eight-dsa-1

April 19, 2024

1 Install the Necessary Libraries

Name : Pratik Yuvraj Yawalkar

Roll No. : AI23MTECH11006

Department : AI & ML

[12]:
```
!pip install astroml numpy pandas scipy matplotlib seaborn corner emcee pymc3␣
↪dynesty
import warnings
warnings.filterwarnings('ignore')
```

```
Requirement already satisfied: astroml in /usr/local/lib/python3.10/dist-
packages (1.0.2.post1)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
(1.22.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages
(2.0.3)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages
(1.7.3)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-
packages (3.7.1)
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-
packages (0.13.1)
Requirement already satisfied: corner in /usr/local/lib/python3.10/dist-packages
(2.2.2)
Requirement already satisfied: emcee in /usr/local/lib/python3.10/dist-packages
(3.1.6)
Requirement already satisfied: pymc3 in /usr/local/lib/python3.10/dist-packages
(3.11.5)
Requirement already satisfied: dynesty in /usr/local/lib/python3.10/dist-
packages (2.1.3)
Requirement already satisfied: scikit-learn>=0.18 in
/usr/local/lib/python3.10/dist-packages (from astroml) (1.2.2)
Requirement already satisfied: astropy>=3.0 in /usr/local/lib/python3.10/dist-
packages (from astroml) (5.3.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
```

```
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-
packages (from pandas) (2024.1)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (24.0)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-
packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.2)
Requirement already satisfied: arviz>=0.11.0 in /usr/local/lib/python3.10/dist-
packages (from pymc3) (0.12.1)
Requirement already satisfied: cachetools>=4.2.1 in
/usr/local/lib/python3.10/dist-packages (from pymc3) (5.3.3)
Requirement already satisfied: deprecat in /usr/local/lib/python3.10/dist-
packages (from pymc3) (2.1.1)
Requirement already satisfied: dill in /usr/local/lib/python3.10/dist-packages
(from pymc3) (0.3.8)
Requirement already satisfied: fastprogress>=0.2.0 in
/usr/local/lib/python3.10/dist-packages (from pymc3) (1.0.3)
Requirement already satisfied: patsy>=0.5.1 in /usr/local/lib/python3.10/dist-
packages (from pymc3) (0.5.6)
Requirement already satisfied: semver>=2.13.0 in /usr/local/lib/python3.10/dist-
packages (from pymc3) (3.0.2)
Requirement already satisfied: theano-pymc==1.1.2 in
/usr/local/lib/python3.10/dist-packages (from pymc3) (1.1.2)
Requirement already satisfied: typing-extensions>=3.7.4 in
/usr/local/lib/python3.10/dist-packages (from pymc3) (4.11.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from theano-pymc==1.1.2->pymc3) (3.13.4)
Requirement already satisfied: setuptools>=38.4 in
/usr/local/lib/python3.10/dist-packages (from arviz>=0.11.0->pymc3) (67.7.2)
Requirement already satisfied: xarray>=0.16.1 in /usr/local/lib/python3.10/dist-
packages (from arviz>=0.11.0->pymc3) (2023.7.0)
Requirement already satisfied: netcdf4 in /usr/local/lib/python3.10/dist-
packages (from arviz>=0.11.0->pymc3) (1.6.5)
Requirement already satisfied: xarray-einstats>=0.2 in
/usr/local/lib/python3.10/dist-packages (from arviz>=0.11.0->pymc3) (0.6.0)
Requirement already satisfied: pyerfa>=2.0 in /usr/local/lib/python3.10/dist-
packages (from astropy>=3.0->astroml) (2.0.1.4)
```

```
Requirement already satisfied: PyYAML>=3.13 in /usr/local/lib/python3.10/dist-
packages (from astropy>=3.0->astroml) (6.0.1)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages
(from patsy>=0.5.1->pymc3) (1.16.0)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn>=0.18->astroml) (1.4.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.18->astroml)
(3.4.0)
Requirement already satisfied: wrapt<2,>=1.10 in /usr/local/lib/python3.10/dist-
packages (from deprecat->pymc3) (1.14.1)
Requirement already satisfied: cftime in /usr/local/lib/python3.10/dist-packages
(from netcdf4->arviz>=0.11.0->pymc3) (1.6.3)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-
packages (from netcdf4->arviz>=0.11.0->pymc3) (2024.2.2)
```

