pulsar6

August 25, 2022

1 Pulsar Emission Data Analysis

[ ]:

```
#currently including any and all Imports that maybe needed for the project.
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn import linear_model
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.linear_model import LogisticRegression, LinearRegression
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.feature_selection import RFE
import datetime as dt
from sklearn.cluster import KMeans
from sklearn.metrics import pairwise_distances
from scipy.cluster.hierarchy import linkage, dendrogram, cut_tree
from scipy.spatial.distance import pdist
from sklearn.feature_extraction.text import TfidfVectorizer
import matplotlib.dates as mdates
from scipy.stats import pearsonr
from scipy import stats
import statistics
import math
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.stattools import acf, pacf
from statsmodels.tsa.tsatools import lagmat
```

Section for extracting from a tar file.

Currently implemented for original TAR File structure.

[ ]:

```
#This is also found in the main file under tarunzip.py
import tarfile
import os
import sys
```

```
#tar = tarfile.open("pulseTarFile.tar")
#tar.extractall('./Data')
#tar.close()
```

1.1 Beginning of Exploration

1.1.1 Examining the data

In this section we are determining the total integrity of the data to determine if further comprehensive data cleaning and uniforming processes are needed.

```
[ ]: colnames = ['Pulse Number', 'Brightness', 'Uncertainty']
     pulsar6 = pd.read_csv("Data/J1644-4559.pulses", sep = ' ', header = None, names␣
      ↪= colnames)
```

```
[ ]: pulsar6.shape
```

```
[ ]: (698, 3)
```

```
[ ]: pulsar6.head(25)
```

```
[ ]:     Pulse Number  Brightness  Uncertainty
     0              1    0.634671     0.002761
     1              2    0.736945     0.005207
     2              3    0.693834     0.002706
     3              4    1.021866     0.010184
     4              5    0.673845     0.006236
     5              6    0.676883     0.004763
     6              7    0.527039     0.002422
     7              8    0.673417     0.003174
     8              9    0.357076     0.002848
     9             10    0.661704     0.005588
     10            11    0.545564     0.003835
     11            12    0.494655     0.003145
     12            13    0.804260     0.005258
     13            14    0.513362     0.005700
     14            15    0.477025     0.002945
     15            16    0.399571     0.004712
     16            17    0.188069     0.002452
     17            18    0.748592     0.005468
     18            19    0.723437     0.004548
     19            20    0.960154     0.006765
     20            21    0.707715     0.006011
     21            22    1.074550     0.006831
     22            23    0.961340     0.006617
     23            24    0.754457     0.004117
     24            25    0.773151     0.004920
```

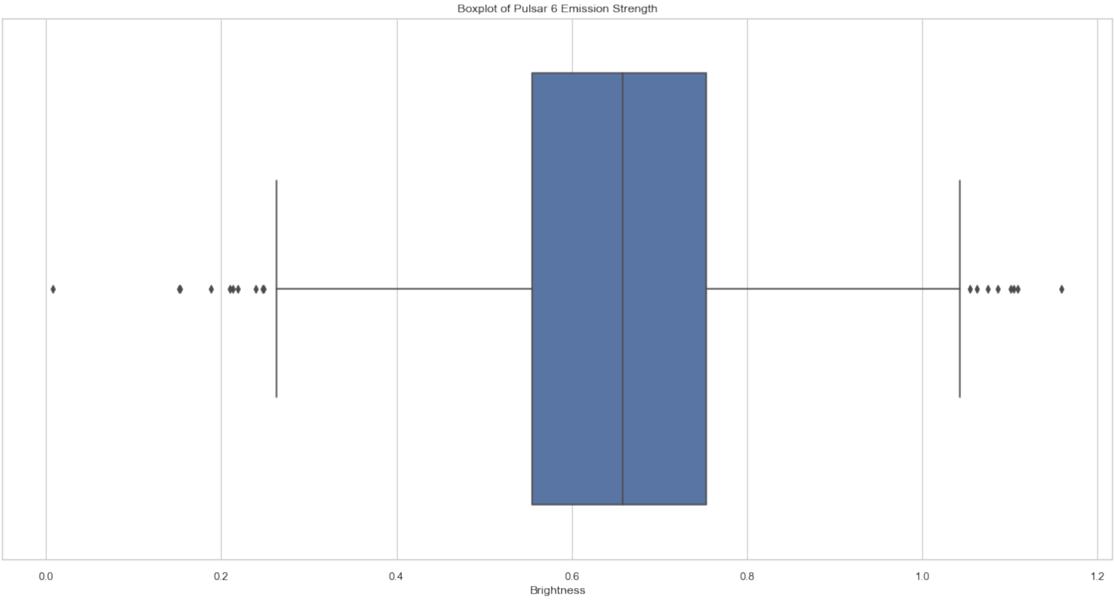
```
[ ]: pulsar6.describe()
```

```
[ ]:        Pulse Number  Brightness  Uncertainty
     count     698.00000  698.000000   698.000000
     mean      349.50000    0.654319     0.004445
     std       201.63953    0.163945     0.001855
     min         1.00000    0.007642     0.002129
     25%       175.25000    0.555267     0.003086
     50%       349.50000    0.658295     0.003951
     75%       523.75000    0.753396     0.005349
     max       698.00000    1.159334     0.016097
```

```
[ ]: pulsar6["Brightness"].describe()
```

```
[ ]: count    698.000000
     mean       0.654319
     std        0.163945
     min        0.007642
     25%        0.555267
     50%        0.658295
     75%        0.753396
     max        1.159334
     Name: Brightness, dtype: float64
```

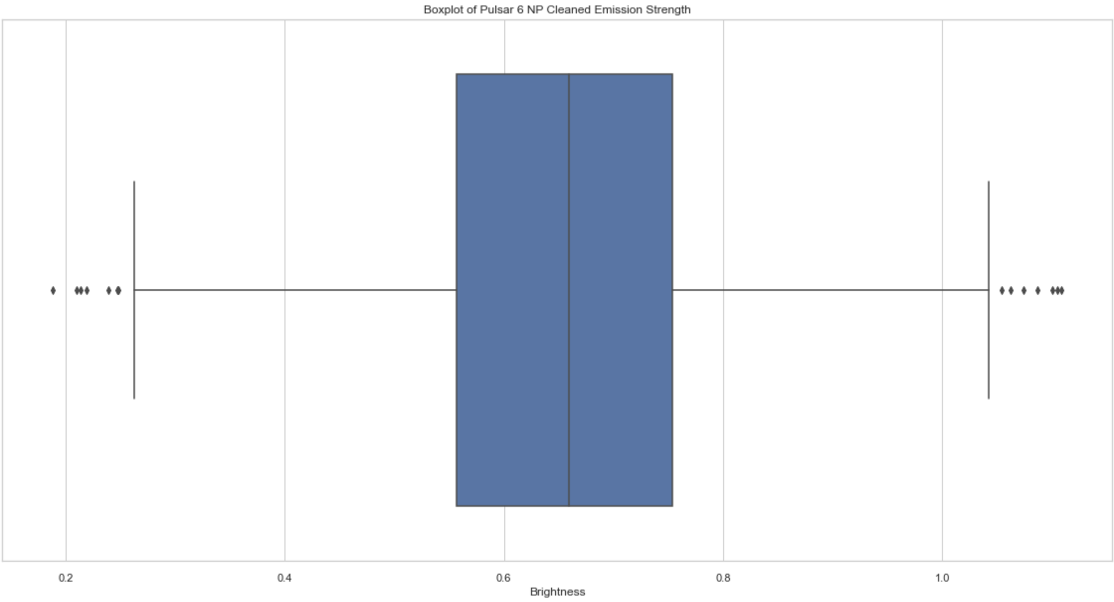
```
[ ]: plt.figure(figsize=(20,10))
     sns.set_theme(style="whitegrid")
     ax = sns.boxplot(x=pulsar6["Brightness"]).set_title("Boxplot of Pulsar 6␣
      ↪Emission Strength")
```



```
[ ]: #numpy method of outlier removal

     pulsar6npcleaned = pulsar6[(np.abs(stats.zscore(pulsar6["Brightness"])) <3)]
     pulsar6npcleaned
```

```
[ ]:      Pulse Number  Brightness  Uncertainty
     0               1    0.634671     0.002761
     1               2    0.736945     0.005207
     2               3    0.693834     0.002706
     3               4    1.021866     0.010184
     4               5    0.673845     0.006236
     ..            …          …           …
     693           694    0.776083     0.008928
     694           695    0.625382     0.006018
     695           696    0.647559     0.003765
     696           697    0.312449     0.002901
     697           698    0.548353     0.009056

     [694 rows x 3 columns]
```

```
[ ]: plt.figure(figsize=(20,10))
     sns.set_theme(style="whitegrid")
     ax = sns.boxplot(x=pulsar6npcleaned["Brightness"]).set_title("Boxplot of Pulsar␣
      ↪6 NP Cleaned Emission Strength")
```



```
[ ]: pulsar6npcleaned["Brightness"].describe()
```

```
[ ]: count    694.000000
     mean       0.655970
     std        0.159160
     min        0.188069
     25%        0.556461
     50%        0.658903
     75%        0.753396
     max        1.109122
     Name: Brightness, dtype: float64
```

```
[ ]: pulsar6npcleaned["Brightness"].median()
```

```
[ ]: 0.6589028
```

```
[ ]: medianpulse6 = pulsar6["Brightness"].median()
     print("Median of Pulsar6: ", medianpulse6)
     pulsar6['Binary'] = np.where(pulsar6['Brightness'] > medianpulse6, 1, 0)
```

```
Median of Pulsar6:  0.65829515
```

```
[ ]: pulsar6
```

```
[ ]:      Pulse Number  Brightness  Uncertainty  Binary
     0               1    0.634671     0.002761       0
     1               2    0.736945     0.005207       1
     2               3    0.693834     0.002706       1
     3               4    1.021866     0.010184       1
     4               5    0.673845     0.006236       1
     ..            …         …           …    …
     693           694    0.776083     0.008928       1
     694           695    0.625382     0.006018       0
     695           696    0.647559     0.003765       0
     696           697    0.312449     0.002901       0
     697           698    0.548353     0.009056       0

     [698 rows x 4 columns]
```

```
[ ]: median = pulsar6npcleaned["Brightness"].median()
     print("Median of Pulsar6 np cleaned: ", median)
     pulsar6npcleaned['Binary'] = np.where(pulsar6npcleaned['Brightness'] > median,␣
      ↪1, 0)
```

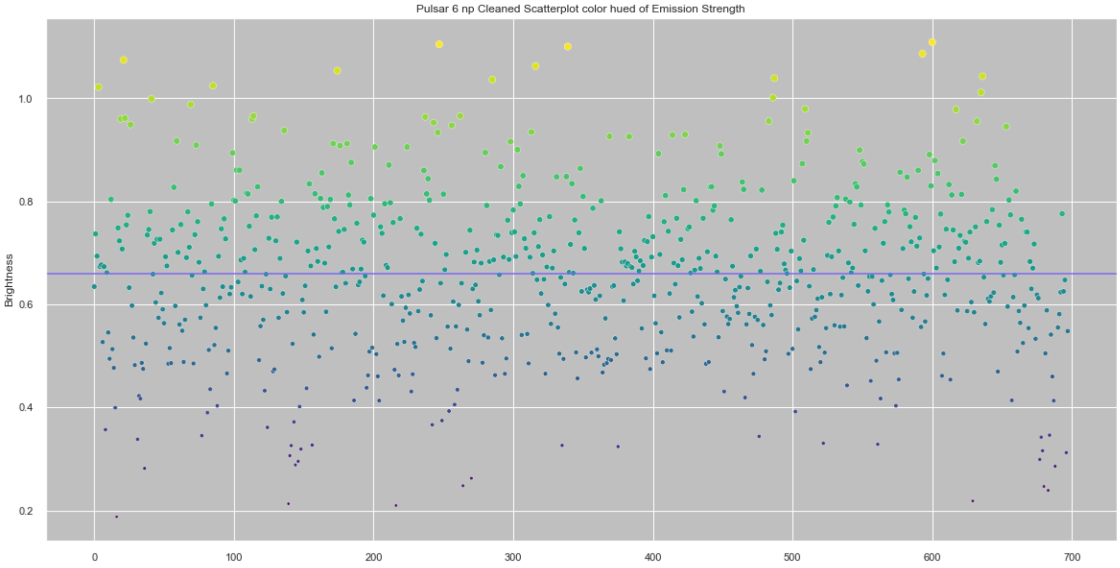
```
Median of Pulsar6 np cleaned:  0.6589028

C:\Users\oxlay\AppData\Local\Temp/ipykernel_36516/1919336679.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  pulsar6npcleaned['Binary'] = np.where(pulsar6npcleaned['Brightness'] > median,
1, 0)
```

[ ]:
```
pulsar6npcleaned
```

[ ]:
```
     Pulse Number  Brightness  Uncertainty  Binary
0               1    0.634671     0.002761       0
1               2    0.736945     0.005207       1
2               3    0.693834     0.002706       1
3               4    1.021866     0.010184       1
4               5    0.673845     0.006236       1
..            …          …           …       …
693           694    0.776083     0.008928       1
694           695    0.625382     0.006018       0
695           696    0.647559     0.003765       0
696           697    0.312449     0.002901       0
697           698    0.548353     0.009056       0

[694 rows x 4 columns]
```

