053 | 2052 | 2052 | 2052 | 2052 | 2052 |
| 35 | | 2053 | 2055 | 2055 | 2053 | 2053 | 2053 | 2055 | 2053 | 2055 | 2053 | 2051 | 2050 | 2052 | 2051 | 2052 | 2052 | 2053 | 2052 | 2052 | 2052 | 2052 | 2052 | 2054 | 2056 | 2053 | 2051 | 2052 | 2051 | 2054 | 2054 | 2041 | 2049 | 2041 | 2052 | 2047 | 2055 | 2053 | 2052 | 2052 | 2052 | 2053 | 2049 | 2051 |
| 40 | | 2053 | 2055 | 2058 | 2053 | 2053 | 2053 | 2053 | 2053 | 2053 | 2053 | 2052 | 2051 | 2051 | 2049 | 2054 | 2048 | 2051 | 2051 | 2052 | 2053 | 2051 | 2052 | 2052 | 2054 | 2052 | 2050 | 2049 | 2052 | 2057 | 2054 | 2038 | 2041 | 2046 | 2051 | 2051 | 2054 | 2054 | 2053 | 2053 | 2051 | 2053 | 2052 | 2049 |
| 45 | | 2054 | 2055 | 2056 | 2053 | 2053 | 2054 | 2053 | 2050 | 2053 | 2050 | 2052 | 2051 | 2052 | 2050 | 2055 | 2051 | 2053 | 2052 | 2053 | 2052 | 2052 | 2053 | 2050 | 2053 | 2053 | 2051 | 2051 | 2054 | 2058 | 2052 | 2046 | 2037 | 2048 | 2052 | 2047 | 2053 | 2054 | 2051 | 2051 | 2051 | 2049 | 2048 | 2049 |
| 50 | | 2054 | 2054 | 2054 | 2053 | 2053 | 2053 | 2054 | 2052 | 2054 | 2052 | 2048 | 2050 | 2050 | 2052 | 2058 | 2051 | 2052 | 2052 | 2054 | 2052 | 2052 | 2053 | 2050 | 2052 | 2053 | 2052 | 2051 | 2054 | 2053 | 2058 | 2047 | 2040 | 2050 | 2050 | 2046 | 2055 | 2053 | 2052 | 2051 | 2053 | 2052 | 2051 | 2048 |

