TWO POINT ANGULAR CORRELATION FUNCTION FOR AN ASTROPHYSICAL OBJECTS TAKEN WITH SEXTRACTOR

ZAKARIA ABDELRAHMAN[1]
ZERGUINE SOHAIB[2]

20/02/2021

CONTENTS

ABSTRACT

During our stay on the **O**bservatoire **H**aute de **P**rovence, we took a lesson about the usability of computer programming in object identification and distinguishing objects. We used the famous program coded in **C**, **S.E**xtractor as a direct example. Next, we tried to write our own code by ourselves through which we based on a linear separation function, with two essential parameters(α, β); such that the classification satisfies a linear type of separation $f(x) = \alpha x + \beta$, each type yelds to a well defined range of each parameter. Then we will use the **A**ngular **T**wo **P**oint **C**orrelation **F**unction technic, to extract the corresponding Angular distribution of each type and the related specifications.

Guided by **Drs. Cristophe ADAMI & Delphine RUSSEIL**

[1,2] *Department of Physics, Aix–Marseille University, Marseille, France*

1 INTRODUCTION

The object was picked out from the frame using **S.E**xtractor, we succeeded to extract 258 objects between stars and galaxies. S.Extractor is an open source powerful program built in **C** that constructs a catalogue based on several options of the astronomical image. Essentially, it contains an effective data reduction strategies in background substraction. It uses different data refining methods as $\sigma-$ Clipping, as it can perform a complicated background map from a superposition of Background maps for every object. It can give us cleaned data from cosmic showers as well.

The Frame from which we extracted objects, the image is already scientific

A scientific image gives us the most trustful data, it takes on consideration the rejection of noise, and the detection unit 'helth' situation (Normalization via the flat field).

$$\text{Image}_{(\text{Scientific})} = \mathcal{F}(\text{Image}_{(\text{Raw})}) = \frac{\text{Image}_{(\text{Raw})} - \text{Image}_{(\text{Dark})}}{\text{Image}_{(\text{Flat\_Field})}}$$

We set the S.Extractor to give us only the quantities which interests us in the analysis of data, it had given us the objects ordered from left to right and from up to down with:

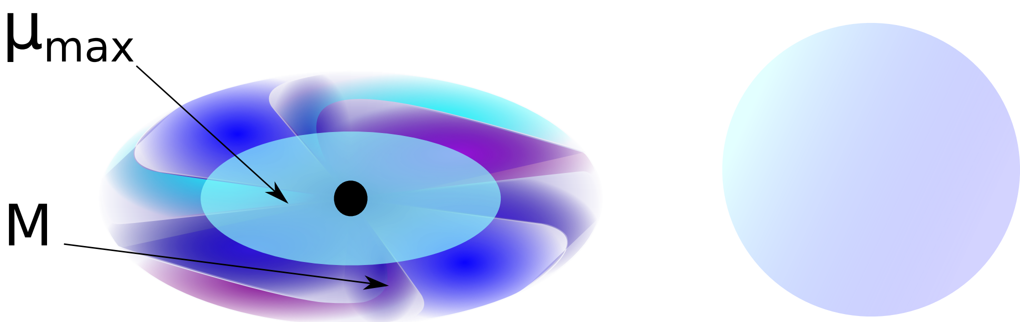
$$\| \text{ \# Object } \| \text{ (X-position) } \| \text{ (Y-position) } \| \ \mu_{max} \ \| \ M \ \|$$

2 SEPARATION

When taking an astrophysical picture on a telescope, we can extract many types of informations, among them we can identify the most bright spots rather then the magnitudes of any illuminating objects. If the image of the Object is quite big, then it almost refers to a galaxy. In galaxies we have two regions, a weak illuminating region on the edges, and a high dense and then illuminating region in the center. the ratio of the Brightness of this central spot μ_{max} over the overall magnitude **M** gives us the parameter α.

Galaxies are the most difficult objects to be detected, they look as stars if they're too far, so in order to resolve them we need to refine the data in a non conventional way.

In our work, we tried to classify these objects in two categories, (stars, galaxies) according to two parameters (linear classification), they are defined as in below:



Galaxy's parameter $\alpha \geqslant 1$

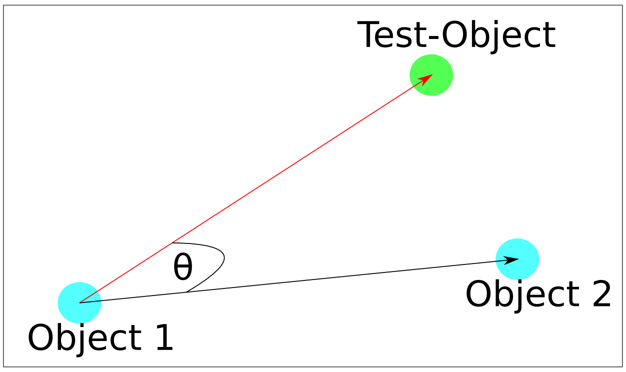
Stars parameter $\alpha \approx 1$

$$\text{Object}_{\{Type\}} = \overbrace{\left[\frac{\text{Brightness}}{\text{Magnitude}}\right]}^{\alpha} \times \text{Object}_{\{Unknown\}} + \underbrace{\text{Brightness}_{\{\text{Cutoff}\}}}_{\beta}$$

1. if $\beta \in]\beta_0, \infty[$ then the object is a star.
2. if $\alpha \in [0, \alpha_0]$ then the object is confirmed to be a star.
3. else; the object is a galaxy.