[ ]:
```
plt.figure(figsize=(20,10))
sns.set_style("darkgrid", {"axes.facecolor": ".75"})
strength = pulsar6npcleaned.Brightness.values
ax = sns.scatterplot(data=pulsar6npcleaned["Brightness"], s= strength*50,␣
 ↪c=strength, cmap="viridis", marker="o").set_title('Pulsar 6 np Cleaned␣
 ↪Scatterplot color hued of Emission Strength')
ax = plt.axhline( y=0.6589028, ls='-',c='mediumslateblue')
```



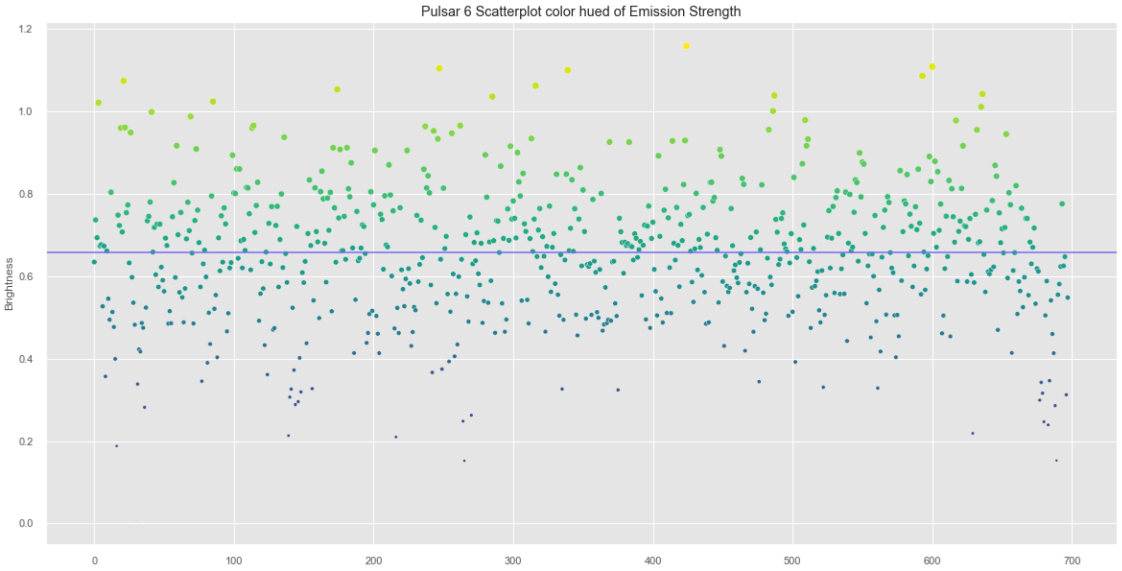
[ ]:

```
print(len(pulsar6npcleaned[(pulsar6npcleaned.Brightness > 0.6589028)]))
print(len(pulsar6npcleaned[(pulsar6npcleaned.Brightness < 0.6589028)]))
```

```
347
347
```

[ ]:

```
plt.figure(figsize=(20,10))
sns.set_style("darkgrid", {"axes.facecolor": ".75"})
strength = pulsar6.Brightness.values
plt.style.use('ggplot')
ax = sns.scatterplot(data=pulsar6["Brightness"], s= strength*50, c=strength,␣
 ↪cmap="viridis", marker="o").set_title('Pulsar 6 Scatterplot color hued of␣
 ↪Emission Strength')
ax= plt.axhline( y=0.65829515, ls='-',c='mediumslateblue')
```



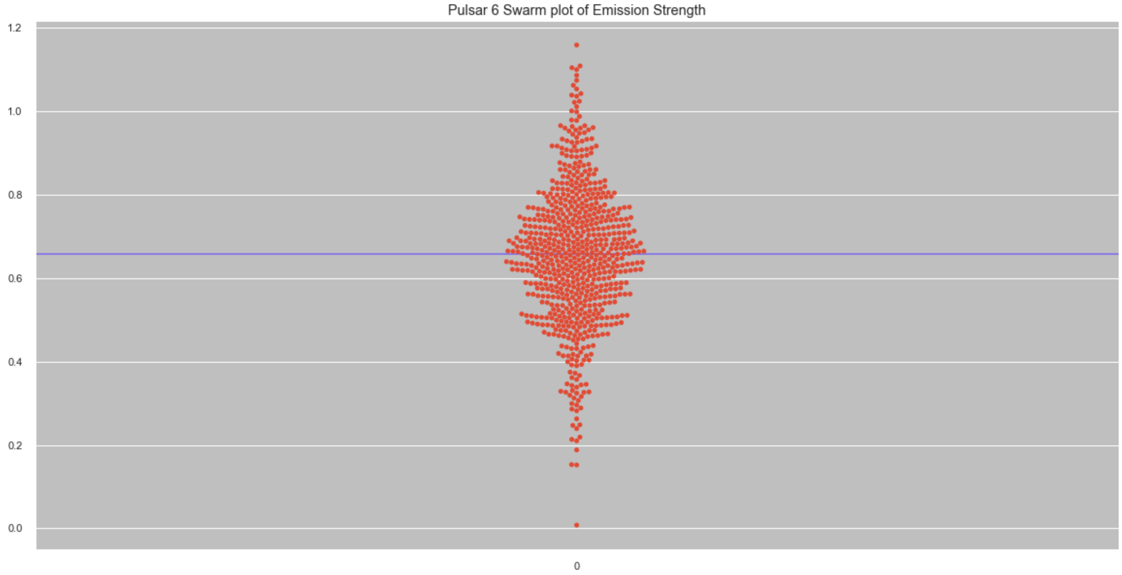
[ ]:

```
print(len(pulsar6[(pulsar6.Brightness > 0.6589028)]))
print(len(pulsar6[(pulsar6.Brightness < 0.6589028)]))
```

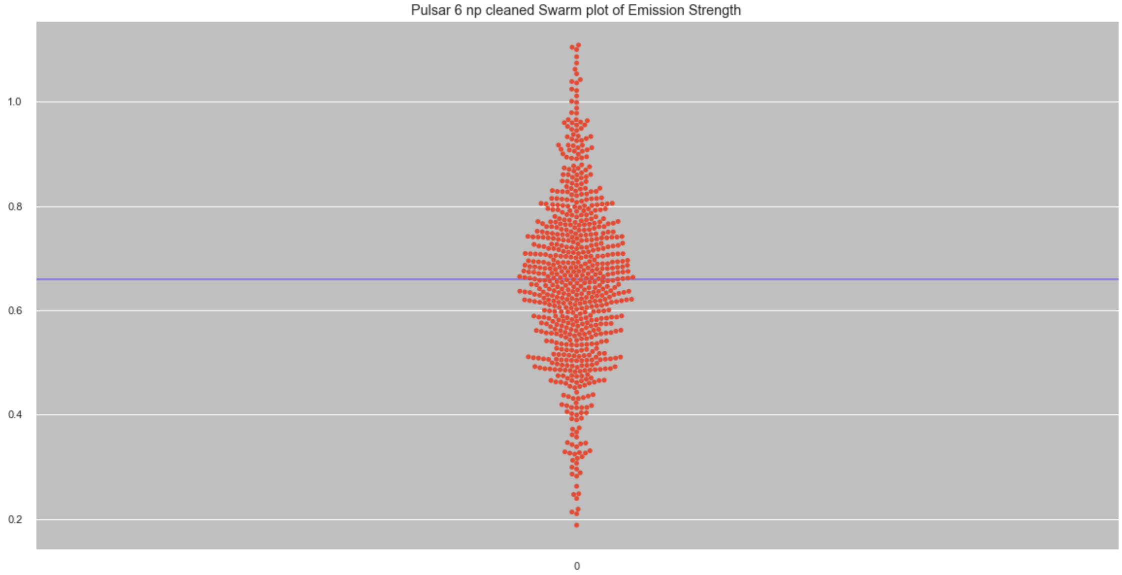
```
348
350
```

[ ]:

```
plt.figure(figsize=(20,10))
sns.set_style("darkgrid", {"axes.facecolor": ".75"})
strength = pulsar6.Brightness.values
ax = plt.axhline( y=0.65829515, ls='-',c='mediumslateblue')
ax = sns.swarmplot(data=pulsar6["Brightness"], c="blue").set_title('Pulsar 6␣
↪Swarm plot of Emission Strength')
```



[ ]:
```
plt.figure(figsize=(20,10))
sns.set_style("darkgrid", {"axes.facecolor": ".75"})
strength = pulsar6npcleaned.Brightness.values
ax = sns.swarmplot(data=pulsar6npcleaned["Brightness"]).set_title('Pulsar 6 np␣
↪cleaned Swarm plot of Emission Strength')
ax = plt.axhline( y=0.6589028, ls='-', c='mediumslateblue')
```



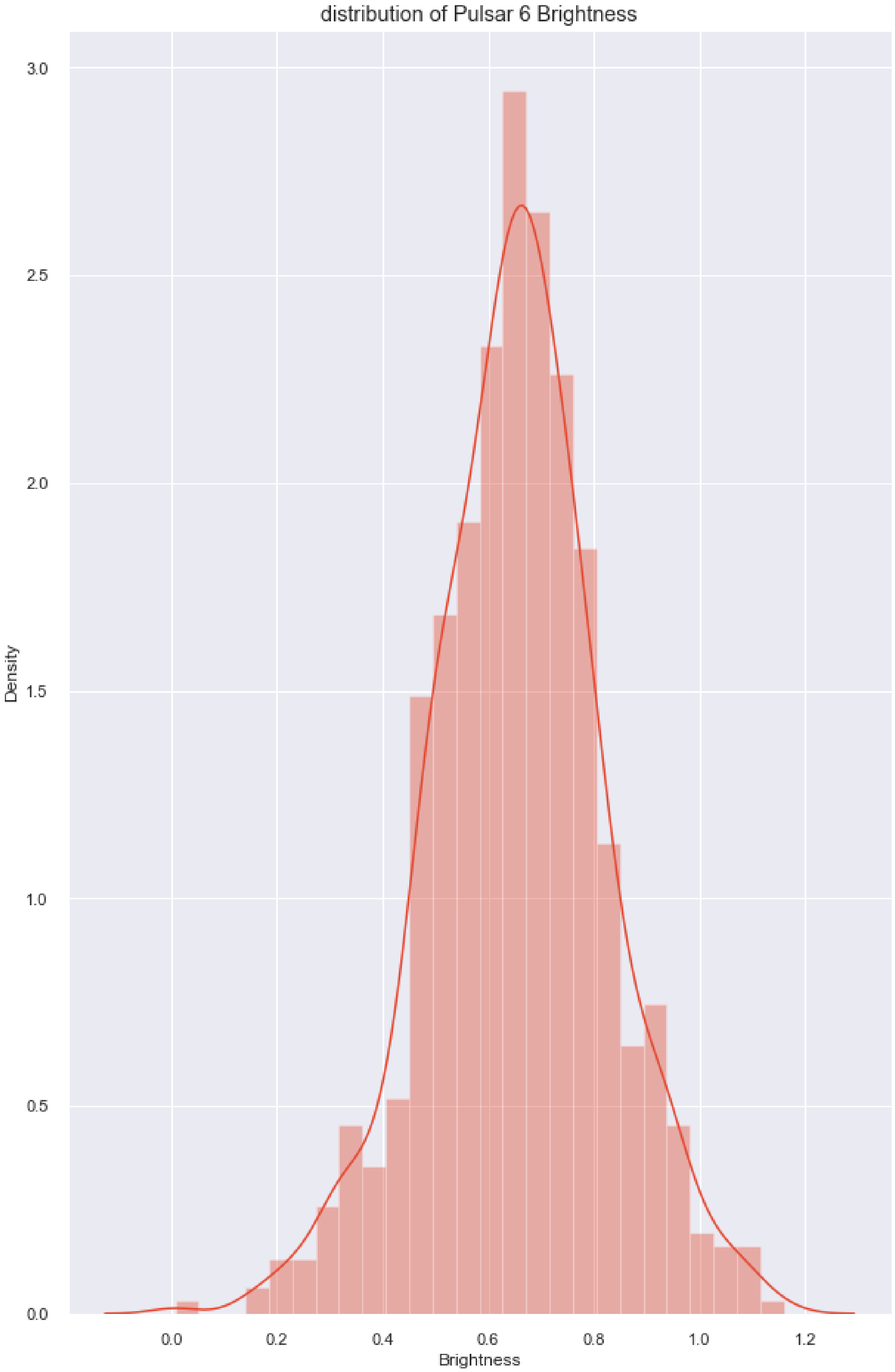
[ ]:

```
plt.figure(figsize=(10, 16))
with sns.axes_style('darkgrid'):
    sns.distplot(pulsar6.Brightness)
plt.title("distribution of Pulsar 6 Brightness")
```

```
c:\Users\oxlay\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

[ ]:

