On the Period-Luminosity Relationship of Cepheid Variables

The demo data set for this part is the Wesenheit index of the OGLE-III fundamental-mode and first overtone classical Cepheids.

The Wesenheit index is defined as `W = I - 1.55(V - I)`, and its main advantage over using simply the I or V photometry is that it is insensitive to extinction. It is denoted by 'W' among the data columns.

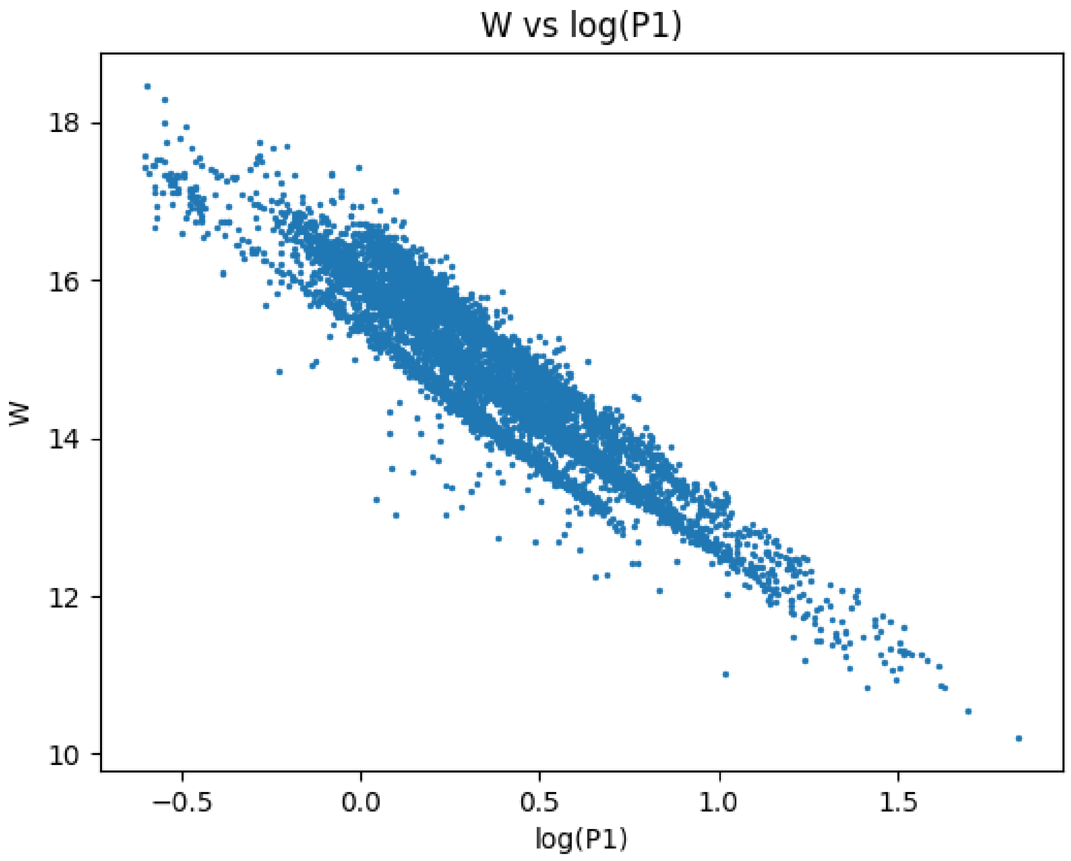
Other columns are 'name', the identifier of the star; 'RA0' (in decimal hours) and 'Decl0' (in decimal degrees), celestial coordinates; 'Mode', the mode of the Cepheid ('F' indicates fundamental-mode, '1' indicates first overtone star); 'Cloud', indicating which Magellanic Cloud the star belongs to; 'logP1', the base-10 logarithm of the period in days; 'VI', the colour V-I.