3 TWO POINT ANGULAR CORRELATION FUNCTION

T.P.A.C.F denoted as $\omega(\theta)$ is a statistical estimator, allowing us to unveil hidden statistical characteristics, The quality of a statistical estimation relies essentially on the convergence of the estimator.



$\omega(\theta)$ is the density of probability for a third Test-Object to be in a direction held on an angle of θ.

Let's consider our data **D** the base of our estimations, we want to create a random data **R**

$$\omega(\theta) = \mathcal{A}(\mathbf{D}, \mathbf{R}) = \frac{(\mathbf{D_i} - \mathbf{R_i})(\mathbf{D^j} - \mathbf{R^j})}{|DR|} = \frac{D_i D^j + R_i R^j - 2 \times D_i R^j}{D_i R^j}$$

Such that the spacetime metric is flat: $\eta^i_j = \delta^i_j = \mathbb{1}_4$. And The distances between the same points is null $D_i D^i = R_i R^i = 0$ DD is the scalar product of two objects "distance $\times$ angle", RR is defined betwen two random virtual data and DR is defined betwen real data and random virtual data.

We take the number of a random data same as the number of data in every case, total data(258). After these heavy calculations on the Laptop, we managed to plot a histogram, that describes a distribution of the θ's. the corresponding function is therefore $\omega(\theta)$.

Important: In order to compare the results "mean angle and its corresponding error" between stars and Galaxies, we need to set the Random Virtual data points to be proportional to the number of separated data. i.e:
31 Galaxies $\longrightarrow$ 31 Random Points.
227 Stars $\longrightarrow$ 227 Random Points.

4 OUR CODE

4.0.1 The Libraries we need:

[4]:
```
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (16,8)
```

4.0.2 Data segmentation

[2]:
```
with open("data.txt",'r') as d:
    N = []
    X = []
    Y = []
    MUMAX = []
    MAG = []
##########################################
    for line in d:
        l = line.split()
        n,x,y,mumax,mag = int(l[0]),float(l[1]),float(l[2]),float(l[3]),float(l[4])
###Object_index:
        N.append(n)
###Object_X_Position:
        X.append(x)
###Object_Y_Position:
        Y.append(y)
###Object_MU_MAX:
        MUMAX.append(mumax)
###Object_Magnitude:
        MAG.append(mag)
##########################################
```

[3]:
```
NP = len(N)
NP
```

[3]: 258

4.0.3 Galaxy/Stars distinction

4.0.4 Brightness Cut_off

[4]:
```
Cut_off = []
Ratio = []
ss=0
c1=0
c2=0
c3=0
c4=0
for i1 in MUMAX:
    ss+=1
    if (i1 > (-13)):
        Cut_off.append(ss)
```

4.0.5 Brightness/Magnitude Criterion

[5]:
```
for k in MUMAX:
    c2+=1
    c3=0
    for j in MAG:
        c3+=1
        if ((c2 == c3) and (c3 in Cut_off)):
            if ((k/j) < 0.94):
                Ratio.append(c3)
```