```
Text(0.5, 1.0, 'distribution of Pulsar 6 Brightness')
```


distribution of Pulsar 6 Brightness
Density
Brightness
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2

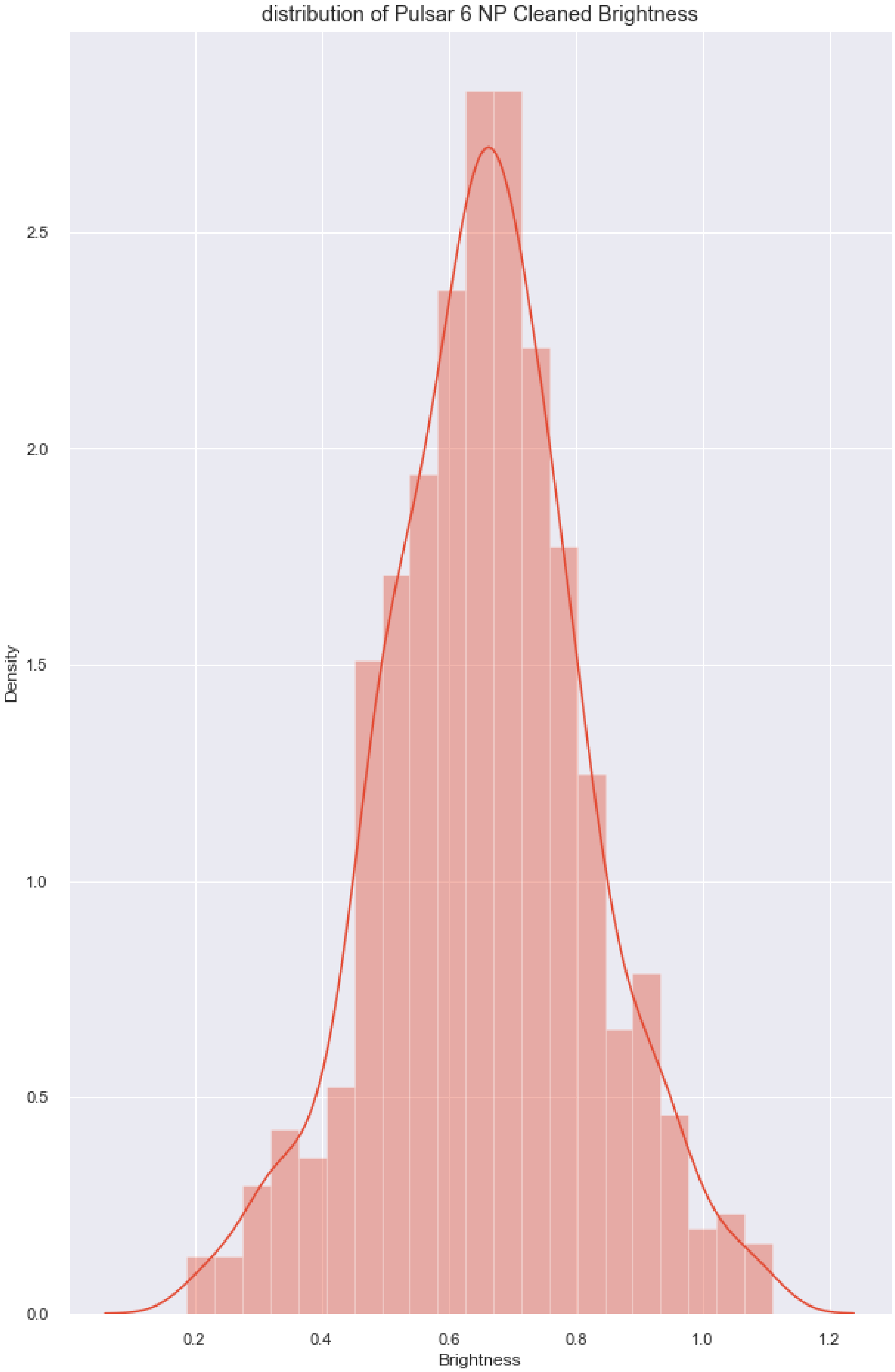
[ ]:

```python
plt.figure(figsize=(10, 16))
with sns.axes_style('darkgrid'):
    sns.distplot(pulsar6npcleaned.Brightness)
plt.title("distribution of Pulsar 6 NP Cleaned Brightness")
```

```
c:\Users\oxlay\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

[ ]:

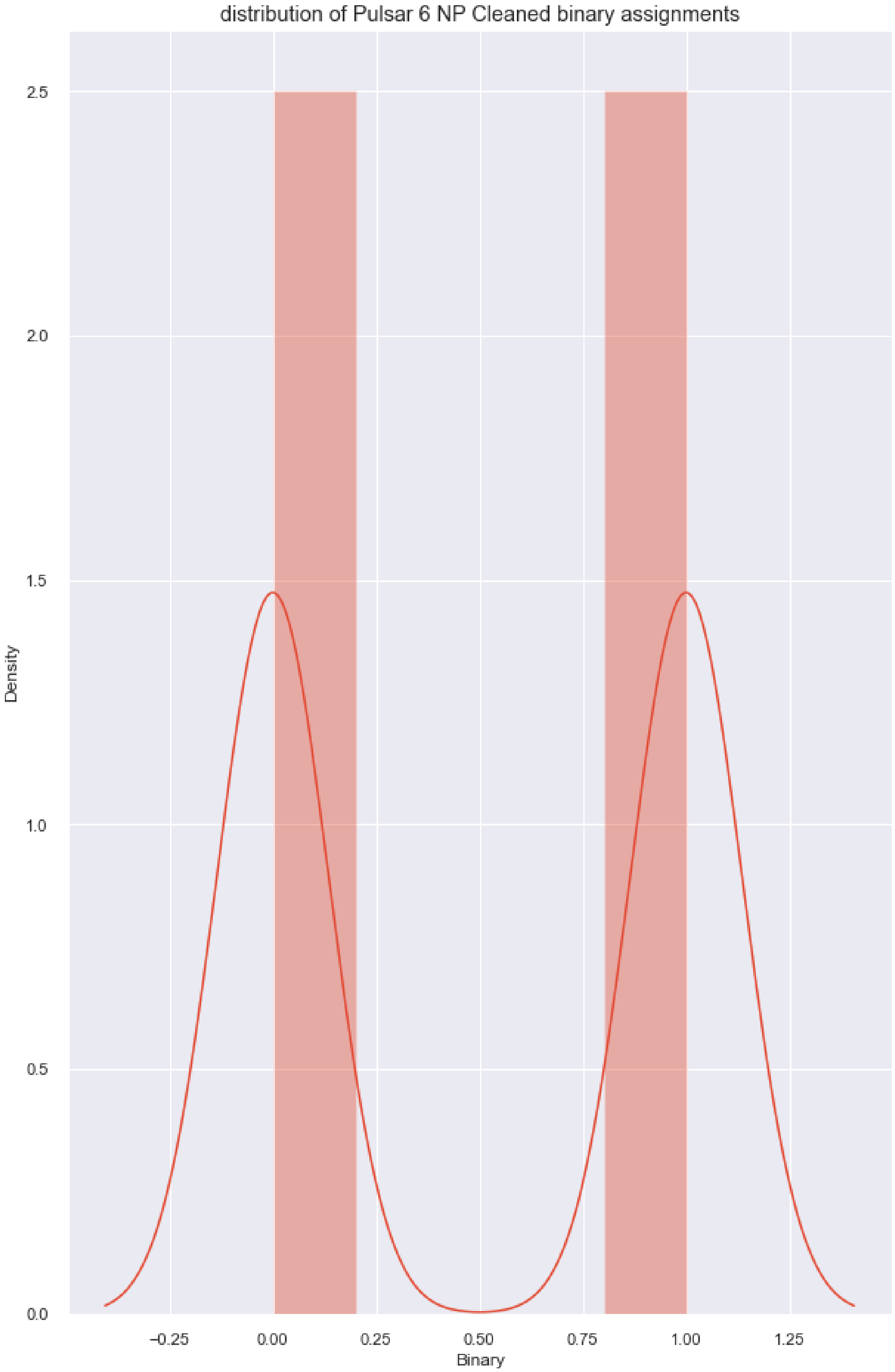
```
Text(0.5, 1.0, 'distribution of Pulsar 6 NP Cleaned Brightness')
```


distribution of Pulsar 6 NP Cleaned Brightness
Density
0.0
0.5
1.0
1.5
2.0
2.5
0.2
0.4
0.6
0.8
1.0
1.2
Brightness

```
[ ]: plt.figure(figsize=(10, 16))
     with sns.axes_style('darkgrid'):
         sns.distplot(pulsar6npcleaned.Binary)
     plt.title("distribution of Pulsar 6 NP Cleaned binary assignments")
```

```
c:\Users\oxlay\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

```
[ ]: Text(0.5, 1.0, 'distribution of Pulsar 6 NP Cleaned binary assignments')
```


distribution of Pulsar 6 NP Cleaned binary assignments
Density
0.0
0.5
1.0
1.5
2.0
2.5
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
Binary

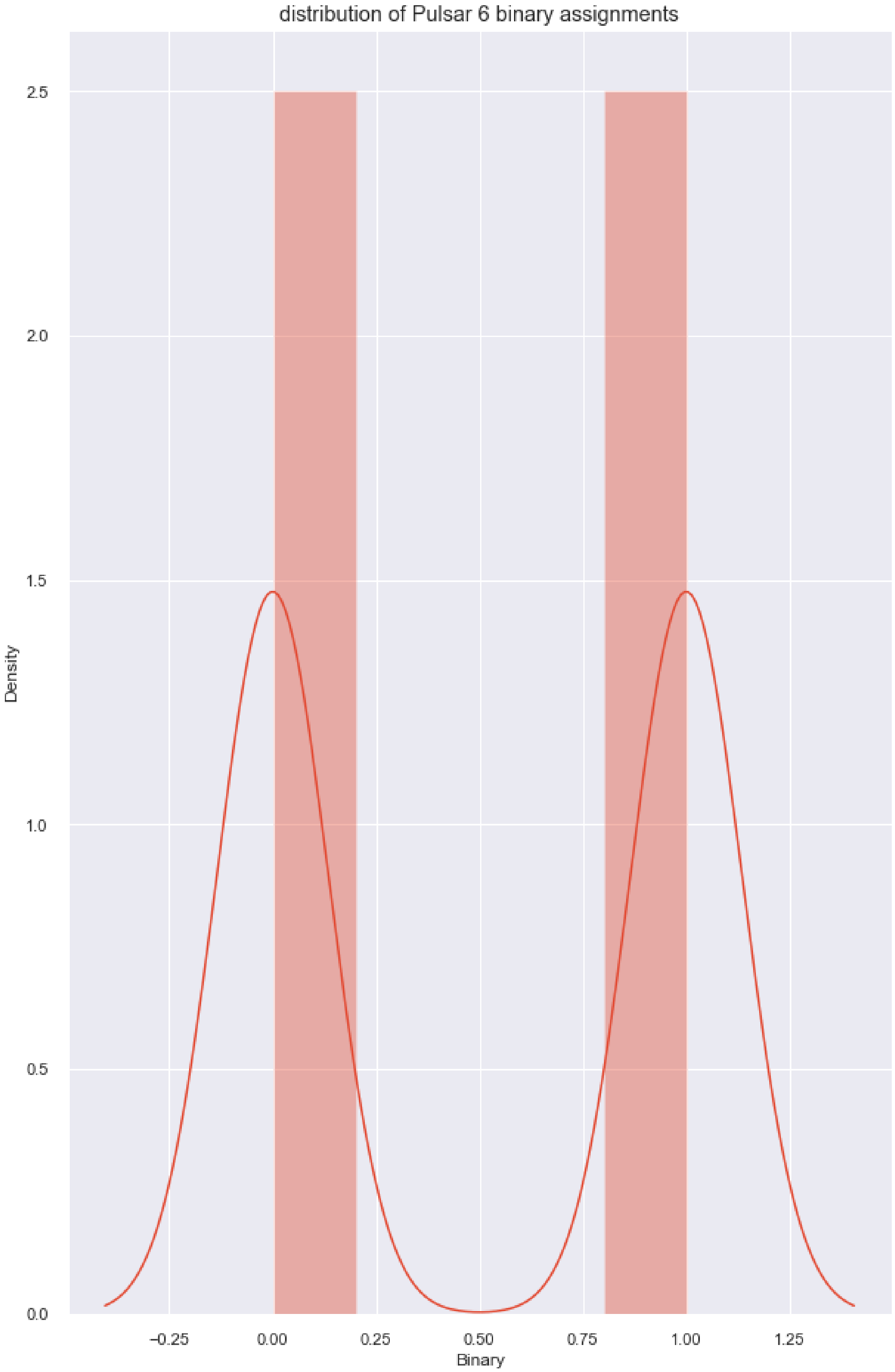
[ ]:

```python
plt.figure(figsize=(10, 16))
with sns.axes_style('darkgrid'):
    sns.distplot(pulsar6.Binary)
plt.title("distribution of Pulsar 6 binary assignments")
```

```
c:\Users\oxlay\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

[ ]:

```
Text(0.5, 1.0, 'distribution of Pulsar 6 binary assignments')
```


distribution of Pulsar 6 binary assignments
Density
0.0
0.5
1.0
1.5
2.0
2.5
Binary
−0.25
0.00
0.25
0.50
0.75
1.00
1.25

2 Preliminary runs test

2.0.1 Math Logic

$$Z = \frac{R - \tilde{R}}{s_R}$$

$$\tilde{R} = \frac{2_{n1n2}}{n1 + n2} + 1$$

$$s_R^2 = \frac{2_{n1n2}(2n1n2 - n1 - n2)}{(n1 + n2)^2(n1 + n2 - 1)}$$

link to resource: https://www.geeksforgeeks.org/runs-test-of-randomness-in-python/

$ Z_{critical} = 1.96 $ as the confidence interval level of 95% thus this is a 2 tailed test. If the probability as corrosponding to this confidence interval $ H_{null} $ will be rejected as it is not statistically significant as denoted by $|Z| > Z_{critical} $

There is also code attempting to change it from a z-score probability to a P-score for ease of understanding and clarity.