Average: 2053.565455 Count: 903 Sum: 1854369.606

# With conditional Formatting we can give a COLOR to EACH NUMBER



# Delete the Numbers

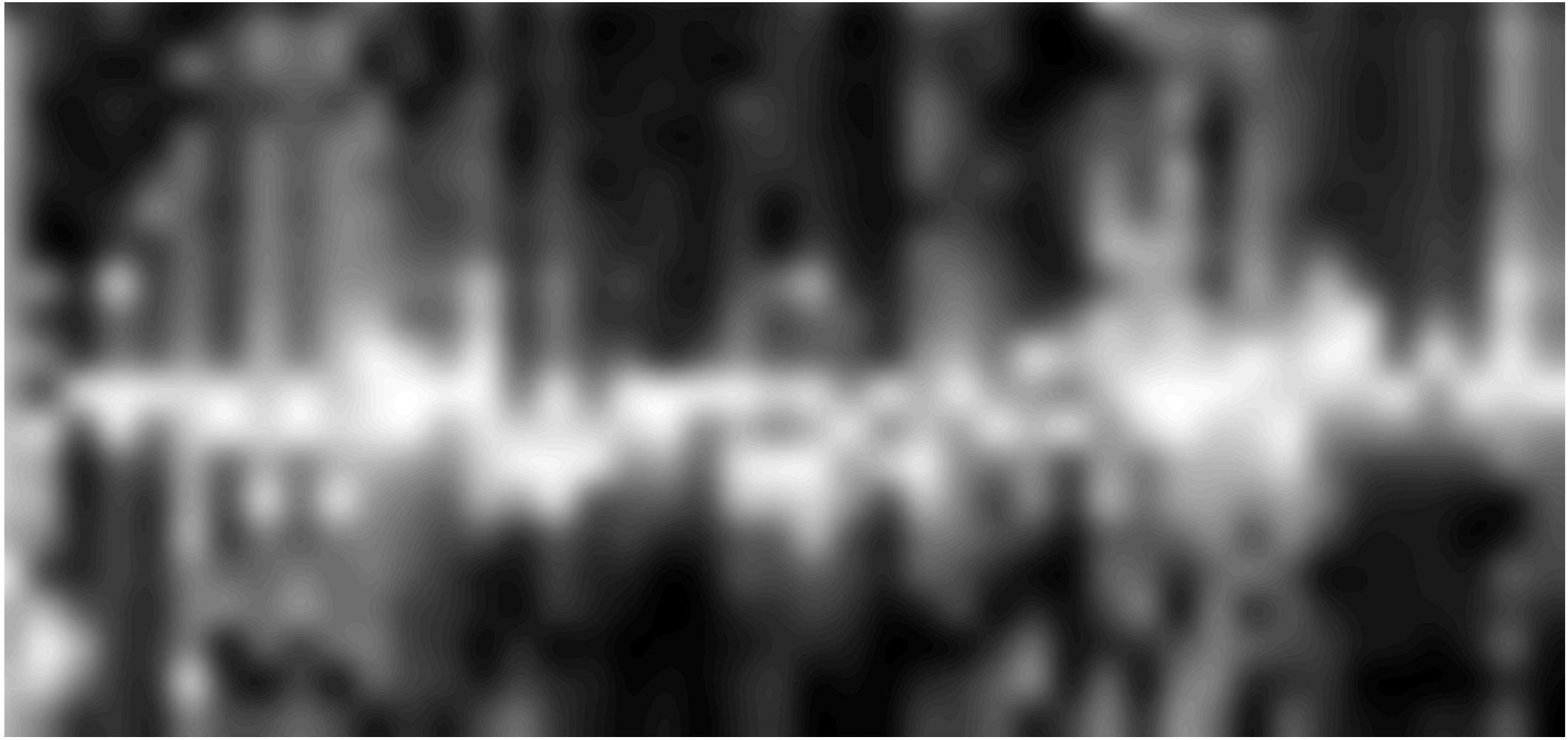


# Blur a little and you got representations of our Galaxy in Neutral Hydrogen

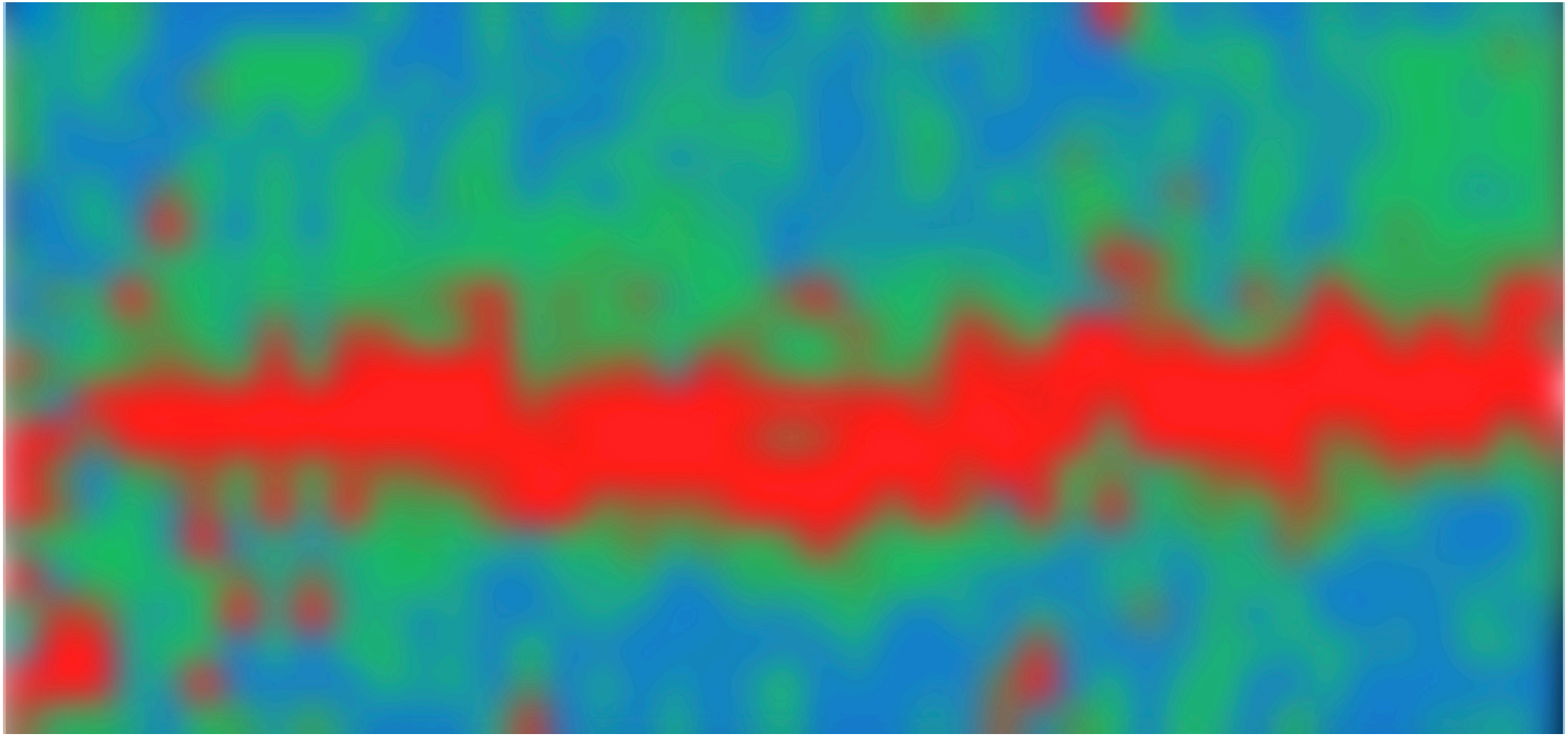
1) The Black White is the difference with highest and lowest number (quantity of Hydrogen)
2) The 3 Color one is more concentrated on the highest numbers (the red)
3) The Python Graph has another algorithm which generates a slightly different view but still clearly the Galaxy and its neighbourhood are visible

REMEMBER THIS IS ALL DONE WITH A RELATIVE SIMPLE RADIO DISH TELESCOPE WITH A DIAMETER OF 1.5 METER !