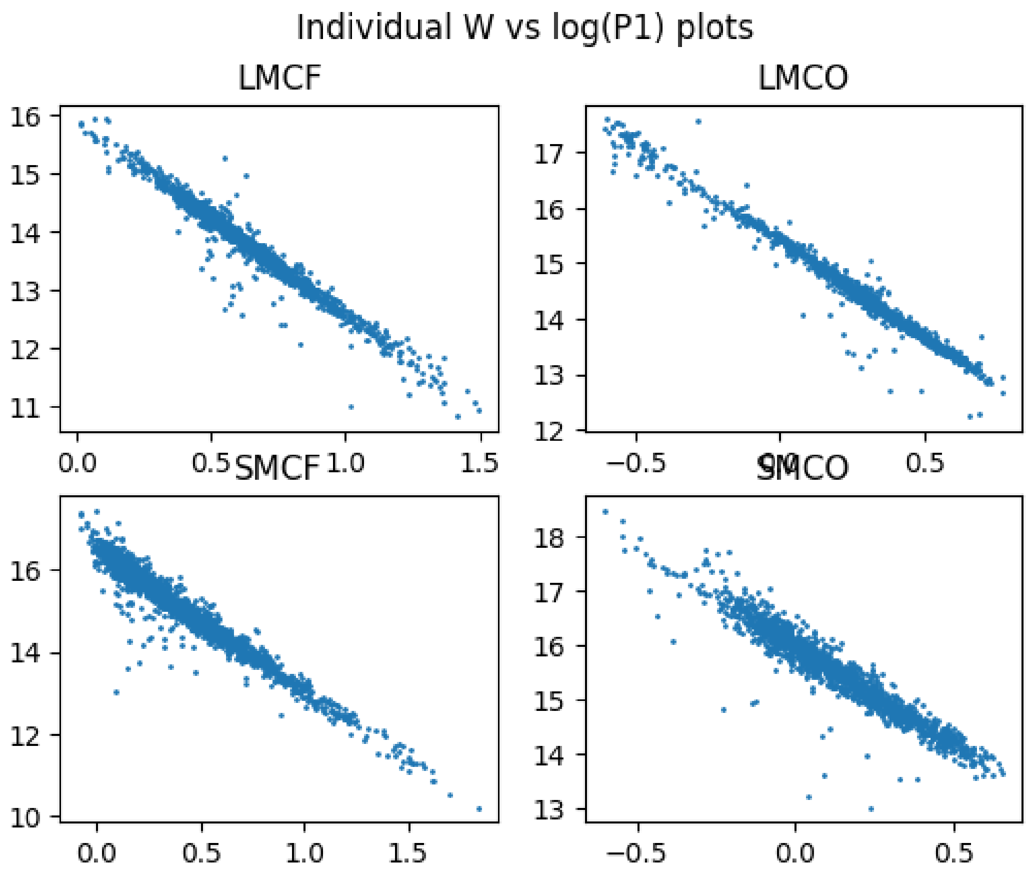
```
# Read in the csv data file using Pandas
import pandas as pd
cep =
pd.read_csv("/home/ambica/Downloads/application-assignment/Cepheids.cs
v")
print(cep.columns)

Index(['name', 'RA0', 'Decl0', 'Mode', 'Cloud', 'W', 'logP1', 'VI'],
dtype='object')

#select Cepheids belonging to the LMC and SMC clouds with F and 1
modes
lmcf = cep[(cep['Cloud'] == 'LMC') & (cep['Mode'] == 'F')]
smcf = cep[(cep['Cloud'] == 'SMC') & (cep['Mode'] == 'F')]

lmco = cep[(cep['Cloud'] == 'LMC') & (cep['Mode'] == '1')]
smco = cep[(cep['Cloud'] == 'SMC') & (cep['Mode'] == '1')]
```

Plot of the `W` on the y-axis vs `log(P1)` on x.

```
import matplotlib.pyplot as plt

#Plot W vs log(P1) for the entire dataset
plt.scatter(cep['logP1'],cep['W'],s=2)
plt.xlabel('log(P1)')
plt.ylabel('W')
plt.title('W vs log(P1)')
plt.show()

#Store dataframes in an array
df=[lmcf,lmco,smcf,smco]
dfname=['LMCF','LMCO','SMCF','SMCO']

#Create subplots
```

```
for i in range(4):
    plt.subplot(2,2,i+1)
    plt.scatter(df[i]['logP1'],df[i]['W'],s=1)
    plt.title(dfname[i])

plt.suptitle('Individual W vs log(P1) plots')
plt.show()
```