3 FUNCTION CODE FOR RUNS TEST

[ ]:

```
# MUST BE PASSED A LIST AND A INT/FLOAT



def runsTest(data, dataMedian):
    runs = 0
    above = 0
    below = 0

    for i in range(len(data)):
        if(data[i] >= dataMedian and data[i-1] < dataMedian) or (data[i] < dataMedian and data[i-1] >= dataMedian):
            runs += 1

        if(data[i] >= dataMedian):
            above += 1

        else:
            below += 1

    R = ((2*above*below)/(above+below))+1
    #sdevTemp = (2*above*below*(2*above*below-above-below))/
↪(((above+below)**2)*(above+below-1))
    #sdevTemp = (2*n1*n2*(2*n1*n2-n1-n2))/(((n1+n2)*2)*(n1+n2-1))
    Sdev = math.sqrt((2*above*below*(2*above*below-above-below))/
↪(((above+below)**2)*(above+below-1)))

    float(Sdev)
    float(R)
    float(runs)
    z = (runs-R)/Sdev
    return z
```

```
[ ]: binaryData1 = pulsar6['Binary'].tolist()
     print("pulsar6 original: ",binaryData1)

     binaryData1nooutlier = pulsar6npcleaned['Binary'].tolist()
     print("\n pulsar6 original: ", binaryData1nooutlier)
```

```
pulsar6 original:  [0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1,
1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0,
1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1,
1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1,
1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1,
0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1,
1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1,
1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1,
1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,
1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1,
0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,
0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0,
0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1,
0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1,
1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0,
0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1,
0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1,
1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1,
1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1,
0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0,
0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0,
1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0,
0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
0, 0, 0]

 pulsar6 original:  [0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1,
1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1,
0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1,
1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1,
1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1,
1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1,
1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1,
1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0,
0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1,
0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1,
1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0,
0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1,
1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0,
0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0,
0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1,
1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0,
0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1,
1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1,
1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1,
0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1,
0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
```

```
[ ]: print("Brightness Median Test")
     Zscore = abs(runsTest(binaryData1, medianpulse6))
     Pval = stats.norm.sf(abs(Zscore))*2
     print('Z Statistic is: ', Zscore)
     print('P Value is : ', Pval)

     if(Zscore >= 1.96):
         print('We reject the null Hypotheses as the Zscore greater than 1.96. Thus␣
      ↪not statistically significant.')

     if(Pval <= 0.05):
         print('We reject the null Hypotheses as the P-value is less than 0.05%.␣
      ↪Thus not statistically significant.')

     print("Binary Median Test")
     binarymedian1 = pulsar6["Binary"].median()

     Zscore = abs(runsTest(binaryData1, binarymedian1))
     Pval = stats.norm.sf(abs(Zscore))*2
     print('Z Statistic is: ', Zscore)
     print('P Value is : ', Pval)
```

```python
if(Zscore >= 1.96):
    print('We reject the null Hypotheses as the Zscore greater than 1.96. Thus␣
 ↪not statistically significant.')

if(Pval <= 0.05):
    print('We reject the null Hypotheses as the P-value is less than 0.05%.␣
 ↪Thus not statistically significant.')


print("Removed outliers from dataset")
Zscore = abs(runsTest(binaryData1nooutlier, median))
Pval = stats.norm.sf(abs(Zscore))*2
print('Z Statistic is: ', Zscore)
print('P Value is : ', Pval)

if(Zscore >= 1.96):
    print('We reject the null Hypotheses as the Zscore greater than 1.96. Thus␣
 ↪not statistically significant.')

if(Pval <= 0.05):
    print('We reject the null Hypotheses as the P-value is less than 0.05%.␣
 ↪Thus not statistically significant.')
```

```
Brightness Median Test
Z Statistic is:  4.545328792576532
P Value is :  5.48495657884083e-06
We reject the null Hypotheses as the Zscore greater than 1.96. Thus not
statistically significant.
We reject the null Hypotheses as the P-value is less than 0.05%. Thus not
statistically significant.
Binary Median Test
Z Statistic is:  4.545328792576532
P Value is :  5.48495657884083e-06
We reject the null Hypotheses as the Zscore greater than 1.96. Thus not
statistically significant.
We reject the null Hypotheses as the P-value is less than 0.05%. Thus not
statistically significant.
Removed outliers from dataset
Z Statistic is:  4.558427804288349
P Value is :  5.153797471801667e-06
We reject the null Hypotheses as the Zscore greater than 1.96. Thus not
statistically significant.
We reject the null Hypotheses as the P-value is less than 0.05%. Thus not
statistically significant.
```

4 Analysis of the preliminary data analysis

We can see here through our printouts the value of both Z Statistic based on the above Runs Test of Randomness and the approximate correlative P-value.

If the conditional prints are not activated it meants there is no statistical significance to reject the $H_{null}$

4.1 $ H_{null} $ is where the numbers are randomly generated and sequenced

4.2 $ H_{alt} $ is where the numbers are not randomly generated or sequenced

Further testing can be done with more variety of datasets with pythonic libraries and R librariest such as NIST and Rrandtest (placeholders cant remember their names)

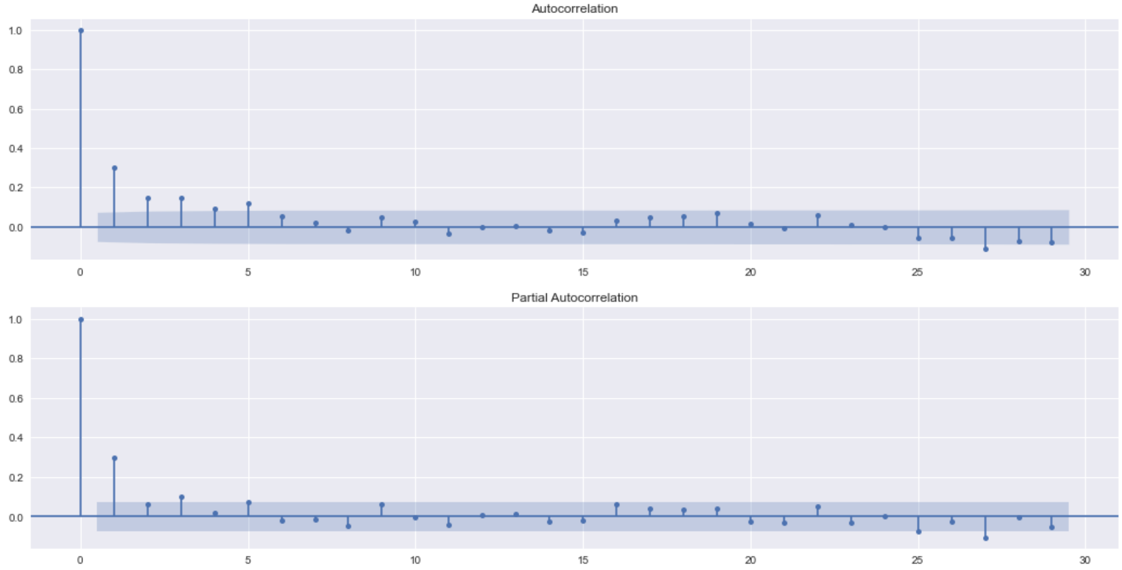
5 Below we begin autocorrelation and autocovariance analysis

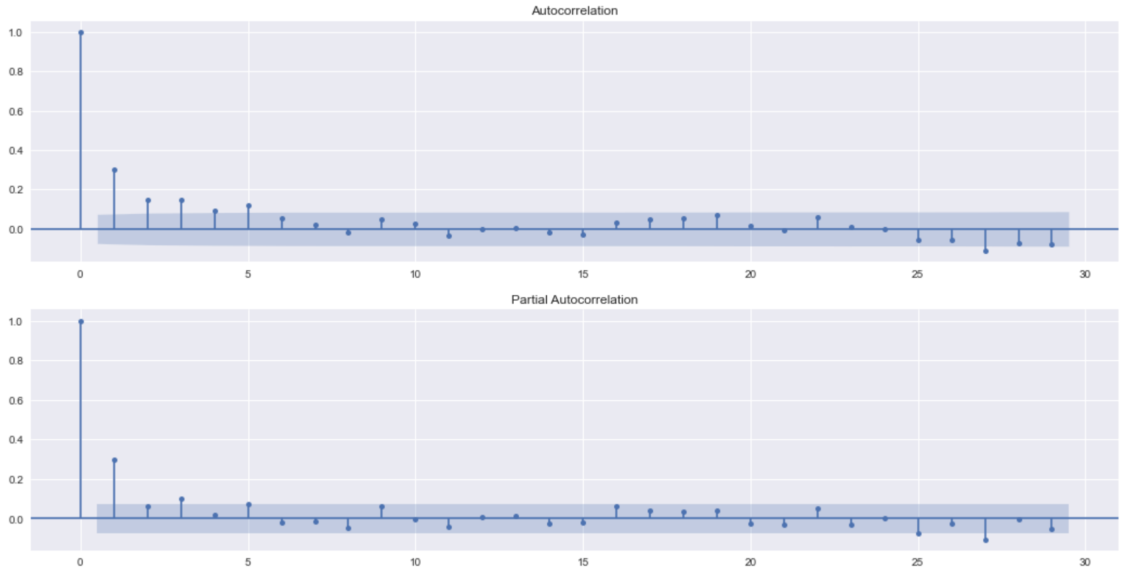
To get started with this I am playing around with guide from: https://towardsdatascience.com/a-step-by-step-guide-to-calculating-autocorrelation-and-partial-autocorrelation-8c4342b784e8

[ ]:

```
plt.style.use("seaborn")
plt.rcParams["figure.figsize"] = (18, 9)

fig, ax = plt.subplots(2,1)

plot_acf(pulsar6['Brightness'], ax=ax[0])
plot_pacf(pulsar6['Brightness'], ax=ax[1], method="ols")
```

[ ]:





```
[ ]: acf(pulsar6['Brightness'], nlags=10)
```

```
c:\Users\oxlay\anaconda3\lib\site-packages\statsmodels\tsa\stattools.py:667:
FutureWarning: fft=True will become the default after the release of the 0.12
release of statsmodels. To suppress this warning, explicitly set fft=False.
  warnings.warn(
```

```
[ ]: array([ 1.        ,  0.29929122,  0.14656878,  0.14948301,  0.09384681,
             0.11707783,  0.05493324,  0.02160374, -0.01711482,  0.04777   ,
             0.02563995])
```

```
[ ]: acfpulsar6 = pd.DataFrame()
     for lag in range(0,11):
         acfpulsar6[f"B_lag_{lag}"] = pulsar6['Brightness'].shift(lag)


     acfpulsar6
```

```
[ ]:      B_lag_0   B_lag_1   B_lag_2   B_lag_3   B_lag_4   B_lag_5   B_lag_6  \
     0    0.634671       NaN       NaN       NaN       NaN       NaN       NaN
     1    0.736945  0.634671       NaN       NaN       NaN       NaN       NaN
     2    0.693834  0.736945  0.634671       NaN       NaN       NaN       NaN
     3    1.021866  0.693834  0.736945  0.634671       NaN       NaN       NaN
     4    0.673845  1.021866  0.693834  0.736945  0.634671       NaN       NaN
     ..        …         …         …         …         …         …
     693  0.776083  0.623757  0.581248  0.555266  0.152886  0.286132  0.413354
     694  0.625382  0.776083  0.623757  0.581248  0.555266  0.152886  0.286132
     695  0.647559  0.625382  0.776083  0.623757  0.581248  0.555266  0.152886
```

```
696  0.312449  0.647559  0.625382  0.776083  0.623757  0.581248  0.555266
697  0.548353  0.312449  0.647559  0.625382  0.776083  0.623757  0.581248

      B_lag_7   B_lag_8   B_lag_9  B_lag_10
0         NaN       NaN       NaN       NaN
1         NaN       NaN       NaN       NaN
2         NaN       NaN       NaN       NaN
3         NaN       NaN       NaN       NaN
4         NaN       NaN       NaN       NaN
..        …         …         …         …
693  0.460095  0.541486  0.346502  0.239302
694  0.413354  0.460095  0.541486  0.346502
695  0.286132  0.413354  0.460095  0.541486
696  0.152886  0.286132  0.413354  0.460095
697  0.555266  0.152886  0.286132  0.413354

[698 rows x 11 columns]
```

```
[ ]: acfpulsar6.corr()["B_lag_0"].values
```

```
[ ]: array([ 1.        ,  0.29938402,  0.14710414,  0.15003691,  0.09455452,
             0.11800036,  0.05537751,  0.02179885, -0.01724535,  0.04863954,
             0.02621294])
```