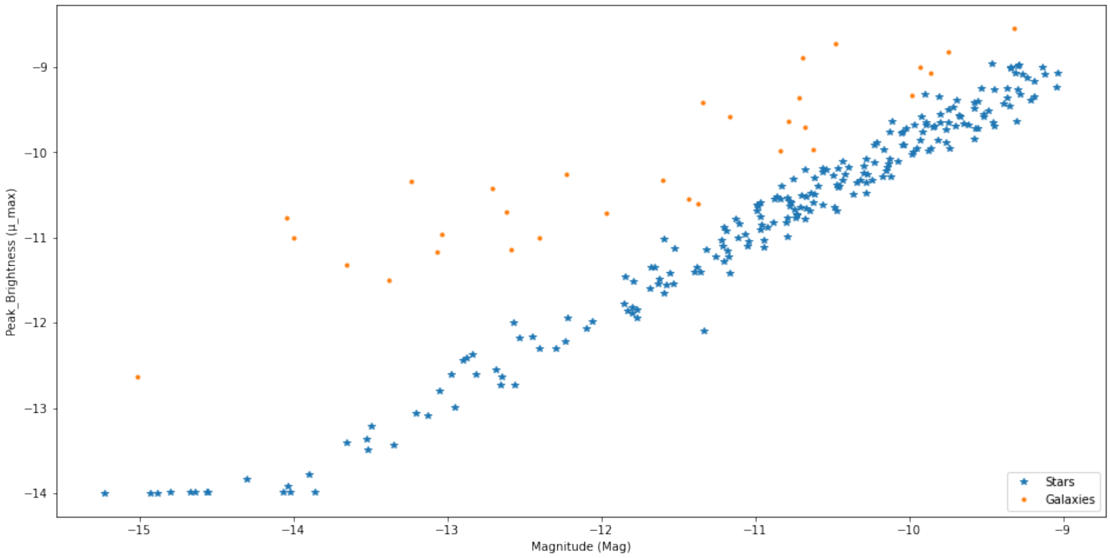
[6]:
```
Star_MUMAX=[]
Star_MAG=[]
Glag_MUMAX=[]
Glag_MAG=[]
XS=[]
XG=[]
YS=[]
YG=[]
for i3 in N:
    if i3 in Ratio:
        Glag_MUMAX.append(MUMAX[i3-1])
        Glag_MAANgulartwopointcorrelationfunction.pyG.append(MAG[i3-1])
        XG.append(X[i3-1])
        YG.append(Y[i3-1])
    else:
        Star_MUMAX.append(MUMAX[i3-1])
        Star_MAG.append(MAG[i3-1])
        XS.append(X[i3-1])
        YS.append(Y[i3-1])
NPS = len(Star_MUMAX)
NPG = len(Glag_MUMAX)
```

[7]:
```
Galaxy_Signals_Ratio = (NPG/NP)
print('Galaxy_Signals_Ratio:',Galaxy_Signals_Ratio)
Star_Signals_Ratio = (NPS/NP)
print('Star_Signals_Ratio:',Star_Signals_Ratio)
```

```
Galaxy_Signals_Ratio: 0.12015503875968993
Star_Signals_Ratio: 0.8798449612403101
```

4.0.6 Visualization

[8]:
```
plt.plot(Star_MAG,Star_MUMAX,'*',label='Stars')
plt.plot(Glag_MAG,Glag_MUMAX,'.',label='Galaxies')
plt.xlabel('Magnitude (Mag)')
plt.ylabel('Peak_Brightness (μ_max)')
plt.legend(loc='lower right')
plt.draw()
plt.savefig('object_separation.pdf')
```

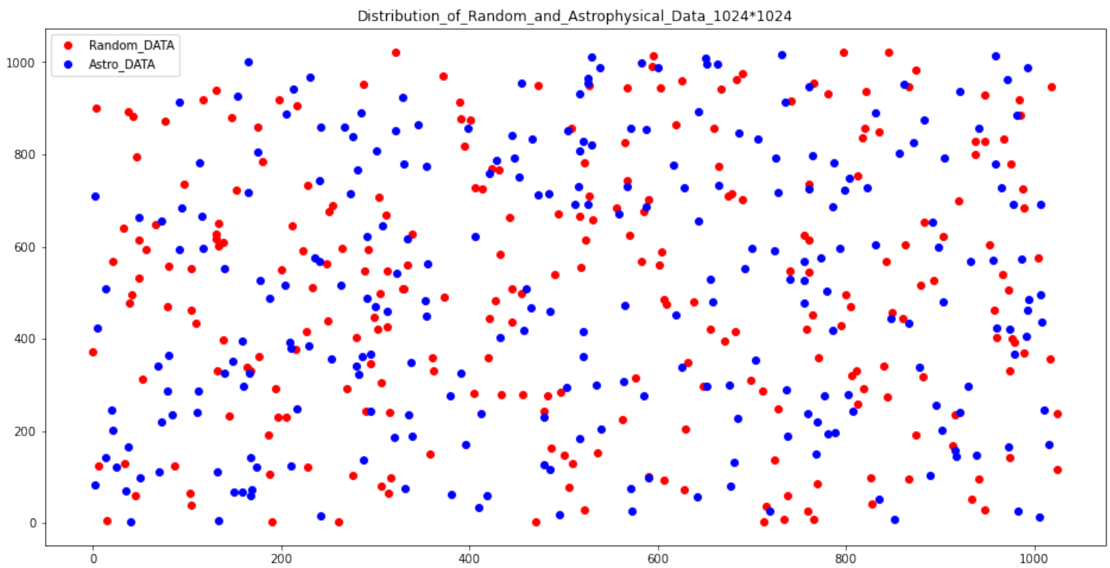


4.0.7 *Creation of Random Data On the same frame with The Astrophysical Data*

4.0.8 *All_of_Data*

[9]:
```python
X1 = np.random.uniform(low=0, high=1024, size=(NP,))
Y1 = np.random.uniform(low=0, high=1024, size=(NP,))
X2 = X
Y2 = Y
```