2 1. Calculate the angular two-point correlation function of galaxies (including errors obtained by 10 bootstrap resamples) using subset of data from the Blanco Cosmology Survey with r-band magnitude between 17 and 20, and using 16 logarithmic-spaced angular bins from 1/60 ∘ to 1 ∘ . Use a linear scale for Y-axis. Galaxies in Blanco Cosmology Survey have spread model > 0.002. This data can be downloaded from http://www.iith.ac.in/~shantanud/BCS05hr_reduced.txt (30 points) (Hint : Look at the astroML source code for Figure 6.17)

[13]:
```
import numpy as np
import matplotlib.pyplot as plt

# Function to calculate the angular distance between two points on the sky
def angular_distance(ra1, dec1, ra2, dec2):
    delta_ra = ra2 - ra1
    delta_dec = dec2 - dec1
    return np.sqrt(delta_ra**2 + delta_dec**2)

# Read data from the text file
data = np.loadtxt('BCS05hr_reduced.txt')  # Replace 'your_data_file.txt' with␣
 ↪the actual file name
```

[14]:
```
# Extract relevant columns from the data
ra_data = data[:, 0]
dec_data = data[:, 1]
mag_data = data[:, 2]
```

```
spread_model = data[:, 3]
```

[15]:
```
# Define data subset: galaxies with r-band magnitude between 17 and 20 and␣
↪spread model > 0.002
selected_galaxies = (mag_data >= 17) & (mag_data <= 20) & (spread_model > 0.002)
ra_selected = ra_data[selected_galaxies]
dec_selected = dec_data[selected_galaxies]
```

[16]:
```
# Define angular bins
num_bins = 16
angular_bins = np.logspace(np.log10(1/60), 0, num=num_bins)
```

[17]:
```
# Calculate pair counts in each angular bin
pair_counts = np.zeros(num_bins - 1)
for i in range(len(ra_selected)):
    for j in range(i + 1, len(ra_selected)):
        distance = angular_distance(ra_selected[i], dec_selected[i],␣
↪ra_selected[j], dec_selected[j])
        bin_index = np.searchsorted(angular_bins, distance, side='right')
        if bin_index < num_bins:
            pair_counts[bin_index - 1] += 1
```

[18]:
```
# Calculate expected pair counts for a random distribution
area = 4 * np.pi * (180 / np.pi)**2  # Total area of the sky in square degrees
mean_density = len(ra_selected) / area
expected_counts = mean_density * (angular_bins[1:]**2 - angular_bins[:-1]**2)
```

[19]:
```
# Calculate correlation function
correlation_function = pair_counts / expected_counts - 1
```

[9]:
```
# Bootstrap resampling for error estimation
num_resamples = 10
bootstrap_correlation_functions = np.zeros((num_resamples,␣
↪len(correlation_function)))
for i in range(num_resamples):
    resample_indices = np.random.choice(len(ra_selected),␣
↪size=len(ra_selected), replace=True)
    resample_pair_counts = np.zeros(num_bins - 1)
    for j in range(len(resample_indices)):
        for k in range(j + 1, len(resample_indices)):
            distance = angular_distance(ra_selected[resample_indices[j]],␣
↪dec_selected[resample_indices[j]],
                                       ra_selected[resample_indices[k]],␣
↪dec_selected[resample_indices[k]])
            bin_index = np.searchsorted(angular_bins, distance, side='right')
            if bin_index < num_bins:
```

```
                resample_pair_counts[bin_index - 1] += 1
    bootstrap_correlation_functions[i] = resample_pair_counts / expected_counts␣
 ↪- 1
```

[10]:
```
# Calculate bootstrap errors
bootstrap_errors = np.std(bootstrap_correlation_functions, axis=0)
```

[21]:
```
# Plot correlation function with errors
plt.errorbar(angular_bins[:-1], correlation_function, yerr=bootstrap_errors,␣
 ↪fmt='xg')
plt.xlabel('Angular Separation (degrees) on x-axis')
plt.ylabel('Correlation Function on y-axis')
plt.xscale('log')
plt.yscale('linear')
plt.title('Angular Two-Point Correlation Function Plot')
plt.show()
```