Summarization, Mapping, Hotspot Discovery and Change Analysis of High-Intensity Solar Flare Events

Helios Project

The goal of the project is to design and implement a system called Helios, which is capable of summary generation, mapping, hotspot discovery, and change analysis of high-intensity solar flares events.

Brayan A. Gutierrez (1865588)

University of Houston, bagutie3@cougarnet.uh.edu

Jericka A. Ledezma (1968730)

University of Houston, jaledez3@cougarnet.uh.edu

Katie D. To (2026435)

University of Houston, kdto2@cougarnet.uh.edu

Ulises U. Ramirez (1419086)

University of Houston, uuramirez@uh.edu.com

1 Task 1 - Subtask 1

In order to develop Method 1, we first batched the dataset into 11 batches based on months. From each batch, a random set of coordinates were chosen, then a value of 50 was subtracted and added to the x,y coordinate values in order to create a circle around the origin coordinates. Deciding how to choose coordinates was the first challenge as every batch is going to be different. Although the batches are going to be very similar, the specific coordinates on each batch will be different. In order to group the dataset without bias, we decided to add all coordinates and their given total.counts into a data frame and randomly choose a coordinate from the data frame.

The radius was manually chosen based on testing in the data set. This was determined to be a balanced radius based on the maximum size of the coordinates that were based on the data provided. An increase in the radius is possible, however to have a balance in the radius and intensity we limited our radius to 50 units. The intensity value that was calculated with the total.counts column was growing to very large values and any bigger radius would have created values that were too big to use efficiently. However any smaller value for the radius would have created a large amount of area to be checked. This is not only inefficient but too many areas to check would defeat the purpose of the areas we created. A small radius would lead to a large amount of circles, leading us closer to coordinates we started with.

Each coordinate inside the smaller area of the circle was read by the function, an area based on value for radius was created and any coordinate inside the area was then removed out of the entire batch to only leave us with a data frame containing the remaining coordinates in the circle. From there, we added the total_counts of every point in that radius, removed every point except the origin, then returned the row with the overall total_counts paired with the origin coordinates and the remaining values left in the batch that were not a part of the counted dataframe. This was repeatedly called until every point was removed from the batch. With this, points were not revisited or counted twice. Only once the final coordinate is removed from the data frame, the algorithm is going to stop assuring that all coordinates are checked and accounted for. The final data frame from this method will include the random coordinates that were selected and the new total.count value that was added from the surrounding coordinates to provide the final value.

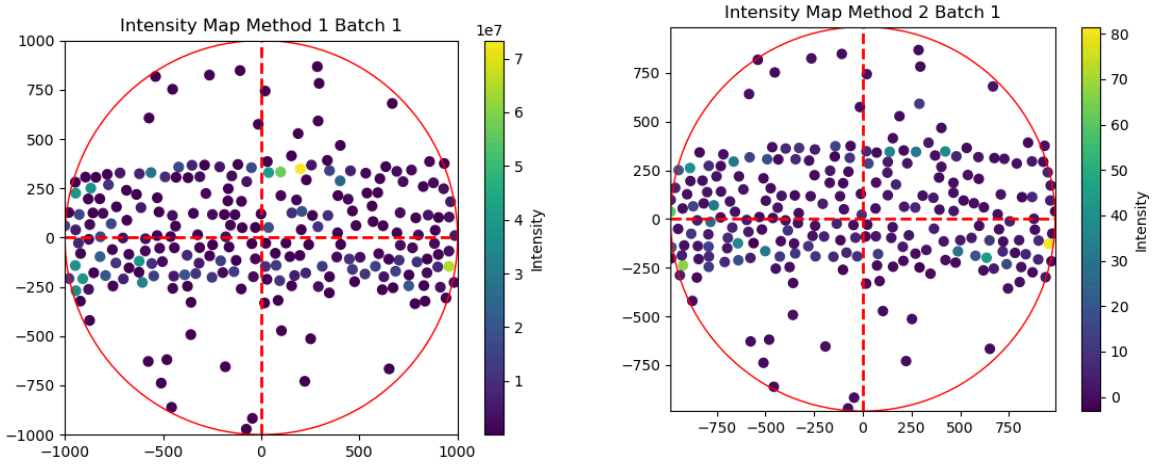
Method 1 creates intensity maps for the batches using total.counts to help understand where the solar flares are located. Ulises Ramirez created the method of taking the batches and creating an algorithm that will recursively check the data in the batches. The algorithm was successfully created but a single batch would take close to an hour to analyze. Inefficiency became an issue as it would take too long to analyze a single batch and analyzing the whole dataset for other parts of analysis would become very problematic. After several edits by Brayan Guiterriez to no success, we finally used Chat GPT to make method 1 more efficient. Chat GPT split the existing code from 1 method to 3 methods and created a double recursion to increase the processing time. The purpose behind using total.counts is to understand how many times a solar flare occurred within a range of x and y coordinates.