[10]:
```python
plt.scatter(X1,Y1,c='r',label='Random_DATA')
plt.scatter(X2,Y2,c='b',label='Astro_DATA')
plt.title("Distribution_of_Random_and_Astrophysical_Data_1024*1024")
plt.legend()
plt.draw()
plt.savefig('1024*1024_visualization.pdf')
```

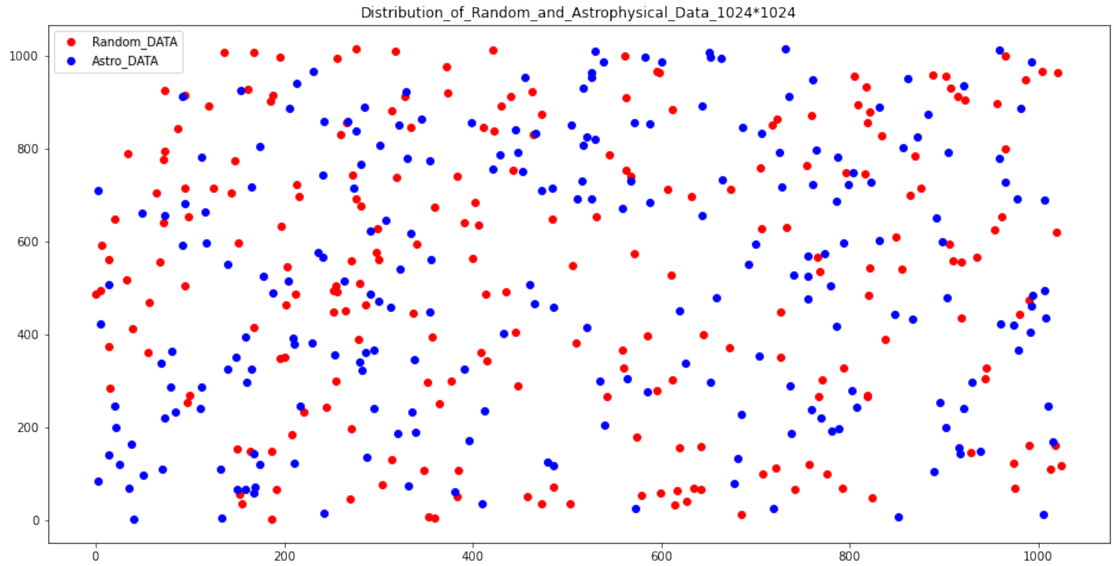


4.0.9 Separated Astral Objects

4.0.10 Stars

[11]:
```
XS1 = np.random.uniform(low=0, high=1024, size=(NPS,))
YS1 = np.random.uniform(low=0, high=1024, size=(NPS,))
XS2 = XS
YS2 = YS
```

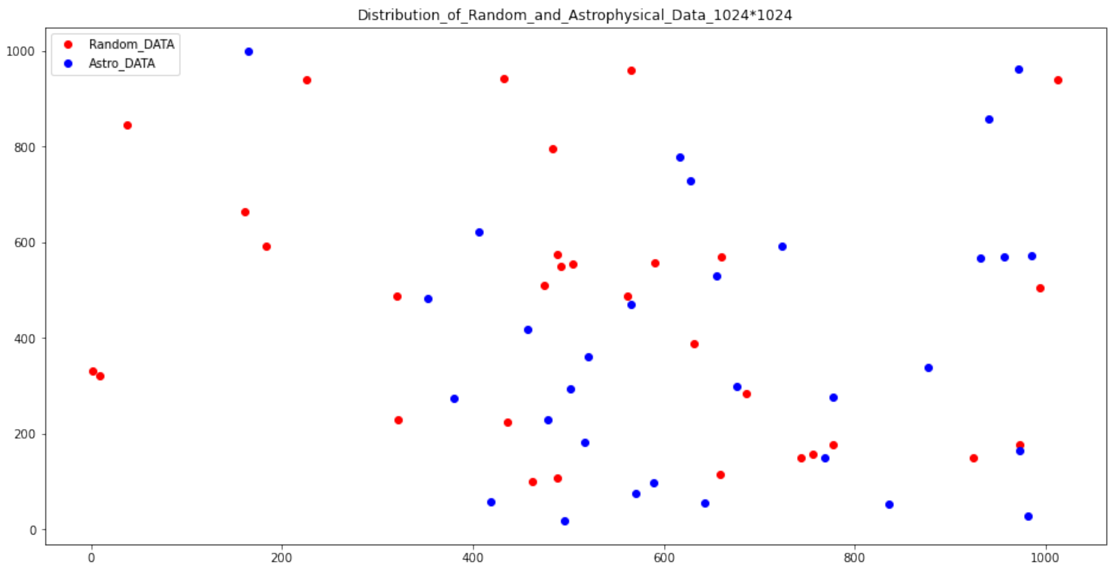
[12]:
```
plt.scatter(XS1,YS1,c='r',label='Random_DATA')
plt.scatter(XS2,YS2,c='b',label='Astro_DATA')
plt.title("Distribution_of_Random_and_Astrophysical_Data_1024*1024")
plt.legend()
plt.draw()
plt.savefig('1024*1024_visualization_Star.pdf')
```