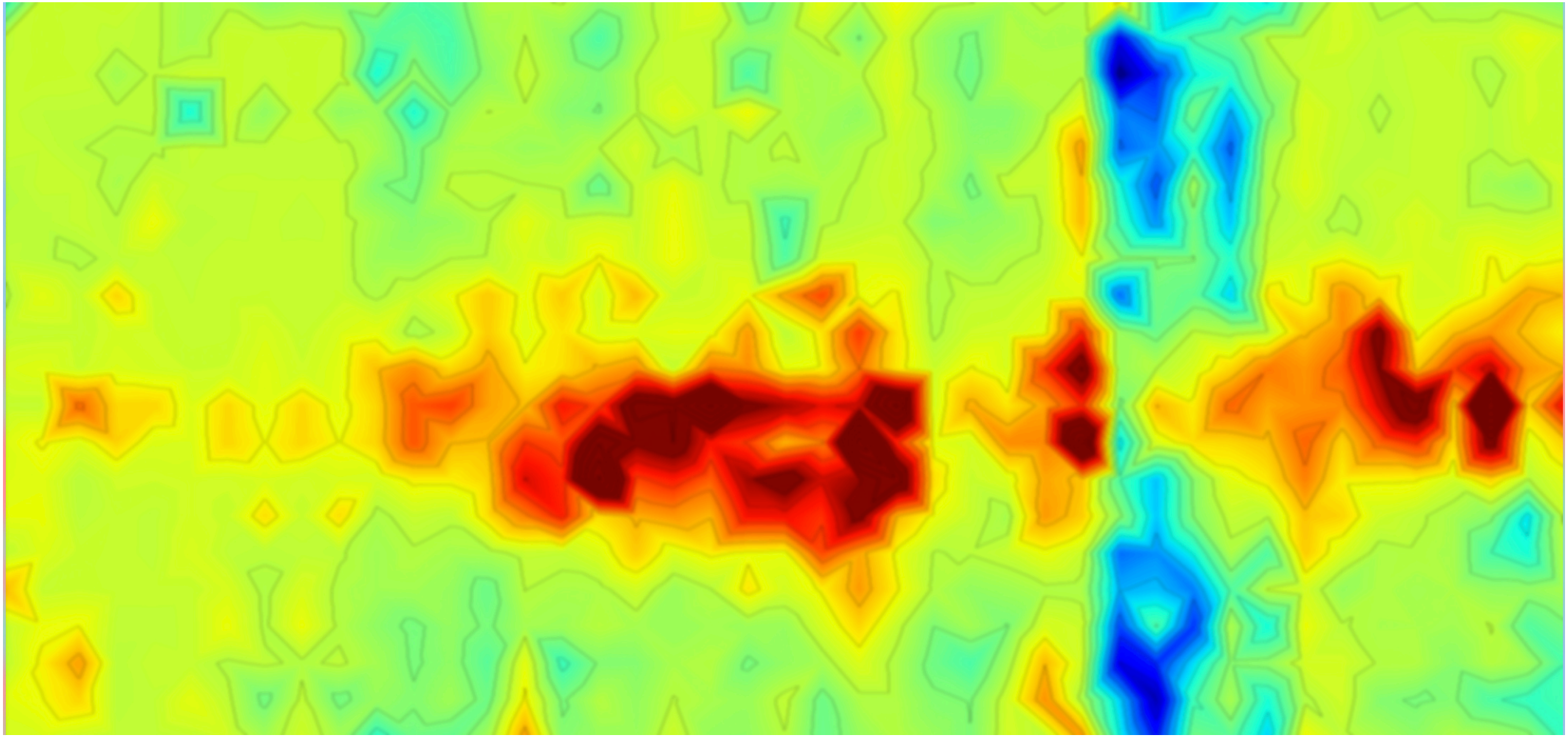
# Black White



# 3 Color



# Python Automated Script



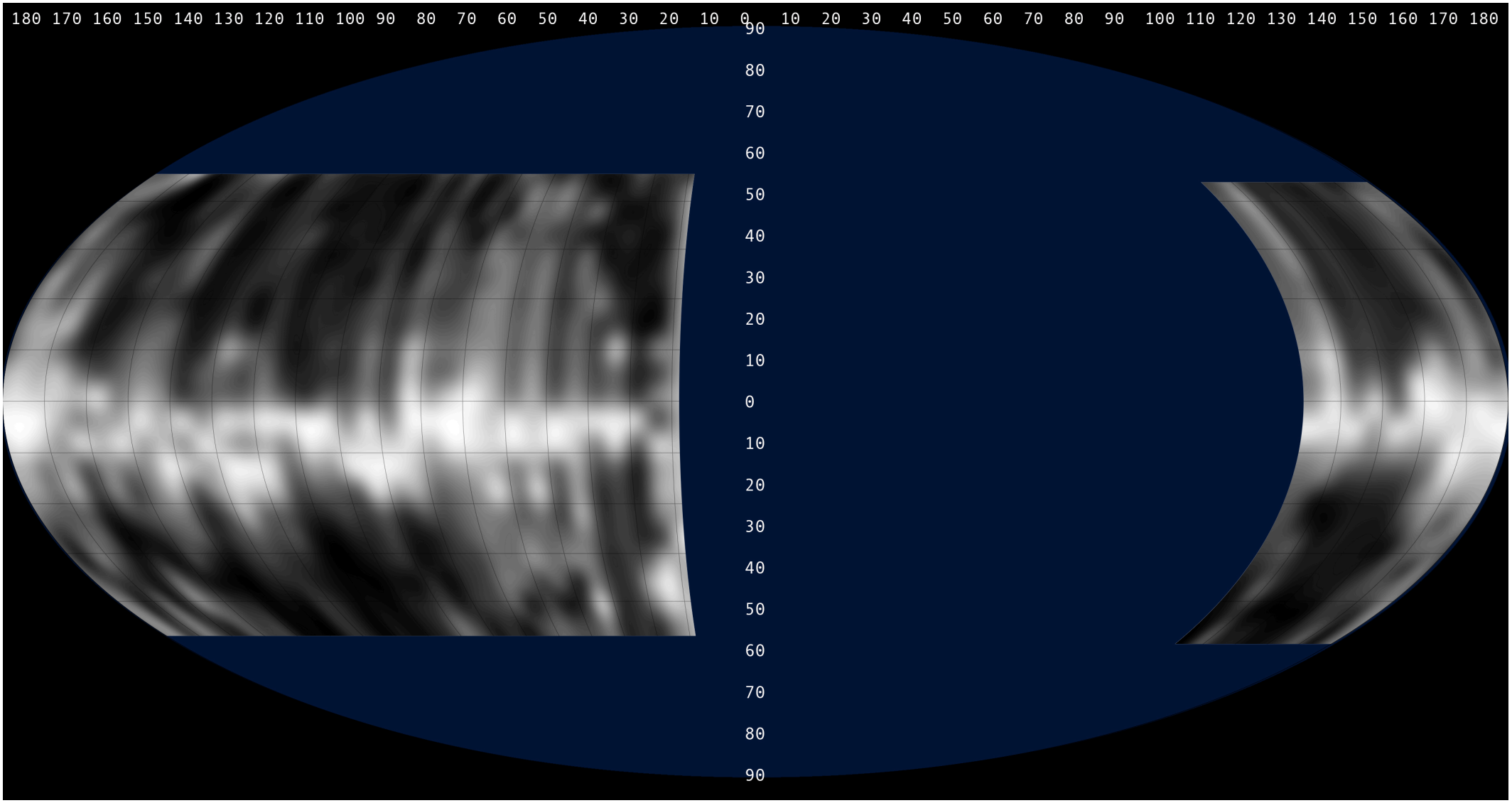
# Black And the Final State is generating a ‘MOLLWEIDE’ Representation

A Mollweide Cartographic is the whole Universe in 1 ‘Globe’ picture.
From -180 to 180 degrees horizontal en -90 to 90 degrees vertical.