5.0.1 Getting every 5th as per the auto correlation

5.0.2 Creating a new set of discrete 100 sets and examining them specifically

5.0.3 Further Random testing to move into extensive testing

Getting every 5th as per the auto correlation

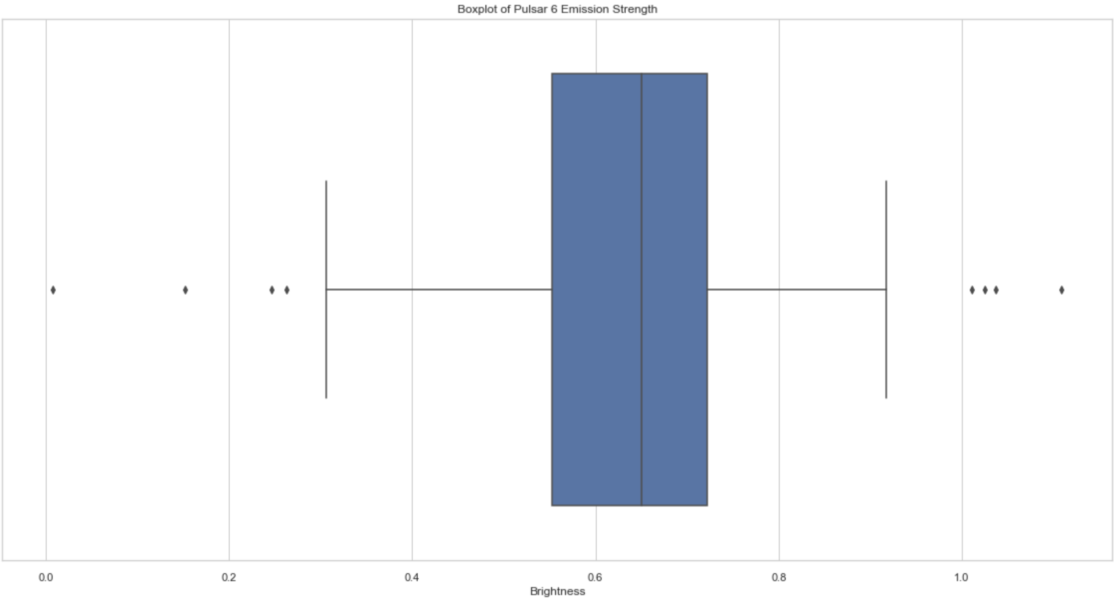
```
[ ]: held5ths = pulsar6[pulsar6.index % 5 == 0]
     held5ths
```

```
[ ]:      Pulse Number  Brightness  Uncertainty  Binary
     0               1    0.634671     0.002761       0
     5               6    0.676883     0.004763       1
     10             11    0.545564     0.003835       0
     15             16    0.399571     0.004712       0
     20             21    0.707715     0.006011       1
     ..            …         …            …         …
     675           676    0.618826     0.002507       0
     680           681    0.246916     0.004276       0
     685           686    0.541486     0.003149       0
     690           691    0.555266     0.003657       0
     695           696    0.647559     0.003765       0

     [140 rows x 4 columns]
```

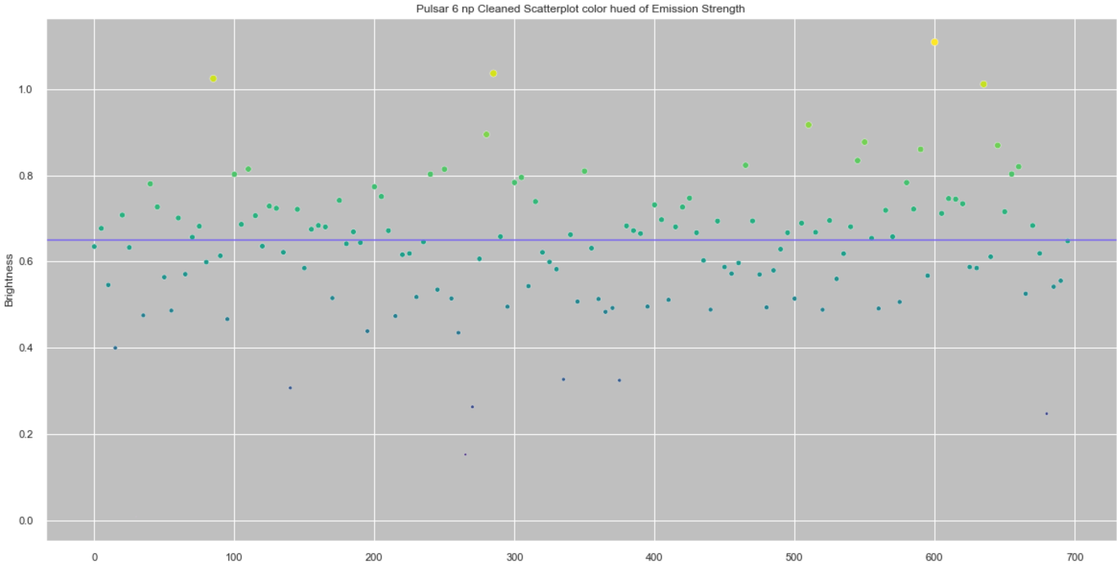
```
[ ]: medianheld5ths = held5ths["Brightness"].median()
     medianheld5ths
```

```
[ ]: 0.6508051
```

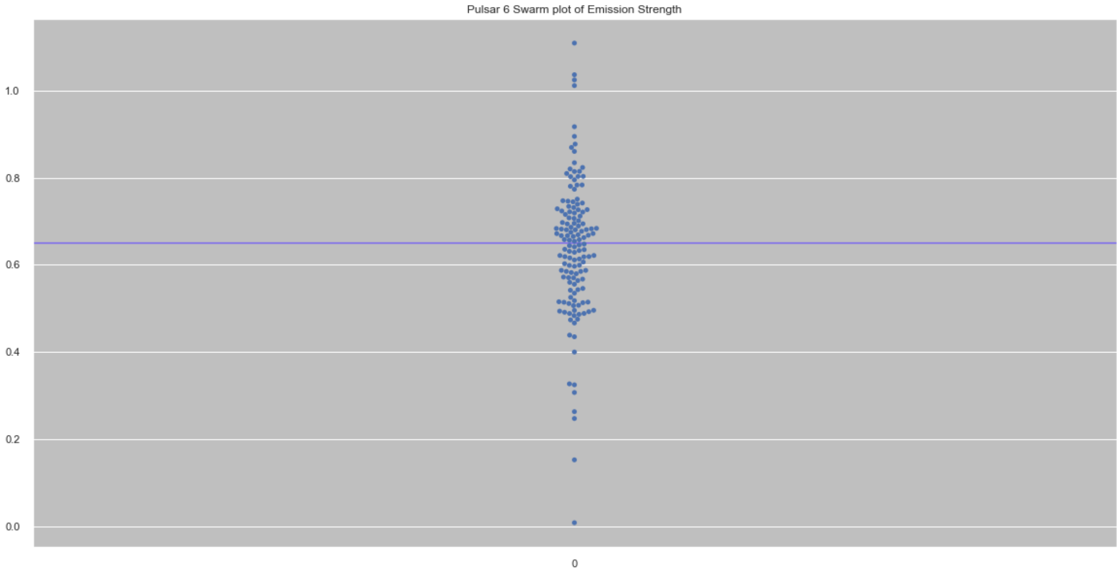
```
[ ]: plt.figure(figsize=(20,10))
     sns.set_theme(style="whitegrid")
     ax = sns.boxplot(x=held5ths["Brightness"]).set_title("Boxplot of Pulsar 6␣
      ↪Emission Strength")
```



```
[ ]: plt.figure(figsize=(20,10))
     sns.set_style("darkgrid", {"axes.facecolor": ".75"})
     strength = held5ths.Brightness.values
     ax = sns.scatterplot(data=held5ths["Brightness"], s= strength*50, c=strength,␣
      ↪cmap="viridis", marker="o").set_title('Pulsar 6 np Cleaned Scatterplot color␣
      ↪hued of Emission Strength')
     ax = plt.axhline( y=0.6508051, ls='-',c='mediumslateblue')
```



[ ]:
```
plt.figure(figsize=(20,10))
sns.set_style("darkgrid", {"axes.facecolor": ".75"})
strength = held5ths.Brightness.values
ax = plt.axhline( y=0.6508051, ls='-',c='mediumslateblue')
ax = sns.swarmplot(data=held5ths["Brightness"], c="blue").set_title('Pulsar 6␣
 ↪Swarm plot of Emission Strength')
```



[ ]:
```
print(len(held5ths[(held5ths.Brightness > 0.6508051)]))
print(len(held5ths[(held5ths.Brightness < 0.6508051)]))
```

```
70
70
```

isolating every 100 rows into discrete sets.

[ ]:

```
size = 100
N = int(len(pulsar6)/size)
pulsarsubframes = [pulsar6.iloc[i*size:(i+1)*size].copy() for i in range(N+1)]
#pulsarsubframes[-1]

frame1 = pulsarsubframes[0]
frame2 = pulsarsubframes[1]
frame3 = pulsarsubframes[2]
frame4 = pulsarsubframes[3]
frame5 = pulsarsubframes[4]
frame6 = pulsarsubframes[5]
frame7 = pulsarsubframes[6]

medianframe1 = frame1["Brightness"].median()
print("Median of Pulsar6: ", medianframe1)
frame1['Binary'] = np.where(frame1['Brightness'] > medianframe1, 1, 0)

medianframe2 = frame2["Brightness"].median()
print("Median of Pulsar6: ", medianframe2)
frame2['Binary'] = np.where(frame2['Brightness'] > medianframe2, 1, 0)

medianframe3 = frame3["Brightness"].median()
print("Median of Pulsar6: ", medianframe3)
frame3['Binary'] = np.where(frame3['Brightness'] > medianframe3, 1, 0)

medianframe4 = frame4["Brightness"].median()
print("Median of Pulsar6: ", medianframe4)
frame4['Binary'] = np.where(frame4['Brightness'] > medianframe4, 1, 0)

medianframe5 = frame5["Brightness"].median()
print("Median of Pulsar6: ", medianframe5)
frame5['Binary'] = np.where(frame5['Brightness'] > medianframe5, 1, 0)

medianframe6 = frame6["Brightness"].median()
print("Median of Pulsar6: ", medianframe6)
frame6['Binary'] = np.where(frame6['Brightness'] > medianframe6, 1, 0)

medianframe7 = frame7["Brightness"].median()
print("Median of Pulsar6: ", medianframe7)
frame7['Binary'] = np.where(frame7['Brightness'] > medianframe7, 1, 0)
```

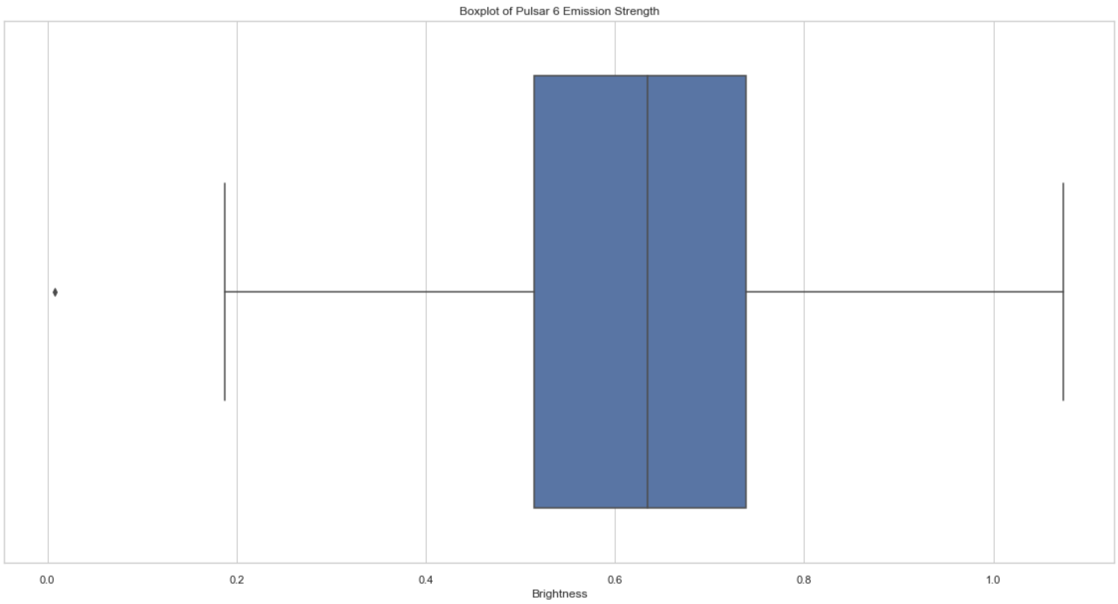
```
Median of Pulsar6:  0.63457545
Median of Pulsar6:  0.6688056
Median of Pulsar6:  0.63955675
Median of Pulsar6:  0.66777675
Median of Pulsar6:  0.6605900499999999
Median of Pulsar6:  0.65585835
Median of Pulsar6:  0.6504474499999999
```

```
[ ]: framebinary = []
```

```
[ ]: print(frame1)

     storeover1 = len(frame1[(frame1.Brightness > frame1["Brightness"].median())])
     storeunder1 = len(frame1[(frame1.Brightness < frame1["Brightness"].median())])

     if (storeover1 > storeunder1):
         framebinary.append(1)
     else:
         framebinary.append(0)
```