4.0.11 Galaxies

[13]:
```
XG1 = np.random.uniform(low=0, high=1024, size=(NPG,))
YG1 = np.random.uniform(low=0, high=1024, size=(NPG,))
XG2 = XG
YG2 = YG
```

[14]:
```
plt.scatter(XG1,YG1,c='r',label='Random_DATA')
plt.scatter(XG2,YG2,c='b',label='Astro_DATA')
plt.title("Distribution_of_Random_and_Astrophysical_Data_1024*1024")
plt.legend()
plt.draw()
plt.savefig('1024*1024_visualization_Galaxy.pdf')
```



4.0.12 *Creation of Distance Functions*

DISTANCE BETWEEN DATA AND RANDOM (D(D-R))

[15]:
```
def RDdist(x1,y1,x2,y2):
    subsX=[]
    subsY=[]
    RD_Dist=[]
    c1,c2=0,0
    #substraction on x axis:
    for i in x1:
        for j in x2:
            subsX.append(i-j)
    #substraction on y axis:
    for i in y1:
        for j in y2:
            subsY.append(i-j)
    #distances between Data and Random coordinates:
    for i in subsX:
        c1 += 1
        c2 =0
        for j in subsY:
            c2 +=1
            if c1 == c2:
                RD_Dist.append(np.sqrt((i**2)+(j**2)))
    return RD_Dist
```

DISATNCES BETWEEN SAME_COORDINATES (D(D-D) OR D(R-R))

[16]:
```
def DDRRdist(x1,y1):
    subsX=[]
    subsY=[]
    DDRR_Dist=[]
    c1,c2=0,0
    #substraction on x axis:
    for i in x1:
        c1 += 1
        c2 = 0
```

```
        for j in x1:
            c2 += 1
            if c1 != c2:
                subsX.append(i-j)
    #substraction on y axis:
    for i in y1:
        c1 += 1
        c2 = 0
        for j in y1:
            c2 += 1
            if c1 != c2:
                subsY.append(i-j)
    #distances between Data and Random coordinates:
    for i in subsX:
        c1 += 1
        c2 =0
        for j in subsY:
            c2 +=1
            if c1 == c2:
                DDRR_Dist.append(np.sqrt((i**2)+(j**2)))
    return DDRR_Dist
```

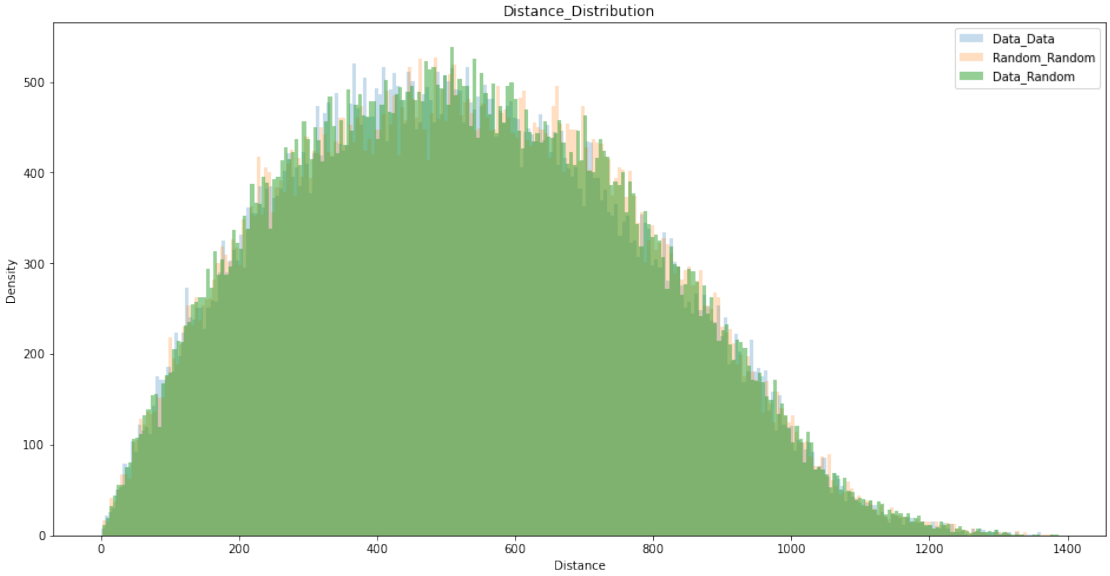
4.0.13 *Calculation and visualization*

4.0.14 *Complete*

[17]:
```
DR = RDdist(X1,Y1,X2,Y2)
DD = DDRRdist(X1,Y1)
RR = DDRRdist(X2,Y2)
```

4.0.15 *Superposed Histograms*

[18]:
```
plt.hist(DD,bins=NP,alpha=0.25,label='Data_Data')
plt.hist(RR,bins=NP,alpha=0.25,label='Random_Random')
plt.hist(DR,bins=NP,alpha=0.5,label='Data_Random')
plt.legend(loc='upper right')
plt.title('Distance_Distribution')
plt.xlabel('Distance')
plt.ylabel('Density')
plt.draw()
plt.savefig('Histograms_superposed.pdf')
```