2 Task 1 - Subtask 2

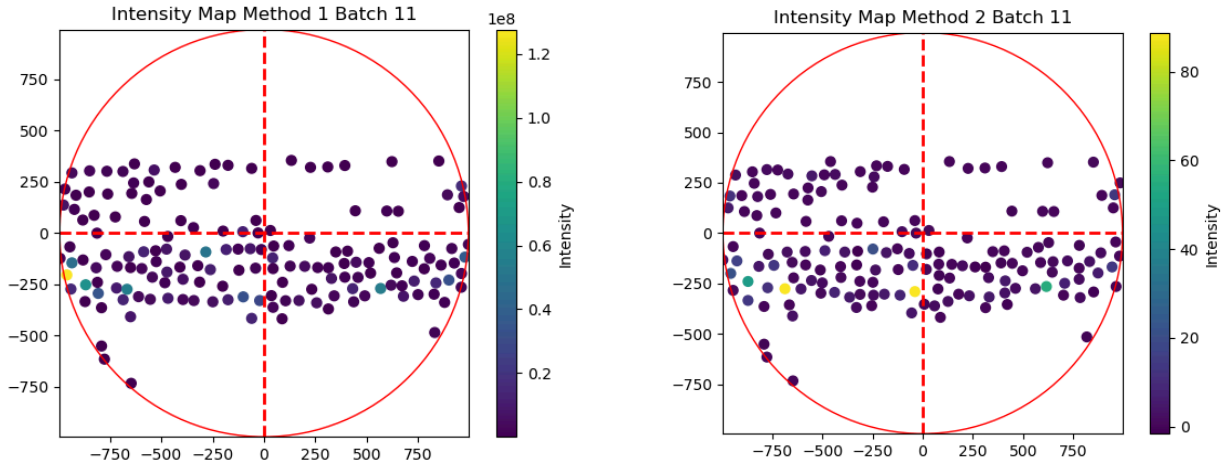
When developing Method 2, we took a similar approach to Method 1. The dataset was batched into 11 batches again, each dependent on their months, and a random coordinate was chosen from every batch to get our origin coordinates. Then a value of 50 units was subtracted and added to coordinate creating a circle around the origin coordinates. Each coordinate in the circle was read by the function, then filtered out of the entire batch to only leave us with a data frame containing the coordinates in the circle. From there, we took the initial and final energy in every coordinate point, calculated the energy midpoint, then added every single midpoint together to get the overall energy within the circle. We also added all of the durations for each coordinate point within the circle. This was repeated until every point was removed from the batch. With this, points were not revisited or counted twice. Although the method was very similar to our first method, the data

that we used for this analysis was different. For this method we had to use duration.s and energy.kev, now 2 attributes in this analysis. This requires us to think differently about intensity, to make sure that one attribute would skew the data drastically, we normalized the data and then added those values into our new combined value; intensity. Other than the changes made to the attributes used to form our new intensity value, method 2 followed the same steps and logic from our method 1.

3 Task 1 - Subtask 3



4 Task 1 - Subtask 4



5 Task 1 - Subtask 5

Looking at the result in our maps we can see that both methods display almost identical areas. There are naturally, different areas being shown but overall very similar. Most of the high intensity areas seem to be

found near the center of our diagrams. High intensity areas are typically found in a span of about 1000 units, from 500 to -500 along the y-axis and along the whole 2000 units along the x-axis.

