LAB4

November 4, 2019

1 Lab 4, Gautam Kovoor, Section AD

[204]:
```
%matplotlib inline
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import scipy
from scipy import stats
from scipy.stats import poisson
import h5py

#this sets the size of the plot to something useful
plt.rcParams["figure.figsize"] = (20,15)
```

2 Checking the Data

We import out data set from the Gamma Ray file and store it into an array. We have to make sure our data follows the following format.

Time (GPS Seconds)	Solar Phase (Degrees)	Earth Longitude (Degrees)	Gamma-Ray Counts

[205]:
```
hf = h5py.File('gammaray_lab4.h5', 'r')
hf.keys()
data = np.array(hf.get('data'))
time = data[0,:]
solarPhase = data[1,:]
earthLongitude = data[2,:]
gammaRays = data[3,:]
hf.close()
```

3 Problem 1

3.1 Part 1

When it comes to exploring the data we are going to follow the Scavenger Hunt approach we discussed in lecture. We will simply start off by making plots of the data and seeing if we can find any relationships.

Each column of data has 25,920,001 entries so we need to divide up our data into segments so we can spot any trends. For all our plots we're going to look for trends in Solar Phase, Earth Longitude and Particle Counts against Time. Specifically, we're going to look for any patterns we see repeating.

In terms of calculations we are told that there number of particles is taken every 100 milliseconds which translates to every $100/1000 = 0.1$ seconds. We are also told that the orbit is roughly 90 minutes which is $90 \times 60 = 5400$ seconds.

Below we have plotted the Solar Phase against the Time, we have two plots, one over the course of one orbit and the other over the course of multiple orbits to show that the orbit time is consistent and repeats over the entire data set.

[206]:
```
fig = plt.figure(figsize=(10,5))
ax1 = fig.add_subplot(2,1,1)
ax1.set_title("Position of the Sun (Degrees) over Time (GPS Seconds)")
ax1.set_xlabel("Time (GPS Seconds)")
ax1.set_ylabel("Position of the Sun (Degrees)")
ax1.plot(time[0:5400], solarPhase[0:5400])

ax2 = fig.add_subplot(2,1,2)
ax2.set_xlabel("Time (GPS Seconds)")
ax2.set_ylabel("Position of the Sun (Degrees)")
ax2.plot(time[5400:200000], solarPhase[5400:200000])
fig.tight_layout()
```

Our second plots below show the Position of the Space Craft over time. Again we can see that the orbit time is consistent and repeats over the data set.

[207]:

```
fig = plt.figure(figsize=(10,5))
ax = fig.add_subplot(1,1,1)
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_color('none')
ax.spines['left'].set_color('none')
ax.spines['right'].set_color('none')
ax.tick_params(labelcolor='w', top='off', bottom='off', left='off', right='off')
ax.set_ylabel("Position of the Spacecraft (Degrees)")

ax1 = fig.add_subplot(2,1,1)
ax1.set_title("Position of the Spacecraft (Degrees) over Time (GPS Seconds)")
ax1.set_xlabel("Time (GPS Seconds)")
ax1.plot(time[0:44000], earthLongitude[0:44000])

ax2 = fig.add_subplot(2,1,2)
ax2.set_xlabel("Time (GPS Seconds)")
ax2.plot(time[5400:200000], earthLongitude[5400:200000])

fig.tight_layout()
```

Now that we have seen how the orbit values for both the Spacecraft and Solar phase repeat over time, we can compare the Particle Counts to these orbits and see if we have an repetitions in the number of particle counts.

As we can see below, there is a repeated amount of particle counts every 5400 seconds which is roughly the 90 minute orbit time of the Space craft. We plot this over the sample size and see that this repeats consistently.

[208]:
```
fig = plt.figure(figsize=(10,5))
ax1 = fig.add_subplot(2,1,1)
ax1.set_xlabel("Time (GPS Seconds)")
ax1.set_ylabel("Gamma Ray Counts")
ax1.plot(time[5400:200000], gammaRays[5400:200000])

ax2 = fig.add_subplot(2,1,2)
ax2.set_xlabel("Time (GPS Seconds)")
ax2.set_ylabel("Position of the Sun (Degrees)")
ax2.plot(time[5400:200000], solarPhase[5400:200000])

fig.tight_layout()
```

The same repeated particle counts can be seen in the below plot of Gamma Ray Counts and Position of the Sun over time. Every 5400 seconds we see a repeat in the amount of particle counts.