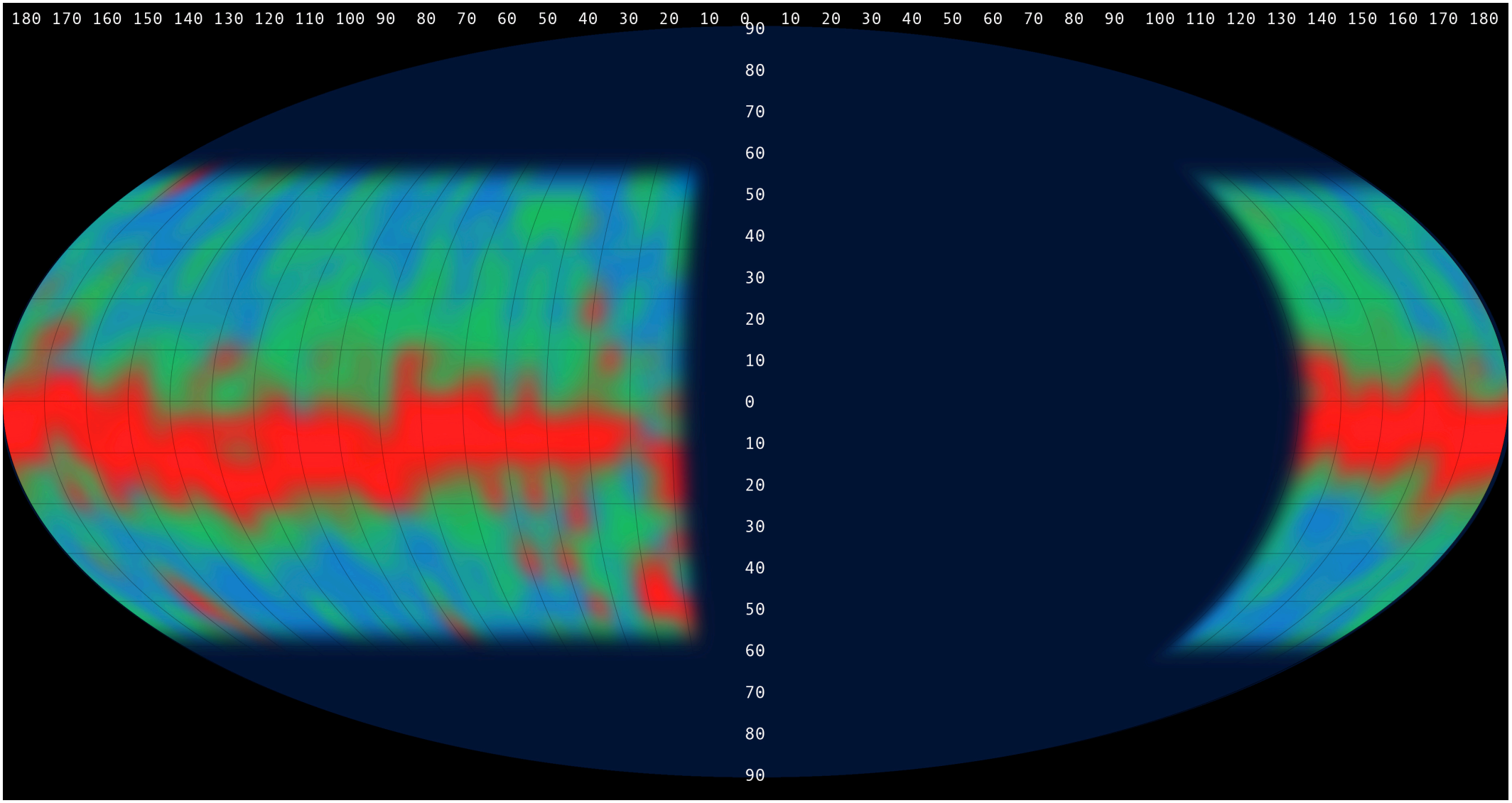
Because there is a significant part of the Universe which is NOT visible in The Netherlands, you see some ‘gaps’ in the final picture.

Also I didn’t record the latitudes -50 to 90 and 50-90 because there is hardly H I (Hydrogen)