There is spatial variation from batch to batch, something that was expected but we can now see that there is small spatial variation depending on the method used, although small it is present. Even when using different attributes the diagrams look very much the same with a focus of solar flares across the x-axis in the -250 to 250 y-axis range. There are more solar flares in this area than any other area on the sun.

When looking at total intensity, the areas of “high” intensity do have more of a significant change than the changes of spatial variation. More change is seen in the batches of both methods. It should be noted that although the specific coordinates of both “low” and “high” intensity do change, the areas with “high” intensity are about the same. There is not a massive change in location, in the first and eleven batches we can see that the areas of high intensity are all found around the same 250 unit areas regardless of method. These results could potentially mean that there is a correlation between total_count and both duration.s and energy.kev. The theory that if a specific area has high total_count it will have high duration.s and energy.key has merit

More testing would be needed to draw this conclusion but there is potential for this result as the similarities between method 1 and method 2 diagram is significant. Not just in spatial variation but intensity variation as well.

6 Task 2 - Subtask A

We went through many methods in order to find a proper way to detect hotspots. In the end, we settled on using the NumPy library’s Histogram2d function. In order to do this, we first ran every single batch through our Method 1, which returned us 11 new batches that had their total.counts summed up in every batch to produce total.counts intensity. As these were returned as DataFrames, we used the Pandas’s library .to_numpy function in order to change the DataFrame into a NumPy array. From there, we found the maximum and minimum of the x and y coordinates in the given batched Method 1 array, and these values were defined as our range values. Our grid_size value was predetermined to be a value of 25 to provide consistent binning across the batches.

The x-values of the batch, the y-values of the batch, the grid_size, and the range_values were passed into histogram2d to return a bi-dimensional histogram of samples x and y as an array. This array is a grid of bins where each bin is representative of a region in the 2D space, and the value in each bin represents the number of data points that fall into that bin. This method was found with the help of Raunak, who provided code to push us in the right direction.

7 Task 2 - Subtask B

We opted to automate the thresholding process rather than find it manually. To do this, we opted to find the hotspots for the entire dataset, not just batches, in order to find the overall counts we had in every bin. First, we ran the entire dataset through Method one. After, we used our hotspot detection method we found in Task 2 - Subtask A, then took the array that was returned and transformed it into a list that could be easily searched. We did this by implementing a for loop that would loop through the bins horizontally, then an inner for loop that would loop through the bins vertically, therefore going in every win and adding their count to a list. At the end, we removed every instance where there was no count, then sorted it, leaving us with a list of values that contained all instances of a bin’s count.

For threshold d1, we opted that it should be the value that sits at the 99th percentile, while the threshold d2 should be the 85th percentile. To achieve this, we took the entire length of the list of counts, multiplied it by

.99, then rounded it up to get what element number those values would be. Then we set our d1 to that slice. This ensures that the d1 threshold is going to return values that should be above 99% of all values, which will produce our very hot spots.

The same methodology is applied for the second threshold d2, except we multiply by .85 to produce a threshold that will return values that are above 85% of all other values. When this threshold is applied, it will count all values above 85% of all others, but below 99% as well.