Fit or estimating straight lines to each of the four samples

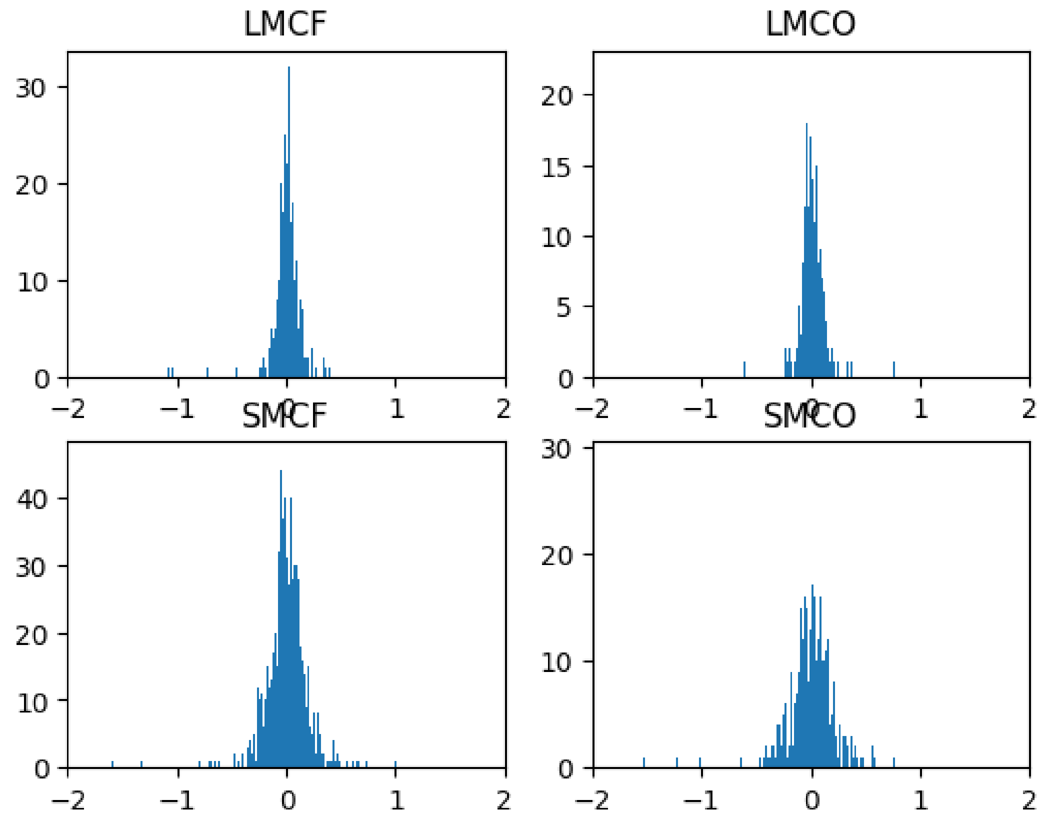
```python
from scipy.optimize import curve_fit

#The straight line to estimate the plots
def func(x,a,b):
    return a*x+b

#Return parameters a,b in the function
params=[['0' for x in range(2)] for x in range(4)]
covs=[['0' for x in range(2)] for x in range(4)]

for i in range(4):
    params[i],covs[i]=curve_fit(func,df[i]['logP1'],df[i]['W'])

print(params)
```

```
[array([-3.32588476, 15.89287088]), array([-3.43224338, 15.38395972]),
array([-3.45364502, 16.48043801]), array([-3.61548559, 15.96236752])]
```

Compute the residuals of each sample to its respective line.

The residuals appear to be normally distributed centred around zero meaning that the regression model as given by the elements of `params` is close to reality, and a linear relationship between the Wesenheit Index and the logarithm of period does exist in actuality.