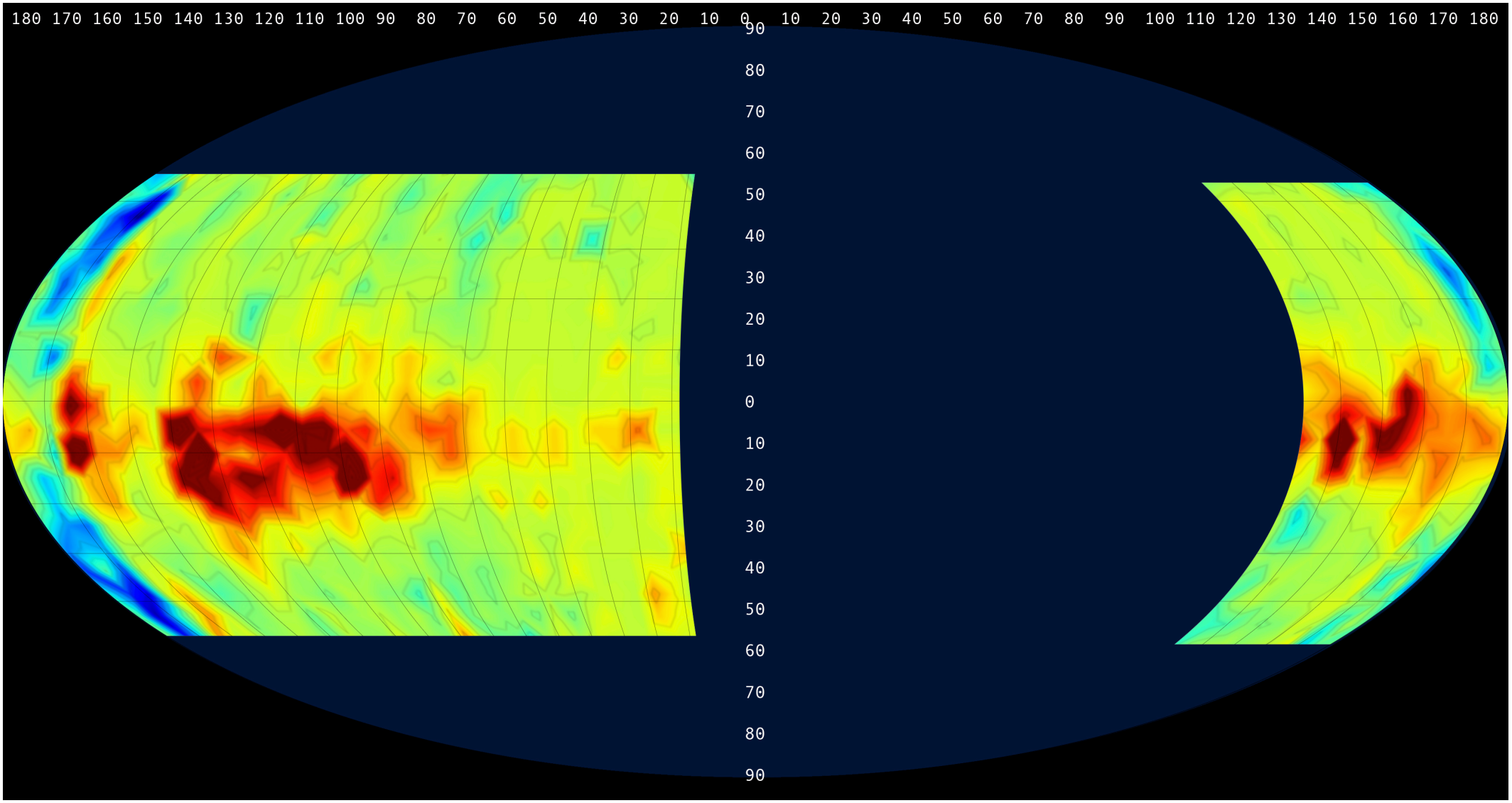
# Mollweide Black and White



# Mollweide 3 Color



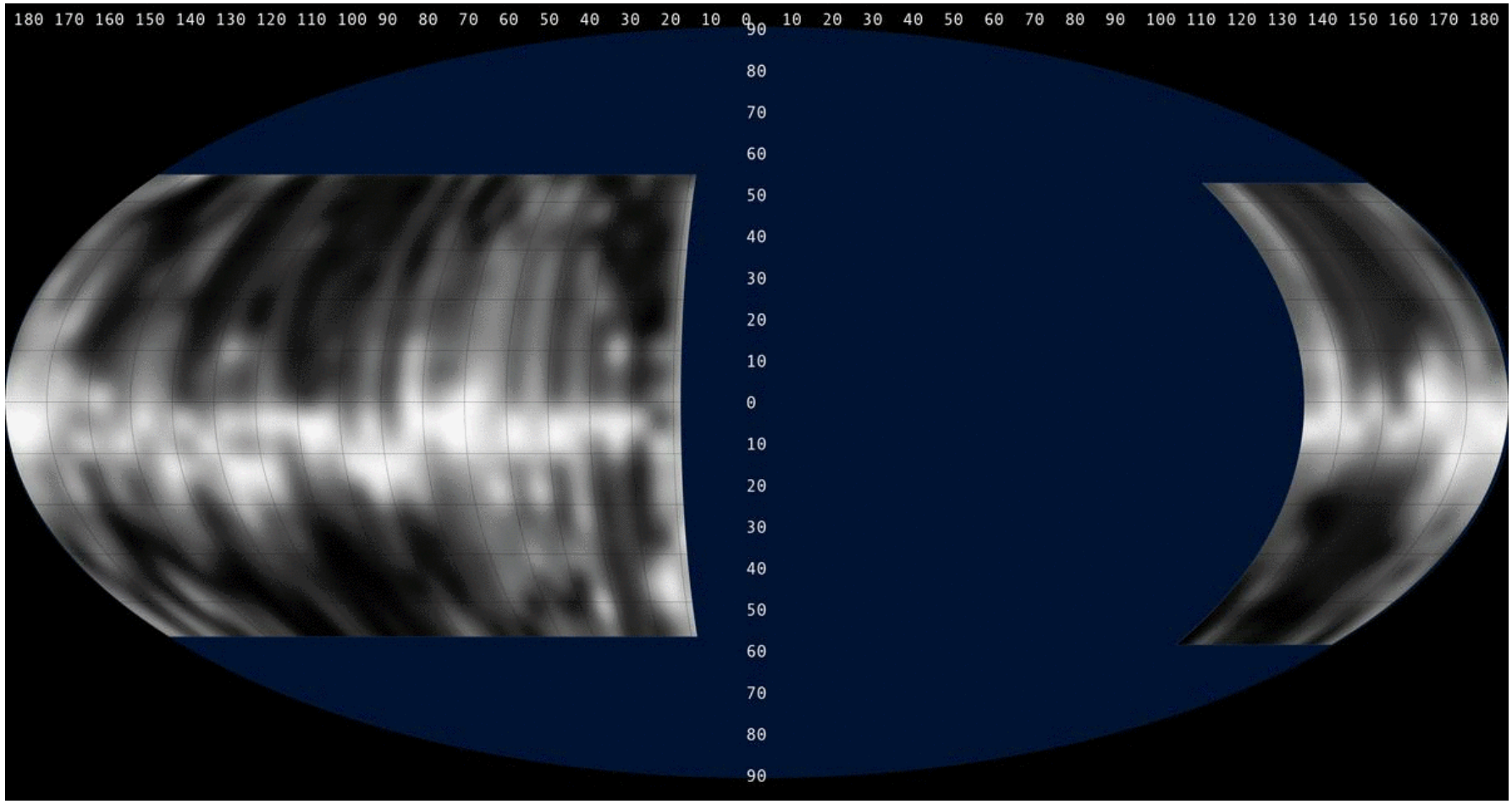
# Mollweide Python Algorithm



Job Geheniau
The Netherlands july 2020
jobgeheniau@gmail.com

Thanks to my parents, Apostolos Spanakis-Misirlis, Simon Bijlsma, Eduard Mol,Andries Boone, Michiel Klaassen, Raydel Abreu and many others

# press play


180 170 160 150 140 130 120 110 100 90 80 70 60 50 40 30 20 10 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180
90
80
70
60
50
40
30
20
10
0
10
20
30
40
50
60
70
80
90