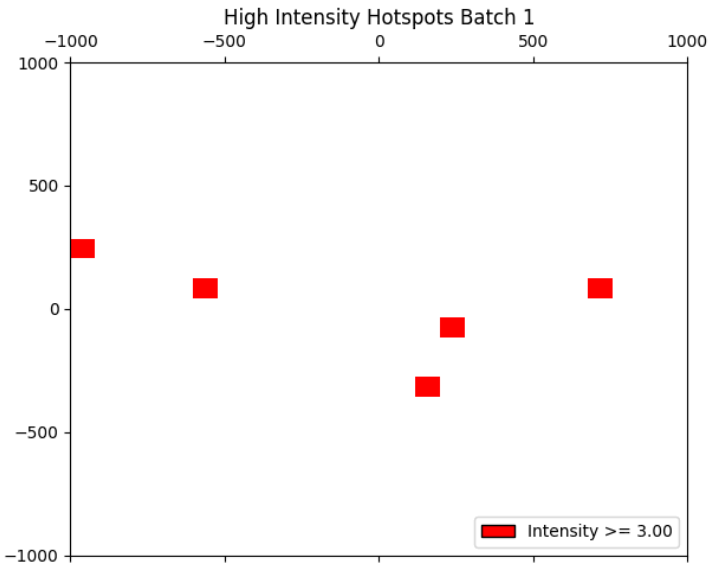
8 TASK 2 - SUBTASK C

The array that is returned from NumPy's histogram2d can be turned into a 2D histogram if put through the Matplotlib.pyplot's matshow() function. We pass the resulting histogram array from histogram2d as the data displayed on the 2D histogram and the prior range_values to outline the vertical and horizontal limits of the 2D histogram. From here, we were provided a 2d histogram with all values, no thresholds applied, with a colorbar displaying the counts rather than a key.

To change this, we first decided to tackle the thresholding. Our threshold selection process was described in TASK 2 - SUBTASK B, but to implement it we used NumPy's where() function. With this, we essentially changed the count of bins with less than the threshold d1 to a count of 0, and those with a count above were changed to have a count of 1. On the other hand, for threshold d2, if they were between thresholds d1 and d2, it was changed to have a count of 1 with values not fitting those parameters changed to have a count of 0.

For threshold d1, we colored the plots red, and threshold d2 plots were colored orange. We used ListedColormap from the matplotlib.colors library in order to color in the bins either red or white on the 2D histogram, then we used Patch() from the matplotlib.patches library to generate the respective keys on the upper right hand corner of both plots. Lastly, a title was added. This method was turned into a function that took the histogram2d array data, the batch number, the grid_size, and the d1 and d2 thresholds in order to make the function callable without needing to repeat it.

9 Task 2 - Subtask D



*****IF GIF IS NOT ANIMATED PLEASE REFER TO "HIGH INTENSITY HOT SPOTS" GIF FILE*****

10 Task 2 - Subtask F

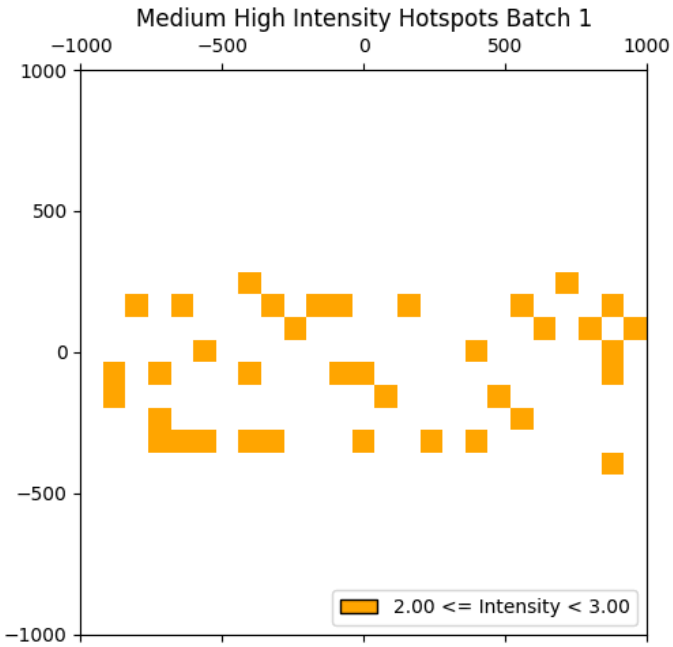
In this plotting, we have placed threshold d1 onto Flare Intensity Estimation Method 1. We measure the intensity of the all total.counts value at a certain point within its radius of (50,50), then we apply the hotspot analysis detection method we developed in Task 2 - Subtask A to see how many of those intensity points fall into a certain area we call bins. We currently have 625 bins in each plot. For the high intensity threshold, we've decided to place our threshold at a count of three intensity total.count intensity points per bin, as this value marks the 99th percentile of all values in a dataset. Therefore, this gif is showing our very hot spots that have an equal to or more than 3 counts of intensity in that bin across 2004-2005 that has been split into 11 batches.