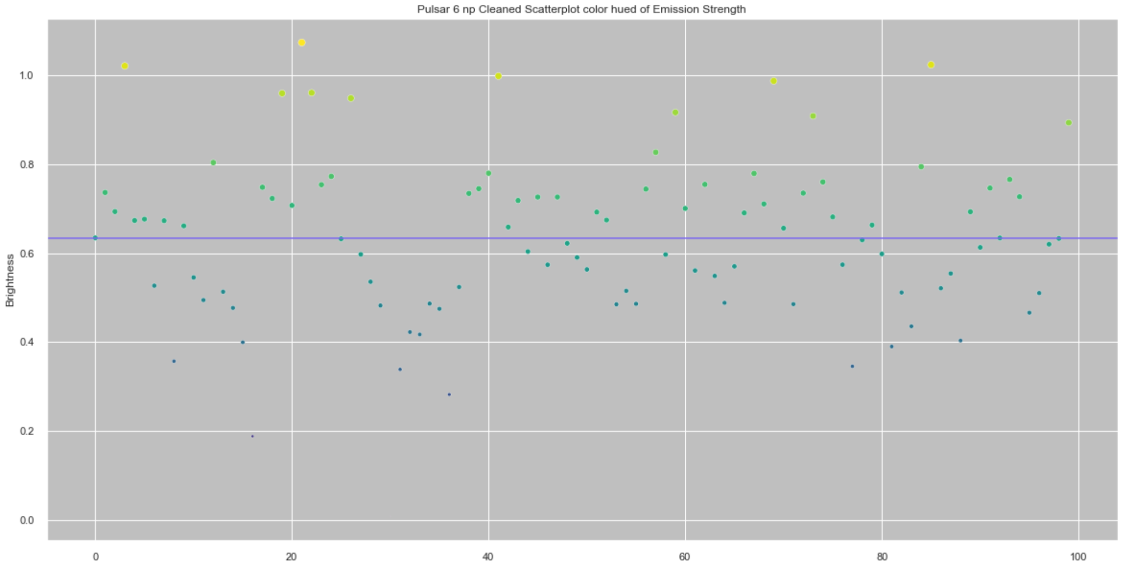
```
    Pulse Number  Brightness  Uncertainty  Binary
0              1    0.634671     0.002761       1
1              2    0.736945     0.005207       1
2              3    0.693834     0.002706       1
3              4    1.021866     0.010184       1
4              5    0.673845     0.006236       1
..           …         …          …      …
95            96    0.466249     0.002850       0
96            97    0.510350     0.003131       0
97            98    0.620342     0.004379       0
98            99    0.633366     0.005906       0
99           100    0.894052     0.008207       1

[100 rows x 4 columns]
```

```
[ ]: plt.figure(figsize=(20,10))
     sns.set_theme(style="whitegrid")
     ax = sns.boxplot(x=frame1["Brightness"]).set_title("Boxplot of Pulsar 6␣
      ↪Emission Strength")
```

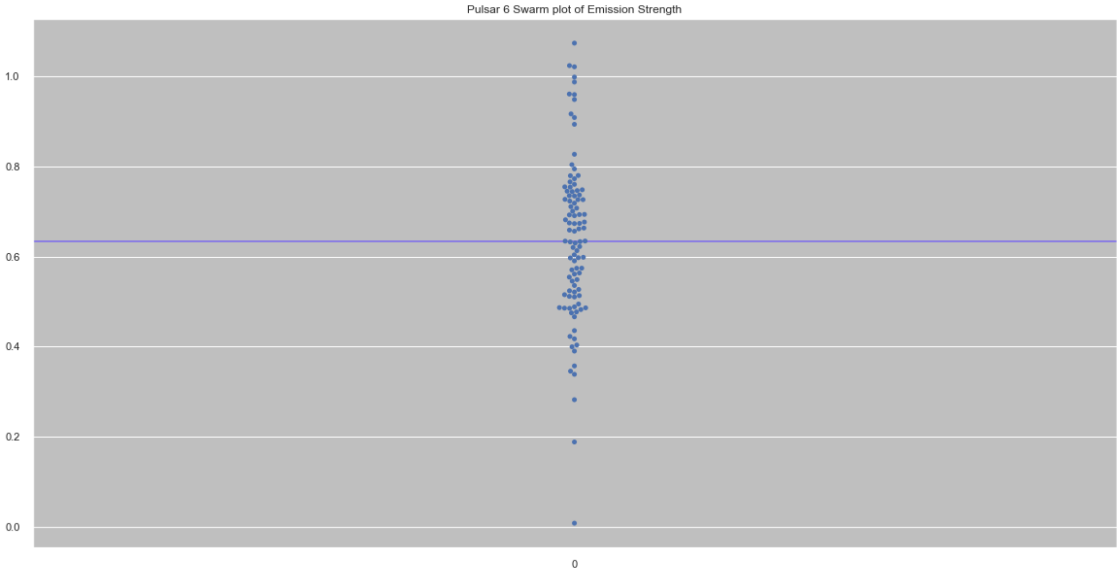


[ ]:

```
plt.figure(figsize=(20,10))
sns.set_style("darkgrid", {"axes.facecolor": ".75"})
strength = frame1.Brightness.values
ax = sns.scatterplot(data=frame1["Brightness"], s= strength*50, c=strength, cmap="viridis", marker="o").set_title('Pulsar 6 np Cleaned Scatterplot color hued of Emission Strength')
ax = plt.axhline( y=0.63457545, ls='-',c='mediumslateblue')
```



[ ]:
```python
plt.figure(figsize=(20,10))
sns.set_style("darkgrid", {"axes.facecolor": ".75"})
strength = frame1.Brightness.values
ax = plt.axhline( y=0.63457545, ls='-',c='mediumslateblue')
ax = sns.swarmplot(data=frame1["Brightness"], c="blue").set_title('Pulsar 6 Swarm plot of Emission Strength')
```

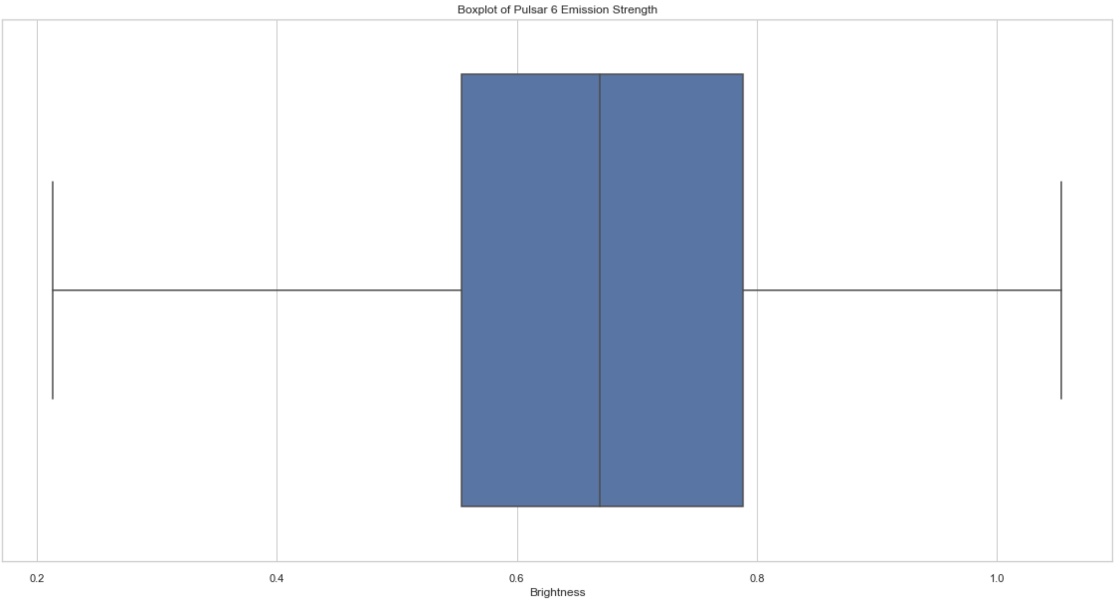


[ ]:
```python
print(frame2)

storeover1 = len(frame2[(frame2.Brightness > frame2["Brightness"].median())])
storeunder1 = len(frame2[(frame2.Brightness < frame2["Brightness"].median())])

if (storeover1 > storeunder1):
    framebinary.append(1)
else:
    framebinary.append(0)
```

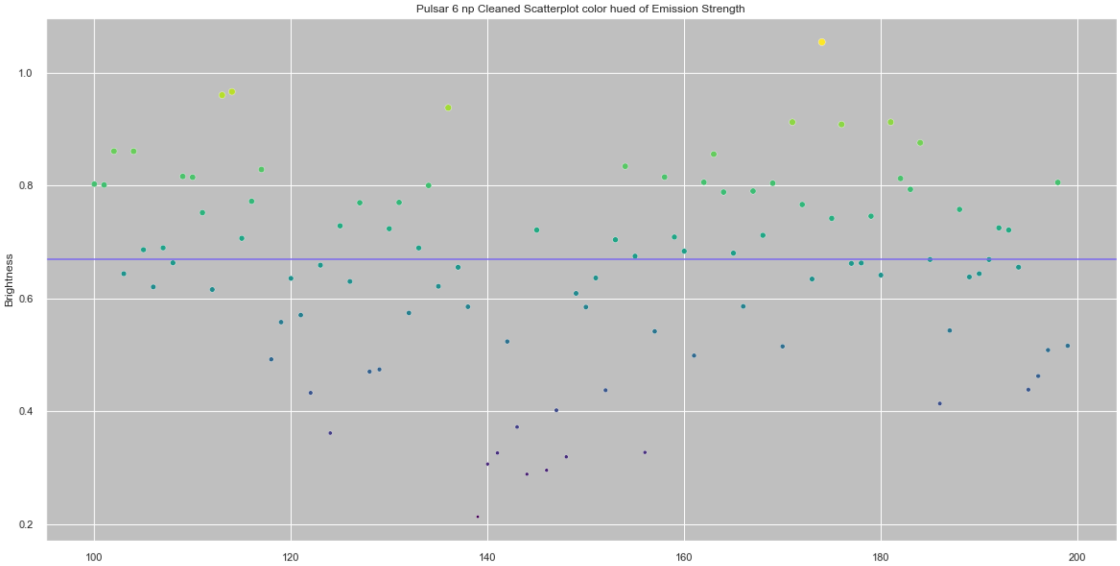
```
     Pulse Number  Brightness  Uncertainty  Binary
100           101    0.802381     0.004107       1
101           102    0.800921     0.002385       1
102           103    0.860724     0.002700       1
103           104    0.643710     0.002618       0
104           105    0.860529     0.002837       1
..            …          …            …        …
195           196    0.438406     0.003504       0
196           197    0.462477     0.003358       0
197           198    0.508498     0.002759       0
198           199    0.805315     0.005269       1
```

```
199           200    0.516107     0.004522       0

[100 rows x 4 columns]
```

```
[ ]: plt.figure(figsize=(20,10))
     sns.set_theme(style="whitegrid")
     ax = sns.boxplot(x=frame2["Brightness"]).set_title("Boxplot of Pulsar 6␣
      ↪Emission Strength")
```



```
[ ]: plt.figure(figsize=(20,10))
     sns.set_style("darkgrid", {"axes.facecolor": ".75"})
     strength = frame2.Brightness.values
     ax = sns.scatterplot(data=frame2["Brightness"], s= strength*50, c=strength,␣
      ↪cmap="viridis", marker="o").set_title('Pulsar 6 np Cleaned Scatterplot color␣
      ↪hued of Emission Strength')
     ax = plt.axhline( y=0.6688056, ls='-',c='mediumslateblue')
```



[ ]:

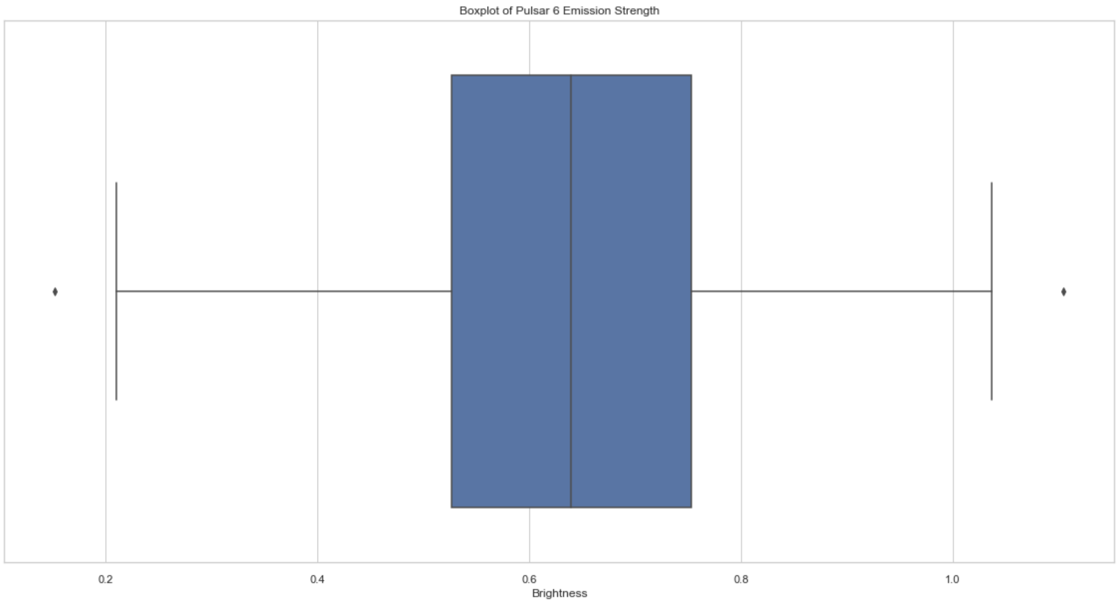
```python
print(frame3)

storeover1 = len(frame3[(frame3.Brightness > frame3["Brightness"].median())])
storeunder1 = len(frame3[(frame3.Brightness < frame3["Brightness"].median())])

if (storeover1 > storeunder1):
    framebinary.append(1)
else:
    framebinary.append(0)
```

```
     Pulse Number  Brightness  Uncertainty  Binary
200           201    0.773417     0.005146       1
201           202    0.905517     0.010704       1
202           203    0.503725     0.004764       0
203           204    0.460606     0.004345       0
204           205    0.413456     0.003170       0
..            …         …           …      …
295           296    0.495127     0.002502       0
296           297    0.763535     0.003293       1
297           298    0.687345     0.002819       1
298           299    0.915965     0.003761       1
299           300    0.739067     0.003667       1

[100 rows x 4 columns]
```

