```
# Author: Ruchita Nagare

# Loading in and viewing the data
In [1]: rdd = sc.textFile("pulsar.dat")

In [2]: rdd.take(5)
Out[2]:
['93.8765312388173 108.20078392209886 1765.5321859344265 6310.123216058047',
 '67.97743441792922 93.25357058016435 2369.906192066804 5113.778913051908',
 '79.74454835155092 100.85608068407274 1324.0936447318852 5299.014486317085',
 '85.90160700924295 91.12335932355143 1385.7936068285733 4572.388096226848',
 '94.75943025493382 76.72257316410906 2000.2549364326635 2841.2044819794382']

#importing math module for ceil function used below
In [3]: import math

# Splitting the rdd on space so that each observation becomes a separate list like entity.
# Converting the values from string to float
# Taking care of the standard deviation by
# rounding ascension using ceil, rounding declination to 0 decimal places
# rounding off timestamp to two decimal places. This will keeps its sensitive in order to identify periods of blip.
# rounding off the frequency to its tenth place.
# These rounding include the errors while generating a practically overlapping data.
In [4]: rdd2 = (rdd.map(lambda x: x.split())
   ...:              .map(lambda x: [float(x[0]),float(x[1]),float(x[2]),float(x[3])])
   ...:              .map(lambda x: [math.ceil(x[0]),round(x[1],0),round(x[2],2),round(x[3],-1)])
   ...:              )

# Viewwing the new rdd
In [5]: rdd2.take(5)
Out[5]:
[[94, 108.0, 1765.53, 6310.0],
 [68, 93.0, 2369.91, 5110.0],
 [80, 101.0, 1324.09, 5300.0],
 [86, 91.0, 1385.79, 4570.0],
 [95, 77.0, 2000.25, 2840.0]]

# We first create a key value pair, where key is (ascension, declination, frequency) and
# values are instantiation count 1.
# We then reduce by key. This gives us the count for every unique astronomical body in the data.
# We then flip the (key,value) pair to have the count as the key.
# We then sort by the key in descending order to identify the most occuring astronomical body.
In [6]: counts_rdd = (rdd2.map(lambda x: ((x[0],x[1],x[3]),1))
   ...:                         .reduceByKey(lambda x,y: x+y)
   ...:                         .map(lambda x: (x[1],x[0]))
   ...:                         .sortByKey(False)
   ...:                         )

# Viewing the top 5 most occuring objects
In [7]: counts_rdd.take(5)
Out[7]:
[(18, (73, 100.0, 6970.0)),
 (12, (109, 101.0, 1710.0)),
 (6, (98, 96.0, 7690.0)),
 (6, (62, 110.0, 4760.0)),
 (6, (117, 119.0, 5840.0))]

# Above we can see that the object with ascension roughly around 73 degrees,
# decension roughly around 100 degrees and frequency roughly around 6970 Mhz
# emits the most repeating blips in the sky.
# So we now filter out this object from our original dataset, and sort it by timestamp.
# This is our winner astronomical object.
```

```
In [8]: winner_rdd = (rdd2.filter(lambda x: (x[0]==73) and (x[1]==100) and (x[3]==6970))
    ...:                     .sortBy(lambda x:x[2])
    ...:                     )

# Viewing the winner and identifying its blip pattern.
In [9]: winner_rdd.collect()
Out[9]:
[[73, 100.0, 2442.0, 6970.0],
 [73, 100.0, 2446.1, 6970.0],
 [73, 100.0, 2450.19, 6970.0],
 [73, 100.0, 2454.3, 6970.0],
 [73, 100.0, 2458.4, 6970.0],
 [73, 100.0, 2462.5, 6970.0],
 [73, 100.0, 2466.6, 6970.0],
 [73, 100.0, 2470.7, 6970.0],
 [73, 100.0, 2474.8, 6970.0],
 [73, 100.0, 2478.9, 6970.0],
 [73, 100.0, 2483.0, 6970.0],
 [73, 100.0, 2487.1, 6970.0],
 [73, 100.0, 2491.2, 6970.0],
 [73, 100.0, 2495.3, 6970.0],
 [73, 100.0, 2499.4, 6970.0],
 [73, 100.0, 2503.5, 6970.0],
 [73, 100.0, 2507.6, 6970.0],
 [73, 100.0, 2511.7, 6970.0]]

# It can be observed that this most regular RF source and it repeats roughly about every 4 secs!!
# Concluding statement: In my opinion the most regular temporarily repeating RF is located around
# ascension 73 degrees, declination 100 degrees and has a frequency of around 6970 Mhz.
# This target is found in the same location, on the same frequency (all within errors)
# emitting the most blips regularly spaced in time after about every 4 seconds during that active period.
```