[209]:
```
fig = plt.figure(figsize=(10,5))
ax1 = fig.add_subplot(2,1,1)
ax1.set_xlabel("Time (GPS Seconds)")
ax1.set_ylabel("Gamma Ray Counts")
ax1.plot(time[5400:200000], gammaRays[5400:200000])

ax2 = fig.add_subplot(2,1,2)
ax2.set_xlabel("Time (GPS Seconds)")
ax2.set_ylabel("Position of the Spacecraft (Degrees)")
ax2.plot(time[5400:200000], earthLongitude[5400:200000])
```

```
fig.tight_layout()
```

3.2 Part 2

Our background represents the number of particle counts we have, and we need to pick the right distribution to model this. In this case, we are simply trying to model the number of events happening in a given time interval. This means we can use the Poisson distribution to model the background. Below is a plot of the probability mass function of the number of Gamma Ray counts. We can see that the mean of the number of Particle Counts is roughly 7.1.

[302]:
```
fig = plt.figure(figsize=(5,5))
ax = fig.add_subplot(1,1,1)
mu = np.mean(gammaRays)
x = np.arange(poisson.ppf(0.01, mu), poisson.ppf(0.99, mu))
ax.plot(x, poisson.pmf(x, mu))
```

[302]:
```
[<matplotlib.lines.Line2D at 0x1f7a1148208>]
```

The particle count varies based on the position of the satellite in the Earth's orbit and the position of the Sun, it increases towards towards the end of the orbit before decreasing again. This is shown in the plots above.

Although the increase in particle counts is obvious, it appears that with each successive orbit the particle counts increase to a higher value. Below we plot the Gamma Ray Counts against time for a larger sample size. We can see that in multiple orbits the maximum value for particle counts keeps changing.

[211]:
```
fig = plt.figure(figsize=(10,5))
ax1 = fig.add_subplot(2,1,1)
ax1.set_xlabel("Time (GPS Seconds)")
ax1.set_ylabel("Gamma Ray Counts")
ax1.plot(time[5400:400000], gammaRays[5400:400000])
```

[211]:
```
[<matplotlib.lines.Line2D at 0x1f7acf1ff60>]
```

3.3 Part 4

We can calculate the 5 sigma threshold at different times by finding the mean of the interval and then using the Poisson Percent Point Function which is the inverse of CDF to find the particle count.

We know that one orbit is 5400 seconds, so we can compare the particle counts at the start and during the middle of the orbit. Our 5 sigma value is taken from http://www.physicscentral.com/buzz/blog/index.cfm?postid=5248358123737529836

[298]:
```
sigma = 0.9999994
mu = np.mean(gammaRays[4900:5400])
# PPF is the equivalent of an inverse in CDF
probability = poisson.ppf(sigma, mu)
print('The particle count at the end is: ', probability)
```

```
The particle count at the end is:  23.0
```

[301]:
```
sigma = 0.999999485
mu = np.mean(gammaRays[15000:15400])
# PPF is the equivalent of an inverse in CDF
probability = poisson.ppf(sigma, mu)
print('The particle count at the middle is: ', probability)
```

```
The particle count at the start is:  22.0
```

4 Problem 2

[231]:
```
hf2 = h5py.File('images.h5', 'r')
```

[232]:
```
hf2.keys()
```

[232]:
```
KeysView(<HDF5 file "images.h5" (mode r)>)
```

[233]:
```python
# Since .keys() wasn't outputting names I wrote a for loop to get the names of␣
↪the keys
for key in hf2.keys():
    print(key)
```

```
image1
imagestack
```

[246]:
```python
images = np.array(hf2.get('imagestack'))
```

[253]:
```python
fig = plt.figure(figsize=(10,10))
fig.suptitle("Plot of Images")
for i in range(1, 11):
    fig.add_subplot(5, 2, i)
    plt.imshow(images[:, :, i-1])
plt.show()
```

4.1 Part 2

I would say there is no signal contamination because the background doesn't vary over time and is spatially consistent.