[ ]:

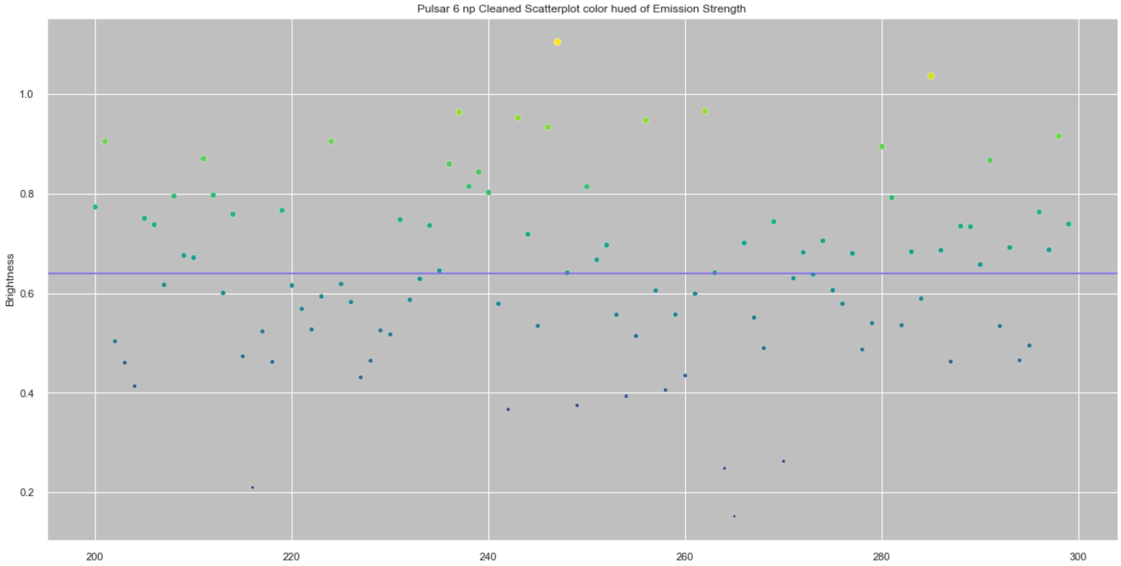
```python
plt.figure(figsize=(20,10))
sns.set_theme(style="whitegrid")
```

```
ax = sns.boxplot(x=frame3["Brightness"]).set_title("Boxplot of Pulsar 6␣
↪Emission Strength")
```



[ ]:

```
plt.figure(figsize=(20,10))
sns.set_style("darkgrid", {"axes.facecolor": ".75"})
strength = frame3.Brightness.values
ax = sns.scatterplot(data=frame3["Brightness"], s= strength*50, c=strength,␣
↪cmap="viridis", marker="o").set_title('Pulsar 6 np Cleaned Scatterplot color␣
↪hued of Emission Strength')
ax = plt.axhline( y=0.63955675, ls='-',c='mediumslateblue')
```



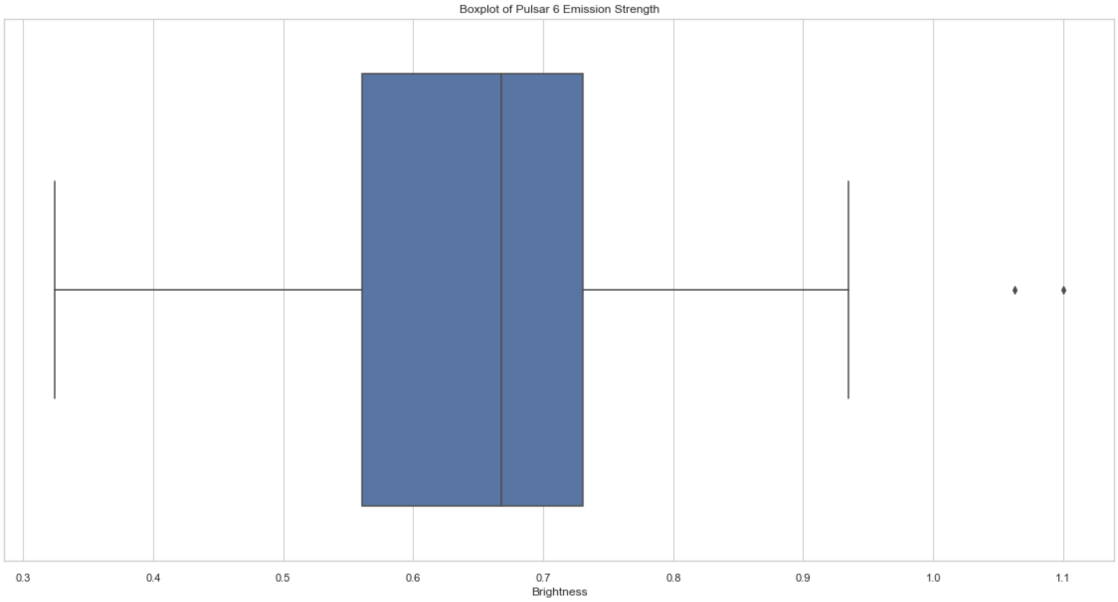
[ ]:

```
print(frame4)

storeover1 = len(frame4[(frame4.Brightness > frame4["Brightness"].median())])
storeunder1 = len(frame4[(frame4.Brightness < frame4["Brightness"].median())])

if (storeover1 > storeunder1):
    framebinary.append(1)
else:
    framebinary.append(0)
```

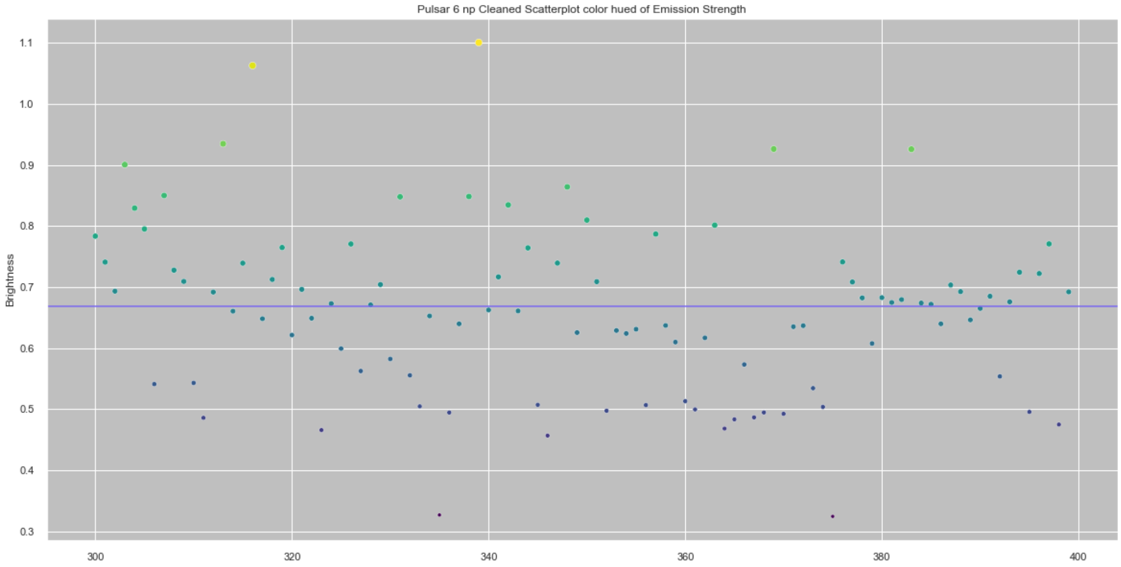
```
     Pulse Number  Brightness  Uncertainty  Binary
300           301    0.783253     0.003680       1
301           302    0.740742     0.007375       1
302           303    0.693110     0.002601       1
303           304    0.900445     0.003646       1
304           305    0.829165     0.002535       1
..            …           …           …      …
395           396    0.495397     0.003248       0
396           397    0.722009     0.008103       1
397           398    0.770565     0.008383       1
398           399    0.474685     0.004108       0
399           400    0.691962     0.004132       1

[100 rows x 4 columns]
```

[ ]:

```
plt.figure(figsize=(20,10))
sns.set_theme(style="whitegrid")
ax = sns.boxplot(x=frame4["Brightness"]).set_title("Boxplot of Pulsar 6␣
 ↪Emission Strength")
```



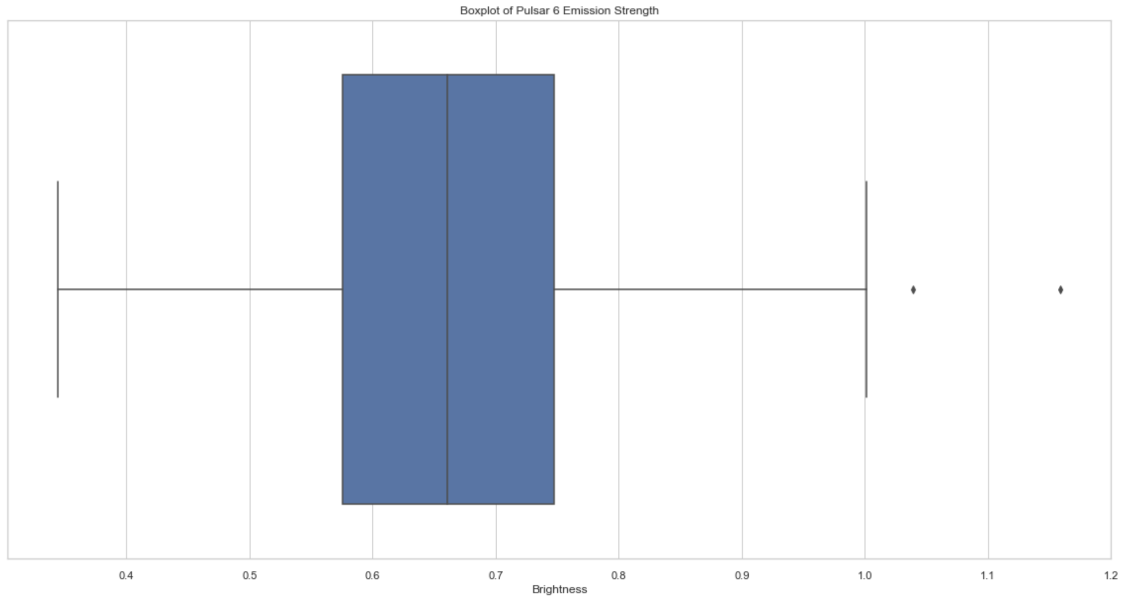
[ ]:
```
plt.figure(figsize=(20,10))
sns.set_style("darkgrid", {"axes.facecolor": ".75"})
strength = frame4.Brightness.values
ax = sns.scatterplot(data=frame4["Brightness"], s= strength*50, c=strength, cmap="viridis", marker="o").set_title('Pulsar 6 np Cleaned Scatterplot color hued of Emission Strength')
ax = plt.axhline( y=0.66777675, ls='-',c='mediumslateblue')
```



```
[ ]: print(frame5)

     storeover1 = len(frame5[(frame5.Brightness > frame5["Brightness"].median())])
     storeunder1 = len(frame5[(frame5.Brightness < frame5["Brightness"].median())])

     if (storeover1 > storeunder1):
         framebinary.append(1)
     else:
         framebinary.append(0)
```

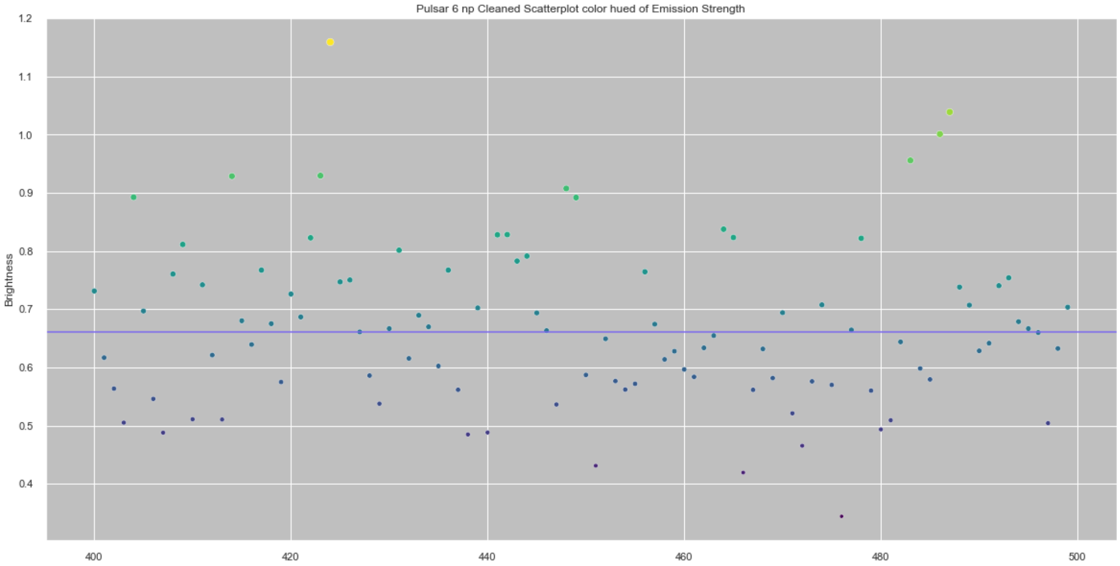
```
     Pulse Number  Brightness  Uncertainty  Binary
400           401    0.731438     0.002577       1
401           402    0.616883     0.002681       0
402           403    0.563571     0.002874       0
403           404    0.505136     0.002388       0
404           405    0.892605     0.007379       1
..            …           …            …     …
495           496    0.666784     0.005140       1
496           497    0.660054     0.005132       0
497           498    0.504216     0.003277       0
498           499    0.632565     0.005493       0
499           500    0.703630     0.005492       1

[100 rows x 4 columns]
```

```
[ ]: plt.figure(figsize=(20,10))
     sns.set_theme(style="whitegrid")
     ax = sns.boxplot(x=frame5["Brightness"]).set_title("Boxplot of Pulsar 6␣
      ↪Emission Strength")
```