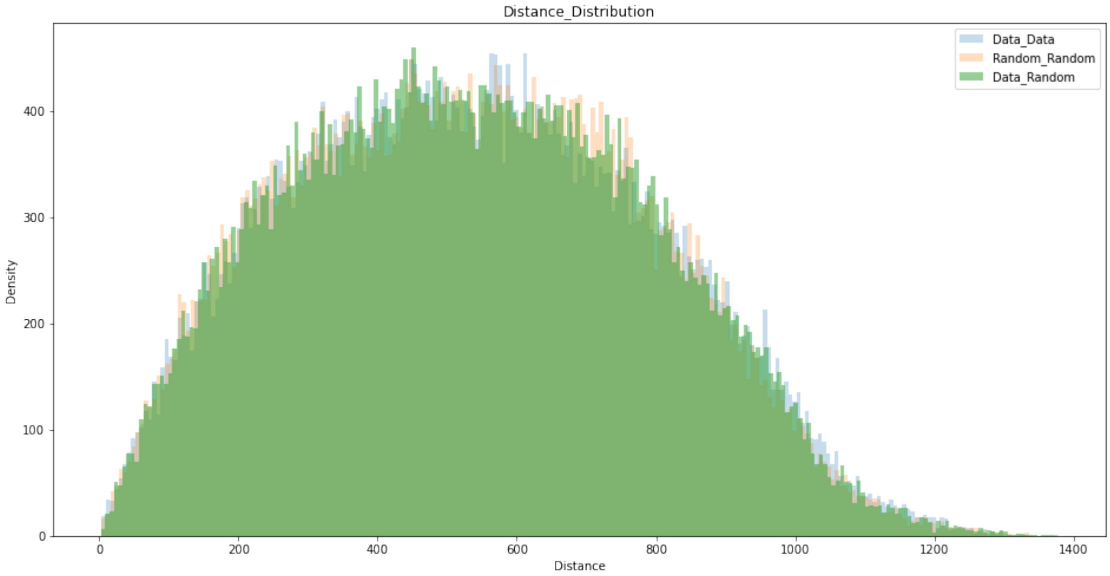


4.0.16 Stars

[19]:
```
DRS = RDdist(XS1,YS1,XS2,YS2)
DDS = DDRRdist(XS1,YS1)
RRS = DDRRdist(XS2,YS2)
```

4.0.17 Superposed Histograms for Stars

[20]:
```
plt.hist(DDS,bins=NPS,alpha=0.25,label='Data_Data')
plt.hist(RRS,bins=NPS,alpha=0.25,label='Random_Random')
plt.hist(DRS,bins=NPS,alpha=0.5,label='Data_Random')
plt.legend(loc='upper right')
plt.title('Distance_Distribution')
plt.xlabel('Distance')
plt.ylabel('Density')
plt.draw()
plt.savefig('Histograms_superposed_stars.pdf')
```