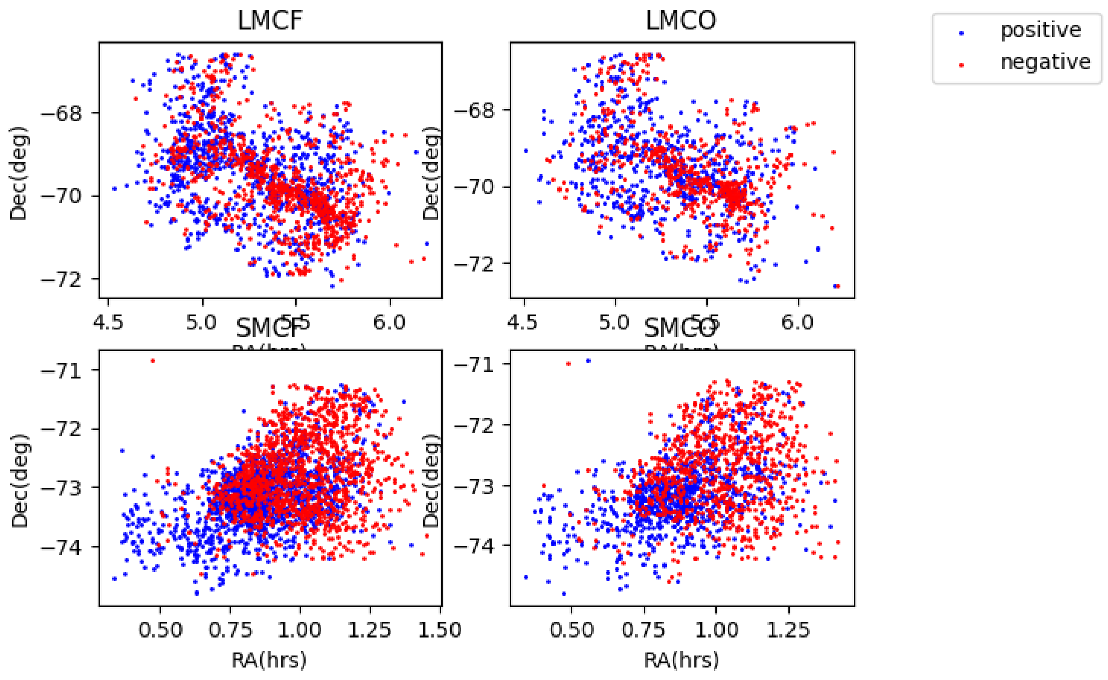
```
#Calculate residuals for each of the four slices
for i in range(4):
    df[i] = df[i].copy()
    df[i]['Prediction'] = func(df[i]['logP1'], params[i][0], params[i]
[1])
    df[i]['Residuals'] = df[i]['W'] - df[i]['Prediction']

    plt.subplot(2, 2, i+1)
    plt.hist(df[i]['Residuals'], bins=1000)
    plt.xlim(-2, 2)
    plt.title(dfname[i])
```



Scatter plot of the residuals as RA (x-axis) vs Dec (y-axis)

From the nature of the scatterplots one can infer that the number of Cepheid Variables with positive and negative deviation are nearly equal. In the small Magellanic Cloud there appears to be a correlation b/w the location and the nature of the deviation; positive deviations are mostly at low RA,Dec and negative deviations are more at high RA, Dec. Distribution is more even for the LMC.

```
for i in range(4):
    plt.subplot(2,2,i+1)
    #Two dataframes for positive and negative residuals
    dfp=df[i][(df[i]['Residuals']>0)]
    dfn=df[i][(df[i]['Residuals']<0)]


plt.scatter(dfp['RA0'],dfp['Decl0'],color='blue',s=1,label='positive')

plt.scatter(dfn['RA0'],dfn['Decl0'],color='red',s=1,label='negative')
    plt.xlabel('RA(hrs)')
    plt.ylabel('Dec(deg)')
    plt.title(dfname[i])
plt.legend(bbox_to_anchor=(1.75,2),loc='lower right')
plt.show()
```


LMCF
LMCO
SMCF
SMCO
Dec(deg)
RA(hrs)
positive
negative