[ ]:

```
plt.figure(figsize=(20,10))
sns.set_style("darkgrid", {"axes.facecolor": ".75"})
strength = frame5.Brightness.values
ax = sns.scatterplot(data=frame5["Brightness"], s= strength*50, c=strength,␣
 ↪cmap="viridis", marker="o").set_title('Pulsar 6 np Cleaned Scatterplot color␣
 ↪hued of Emission Strength')
ax = plt.axhline( y=0.6605900499999999, ls='-',c='mediumslateblue')
```



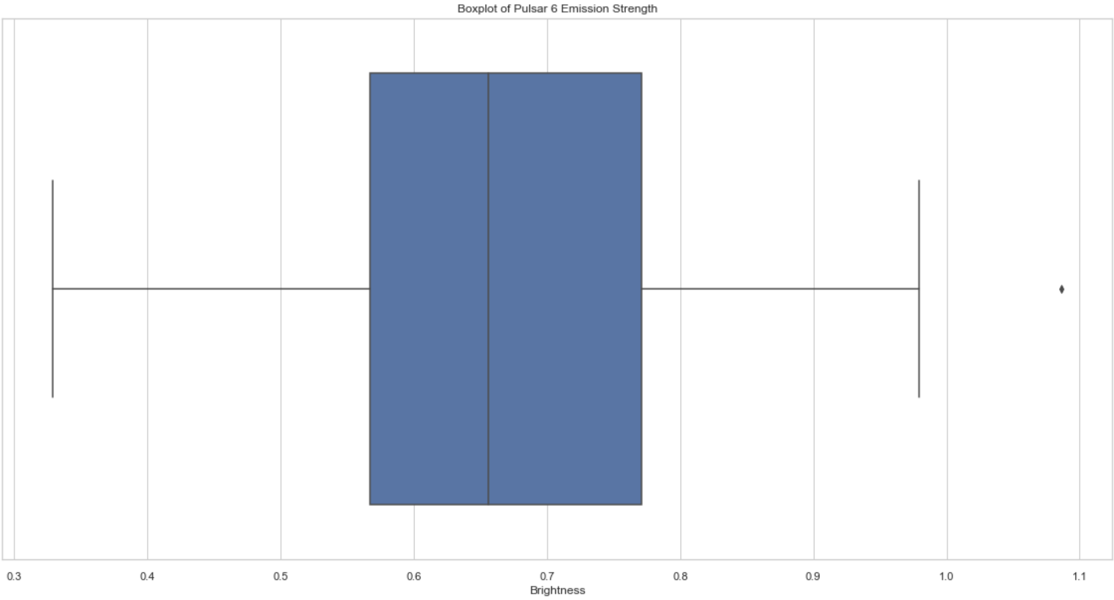
[ ]:

```
print(frame6)

storeover1 = len(frame6[(frame6.Brightness > frame6["Brightness"].median())])
storeunder1 = len(frame6[(frame6.Brightness < frame6["Brightness"].median())])

if (storeover1 > storeunder1):
    framebinary.append(1)
else:
    framebinary.append(0)
```

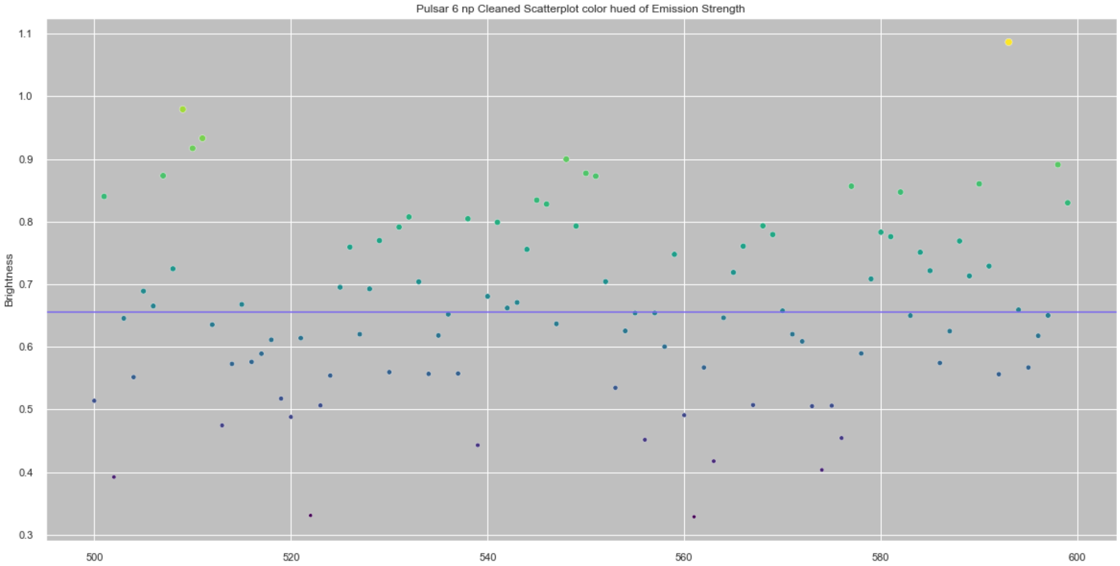
```
     Pulse Number  Brightness  Uncertainty  Binary
500           501    0.513902     0.002946       0
501           502    0.840158     0.003412       1
502           503    0.392136     0.002529       0
503           504    0.645563     0.004307       0
504           505    0.551735     0.003081       0
..            …           …            …       …
595           596    0.567053     0.002552       0
596           597    0.617711     0.003246       0
```

```
597            598    0.650200    0.002647      0
598            599    0.890907    0.004037      1
599            600    0.830011    0.003572      1

[100 rows x 4 columns]
```

```
[ ]: plt.figure(figsize=(20,10))
     sns.set_theme(style="whitegrid")
     ax = sns.boxplot(x=frame6["Brightness"]).set_title("Boxplot of Pulsar 6␣
      ↪Emission Strength")
```



```
[ ]: plt.figure(figsize=(20,10))
     sns.set_style("darkgrid", {"axes.facecolor": ".75"})
     strength = frame6.Brightness.values
     ax = sns.scatterplot(data=frame6["Brightness"], s= strength*50, c=strength,␣
      ↪cmap="viridis", marker="o").set_title('Pulsar 6 np Cleaned Scatterplot color␣
      ↪hued of Emission Strength')
     ax = plt.axhline( y=0.65585835, ls='-',c='mediumslateblue')
```

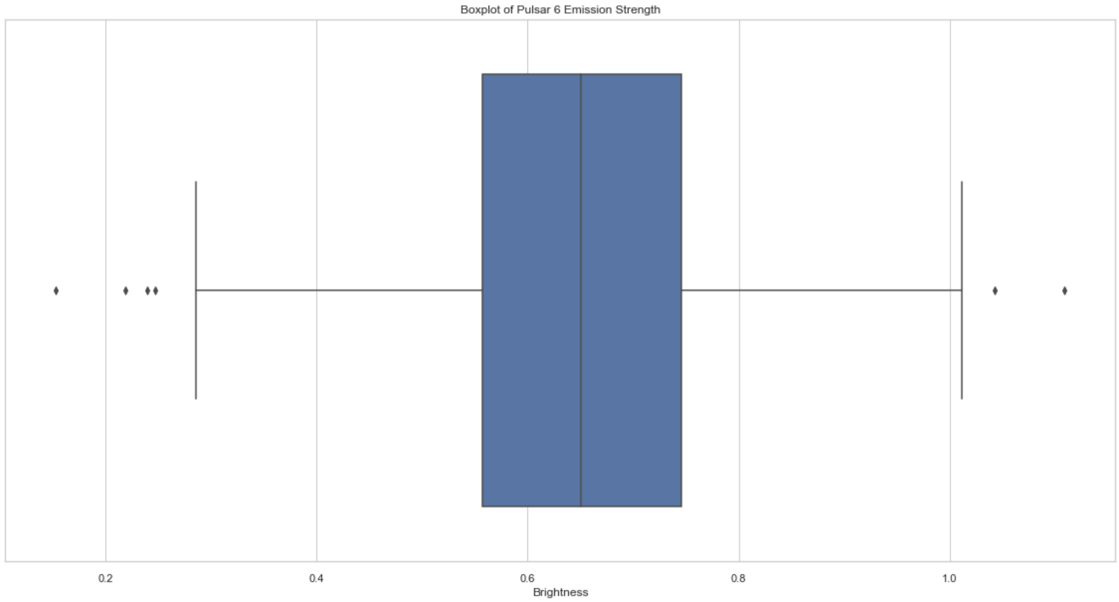


```
[ ]: print(frame7)

     storeover1 = len(frame7[(frame7.Brightness > frame7["Brightness"].median())])
     storeunder1 = len(frame7[(frame7.Brightness < frame7["Brightness"].median())])

     if (storeover1 > storeunder1):
         framebinary.append(1)
     else:
         framebinary.append(0)
```

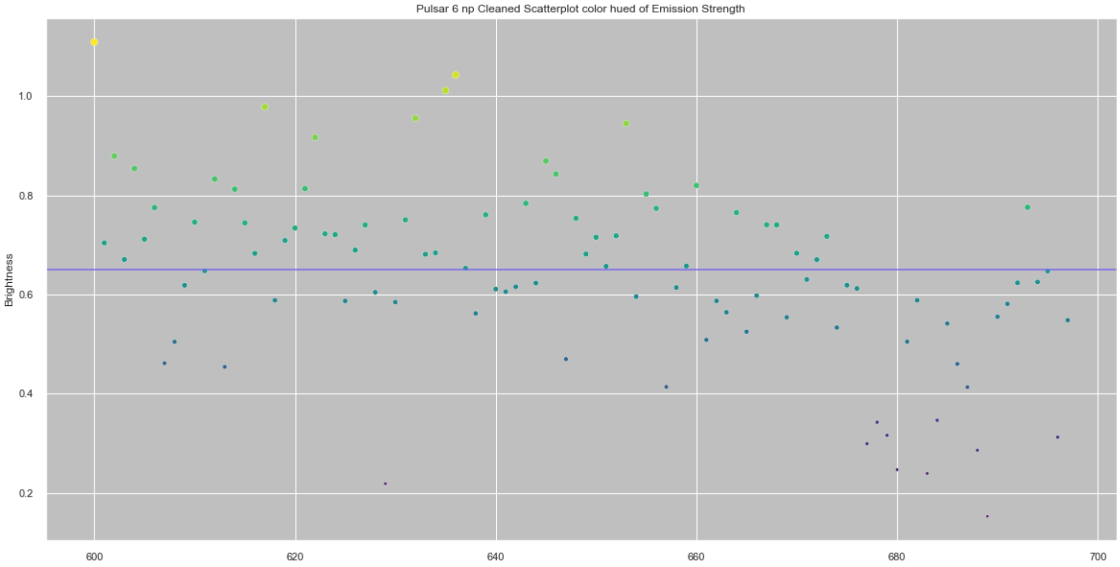
```
     Pulse Number  Brightness  Uncertainty  Binary
600           601    1.109122     0.003188       1
601           602    0.704272     0.002793       1
602           603    0.879200     0.003600       1
603           604    0.670774     0.002567       1
604           605    0.854064     0.005940       1
..            ...         ...          ...     ...
693           694    0.776083     0.008928       1
694           695    0.625382     0.006018       0
695           696    0.647559     0.003765       0
696           697    0.312449     0.002901       0
697           698    0.548353     0.009056       0

[98 rows x 4 columns]
```

```
[ ]: plt.figure(figsize=(20,10))
     sns.set_theme(style="whitegrid")
```

```
ax = sns.boxplot(x=frame7["Brightness"]).set_title("Boxplot of Pulsar 6␣
↪Emission Strength")
```



```
[ ]: plt.figure(figsize=(20,10))
     sns.set_style("darkgrid", {"axes.facecolor": ".75"})
     strength = frame7.Brightness.values
     ax = sns.scatterplot(data=frame7["Brightness"], s= strength*50, c=strength,␣
     ↪cmap="viridis", marker="o").set_title('Pulsar 6 np Cleaned Scatterplot color␣
     ↪hued of Emission Strength')
     ax = plt.axhline( y=0.6504474499999999, ls='-',c='mediumslateblue')
```



[ ]:

```
framebinary
#this didn't go to plan.
```

[ ]:

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

Randomness testing

[ ]:

```
```

[ ]:

```
```

[ ]:

```
```