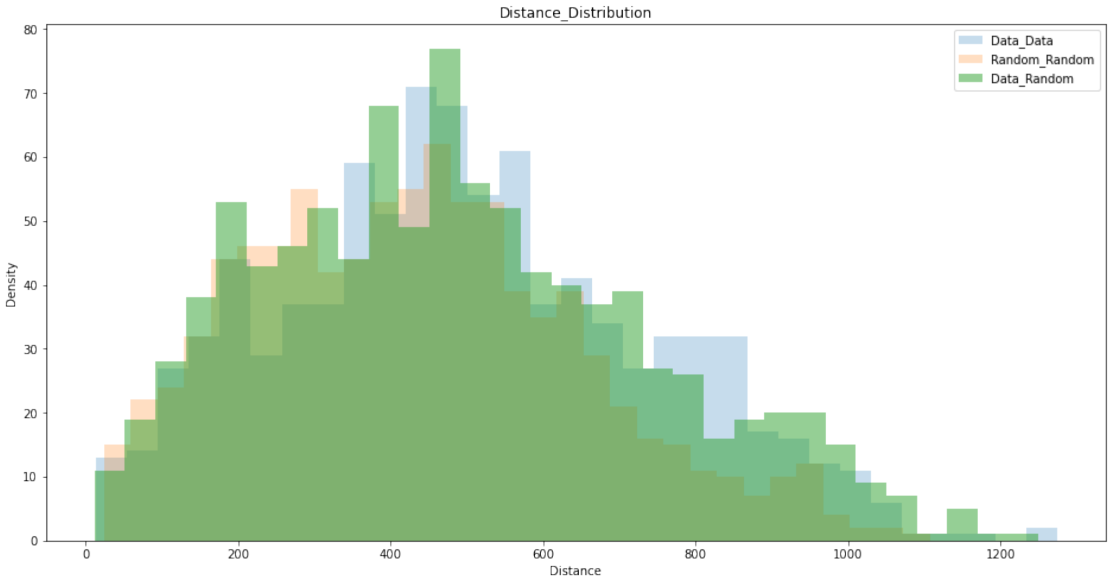


4.0.18 Galaxies

[21]:
```
DRG = RDdist(XG1,YG1,XG2,YG2)
DDG = DDRRdist(XG1,YG1)
RRG = DDRRdist(XG2,YG2)
```

4.0.19 Superposed Histograms for Galaxies

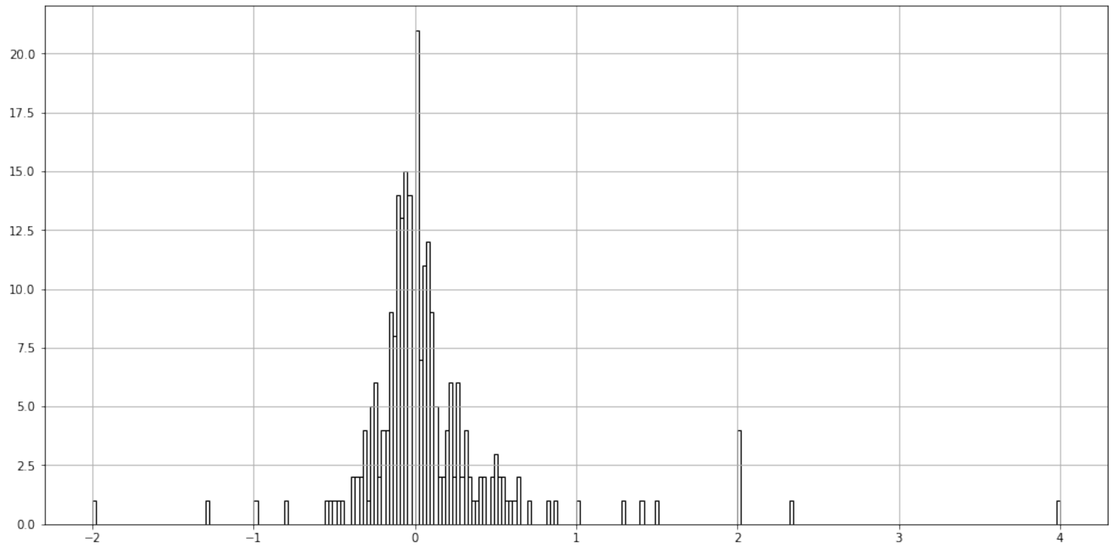
[22]:
```
plt.hist(DDG,bins=NPG,alpha=0.25,label='Data_Data')
plt.hist(RRG,bins=NPG,alpha=0.25,label='Random_Random')
plt.hist(DRG,bins=NPG,alpha=0.5,label='Data_Random')
plt.legend(loc='upper right')
plt.title('Distance_Distribution')
plt.xlabel('Distance')
plt.ylabel('Density')
plt.draw()
plt.savefig('Histograms_superposed_galaxies.pdf')
```