We start off with 5 very hot spots in the first batch, but over time we see the amount slowly decrease until there are absolutely no hotspots in the 8th batch. The lack of any points on the plot on batch 8, halfway through 2005, shows us that there were no bins that were equal to or greater than 3, which means the maximum count for any bin in that batch was most likely 2. We suddenly see a spike of hotspots in batch 9, where there are suddenly 7 bins where there are equal to or more than 3 counts of intensities. The distribution of these usually fall in a band that falls horizontally across the screen, often laying anywhere across the x-axis but falling between -500 and 500 in the y-axis, this is where the equator of the sun would be.

A distinct pattern is somewhat hard to discern, but we can assume that there will be "peaks" in very hot spots that will inevitably start decreasing until another peak where multiple very hot spots occur. Once a new peak happens, it will continue to decrease, and the cycle will repeat. We can also determine that the average of very hot spots will lie around 3 or 4 bins, meaning that they are not that many of them compared to the number of overall bins. Lastly, we can see that not having any very hotspots is not necessarily a bad thing, as

it just means that a majority of the spots will fall into the medium-high threshold, but it appears it will most likely occur after a decline of very hot spots before it.

11 Task 2 - Subtask E



*****IF GIF IS NOT ANIMATED PLEASE REFER TO "MEDIUM HIGH INTENSITY HOT SPOTS" GIF FILE*****

12 Task 2 - Subtask G

In this plotting, we have placed threshold d2 onto Flare Intensity Estimation Method 1. For the medium-high intensity threshold, we've decided to place our threshold at a count of two intensity total.count intensity points per bin, as this value marks the 85th percentile of all values in a dataset. Therefore, this gif is showing our medium-high spots that are equal to or have more than 2 counts of intensity but less than counts of 3 in that bin across 2004-2005 that has been split into 11 batches.

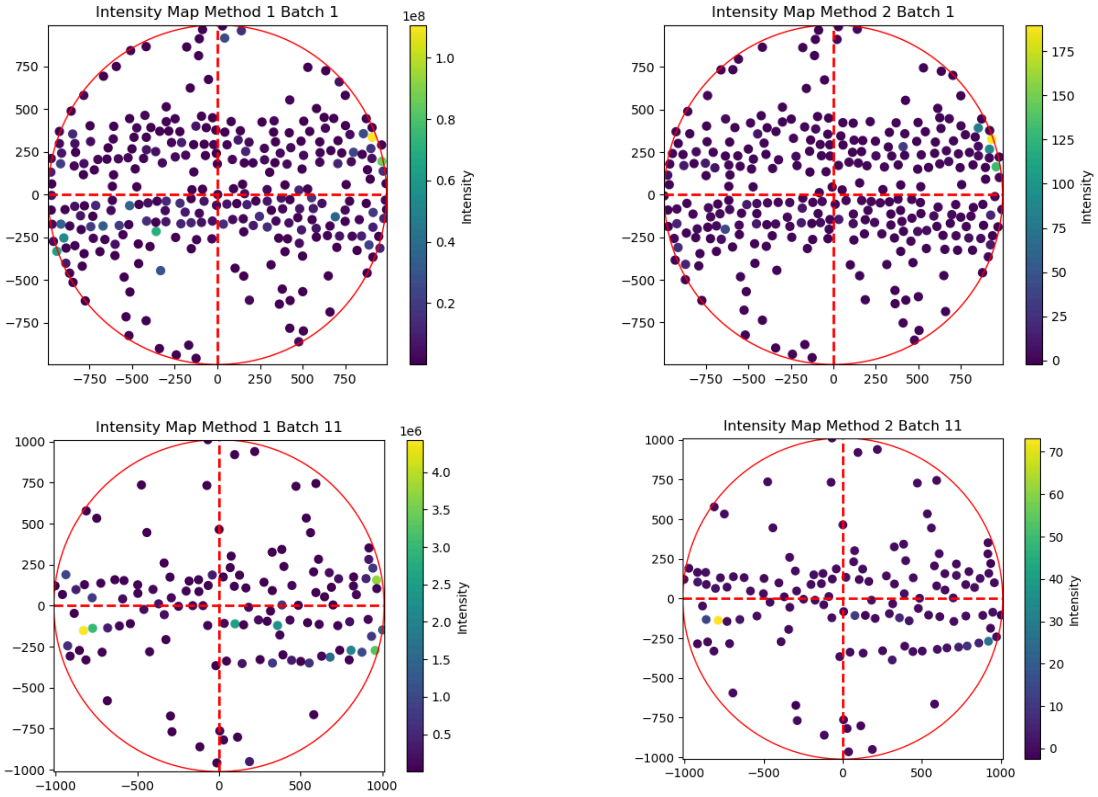
We start off with a fair amount of bins, with the count staying steady and increasing little by little until we hit batch 10, where it decreases drastically. There is no discernable pattern in the medium-high thresholding, besides from the fact the distribution of these usually fall in a band that falls horizontally across the screen, often laying anywhere across the x-axis but falling between -500 and 500 (the equator of the sun) in the y-axis like its high threshold counterparts. There is an average of around 28 bins with counts of 2 intensity spots across all 11 batches, with batch 10 and 11 having the least while batch 4 have the most. There are a few distinct "clusters" amongst the bins, where 3 or 4 of them are side by side and almost connected. There's at least three or four of these in every batch when they have a large amount of bins with a count of 2, so we can assume that these are a common occurrence that is directly proportional to the amount of overall bins in the batch. From this, we can determine that the sun has medium-high sunspots focused on its equator, that is spread about with large low-hot spots in between clusters of medium-high sunspots.

13 Task 2 - Subtask H

Comparing our time series visuals, it's evident that a few intense hotspots (d1) stand out in contrast to the more prevalent medium-high intensity hotspots (d2). This discrepancy appears to align with the Sun's surface dynamics. On average, we observe a consistent presence of "medium-high" hotspots across the entire solar surface, with fewer high-intensity spots scattered around.

In both time series, the identified hotspots, using both thresholds, consistently cluster around the equatorial region across all 11 batches. This pattern aligns with NASA's observations of solar flares and sunspots, reinforcing that these phenomena are indeed more concentrated around the Sun's equator. The dynamic nature of the Sun's surface is reflected in our findings, as the distribution of hotspots shifts with each batch. This aligns with expectations, given the Sun's ever-changing surface conditions over the two-year span covered by the data.

14 Task 3



First, let's take a look at the batches from method 1 for both sets. It's evident that in 2015-2016, the Sun displayed more intense spots across its surface. We can confidently assert that the Sun exhibited higher intensity during this period compared to 2004-2005. However, akin to the first set, the distribution of intensity remains relatively consistent, with each point maintaining a similar level. Interestingly, set one exhibited more

areas of high intensity in contrast to set 2, which seems to have experienced a decrease in intensity from the first set.

Shifting our focus to spatial variation between the two sets for method 1, it's notable that set 2 demonstrates a higher spatial variation compared to set 1.

Now, turning our attention to method 2 batches, we observe that in set 2, there is a significantly greater spatial variation compared to set 1. The points in set 2 are more widely distributed. Additionally, set 2 displays fewer areas of intensity compared to set 1. Similar to the first set, the areas of intensity remain relatively constant between the two batches.

While further testing is necessary to solidify this conclusion, there is potential for this result, as the similarities between the method 1 and method 2 diagrams are noteworthy. This similarity is not only evident in spatial variation but also in intensity fluctuation.

SET 1 COUNT - 17506	**DURATION.S**	**TOTAL.COUNTS**	**ENERGY.KEV.I**	**ENERGY.KEV.F**	**X.POS.ASEC**	**Y.POS.ASEC**	**ENERGY.KEV.MID**	**MONTH**	**YEAR**
MEAN	4.119735E-17	440712.95	7.710385	15.766194	-22.704673	-32.637667	-1.461187E-17	6.193934	2004.417343
STD	1.000029	3368383.13	6.360895	16.46	705.787329	209.14	1.000029	3.469627	0.493135
MIN	-1.100958	23	6.00	12.0	-1004	-974.0	-0.240607	1	2004
MAX	8.889124	318884832	300.0	800.0	996.0	986.0	47.295778	12	2005

SET 2 COUNT - 10779	**DURATION.S**	**TOTAL.COUNTS**	**ENERGY.KEV.I**	**ENERGY.KEV.F**	**X.POS.ASEC**	**Y.POS.ASEC**	**ENERGY.KEV.MID**	**MONTH**	**YEAR**
MEAN	1.054707E-17	278193.04	9.29	19.74	8.90	12.72	-1.450222E-17	5.20	2015.18
STD	1.000046	1285832.98	11.66	32.14	723.14	259.96	1.000046	3.23	0.39
MIN	-1.161273	8	6	12	-1004	-981	-0.25	1.00	2015
MAX	7.8621	68146688	300.0	800.0	1002.0	1012.0	24.52	12.00	2016

The number of entries for both sets are similar to one another, with Set 2 having around 7000 less entries than Set 1. The durations for Set 2 and Set 1 are similar in range, however the means tell us that there are more values that lie on the longer side of the scale than the shorter side of the duration scale, meaning in average the duration of solar flares in 2015 to 2016 were longer than the ones in 2004-2005. Additionally, the STD shows us that many of the values are close to the mean to be exact. The increased magnitude of the solar flares can be attributed to the 11 year solar cycle that has been observed over the last 400 years. Every 11 years the sun's magnetic field completely flips causing the north and south poles to switch places. The intensity maps created show how the solar flares initially concentrate in the upper quadrants 2004 to 2005, but after 10 years the polarities change to the lower quadrants of the sun because of the cycle.

Furthermore, the range of values in total.counts is nearly halved from Set 1 to Set 2, this includes the mean and standard deviation values as well. Energy.kev.i ranges are the same in both sets, but the difference in means and standard deviations signifies that there are more values that are closer to 6 than there are values that are closer to 300 in Set 1 when compared to Set 2. The same is true for energy.kev.f., but the larger gap in standard deviations are most likely a result of energy.kev.f having a larger range of values than energy.kev.i in both sets. The range of values and standard deviation are similar in Set 1 and 2 as well when it comes to x.pos.asec and y.pos.asec, meaning that the parameters of these data sets are very, very similar. The means differ however, which lets us know that the distribution of values fall pretty evenly in between Set 2 but heavily leans towards negative values in Set 1. Lastly, we have the energy.kev.mid, which we calculated on our own as seen in our Method 2 documentation. The ranges are almost doubled in Set 2 compared to Set 1, which is similar to what we observed in the total.counts. The means for both are very close in range with a similar standard deviation as well, meaning we see many more values in the -1 to 1 range in both sets. We have included the month and year values as well, which, as expected, are very similar to one another. We see more values in the first year than the second in both, and the standard deviation of 1 proves that to be true.

15 Member Contributions

- Brayan Gutierrez: Helped batch the solar flare data, helped develop Hotspot Detection, helped make Method 1 run more efficiently, developed the energy midpoint calculation code, helped with briefly describing methods in report, collected all the visuals needed for time series and intensities, compared the time series in Task 2, compared the intensity maps between sets 1 and 2 in Task 3.

- Jericka Ledezma: Helped batch the solar flare data, and implemented the method to help run loop batches through the fetch intensity methods. Briefly helped with the analysis done in task 3.

- Katie To: Helped batch the solar flare data, helped develop hotspot detection, helped briefly describe the methods in the report, helped determine threshold, compiled the time series into animated gifs, summarized the individual time series, found descriptive statistics, helped analyze the descriptive statistics between sets.

- Ulises Ramirez: Helped batch the solar flare data, Developed Methods 1 and 2 for task 1, created the intensity map code, helped develop hotspot detection, added more details to Methods 1 and 2 explanations in Task 1, compared intensity maps in Task 1

16 Sources

NASA. (2021, July 22). *What is the solar cycle?*. NASA. https://spaceplace.nasa.gov/solar-cycles/en/