4.0.20 Angular_two_point_correlation_function_full_data

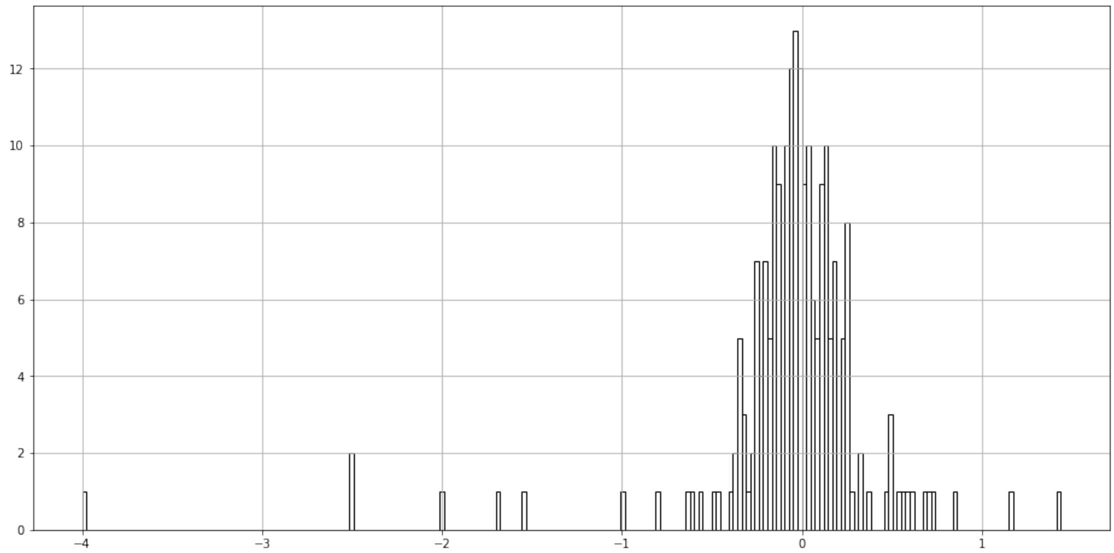
[23]:
```
#def An_Corre_func_Hist(dd,rr,dr):
Total_hist = []
RD_hist, Xedge = np.histogram(DR,bins=NP)
DD_hist, Xedge = np.histogram(DD,bins=NP)
RR_hist, Xedge = np.histogram(RR,bins=NP)
cc1 , cc2, cc3 = 0, 0, 0
for i in DD_hist:
    cc1 += 1
    cc2 = 0
    cc3 = 0
    for j in RR_hist:
        cc2 += 1
        cc3 = 0
```

```
        for k in RD_hist:
            cc3 += 1
            if ((cc1 == cc2) and (cc1 == cc3)):
                if (j!=0):
                    Total_hist.append((i+j-(2*k))/j)

plt.hist(Total_hist,bins=NP, fill=False)
plt.grid()
plt.draw()
plt.savefig('Histogram_Total_Angular_two_point_correlation_function.pdf')
```



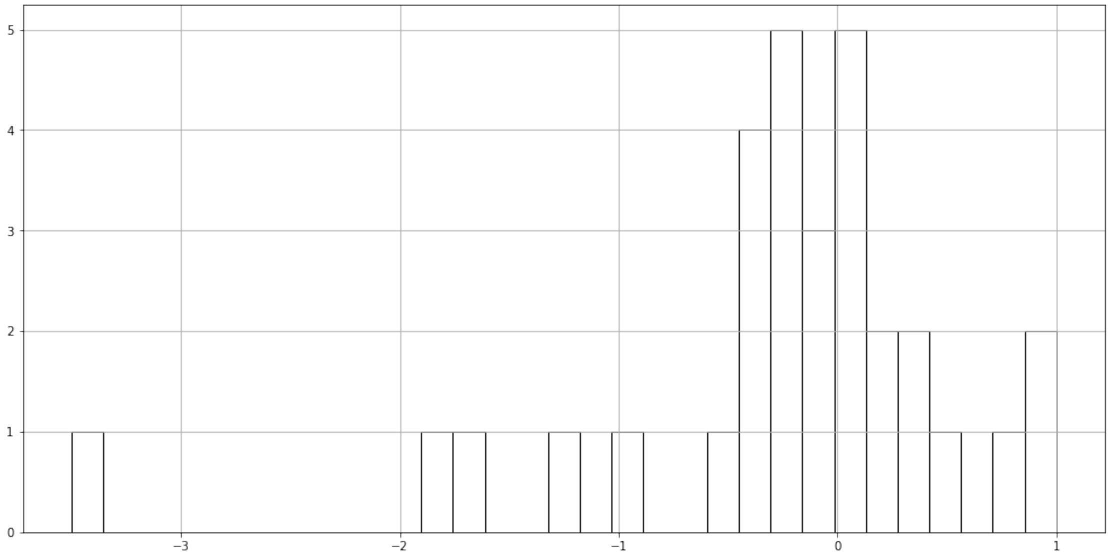
4.0.21 *Angular_two_point_correlation_function_Stars*

[24]:
```
#def An_Corre_func_Hist(dd,rr,dr):
Total_hist_S = []
RDS_hist, XedgeS = np.histogram(DRS,bins=NPS)
DDS_hist, XedgeS = np.histogram(DDS,bins=NPS)
RRS_hist, XedgeS = np.histogram(RRS,bins=NPS)
cc1 , cc2, cc3 = 0, 0, 0
for i in DDS_hist:
    cc1 += 1
    cc2 = 0
    cc3 = 0
    for j in RRS_hist:
        cc2 += 1
        cc3 = 0
        for k in RDS_hist:
            cc3 += 1
            if ((cc1 == cc2) and (cc1 == cc3)):
                if (j!=0):
                    Total_hist_S.append((i+j-(2*k))/j)

plt.hist(Total_hist_S,bins=NPS, fill=False)
plt.grid()
plt.draw()
plt.savefig('Histogram_Total_Angular_two_point_correlation_function_for_Stars.pdf')
```



4.0.22 *Angular_two_point_correlation_function_Galaxies*

[25]:
```
#def An_Corre_func_Hist(dd,rr,dr):
Total_hist_G = []
RDG_hist, XedgeG = np.histogram(DRG,bins=NPG)
DDG_hist, XedgeG = np.histogram(DDG,bins=NPG)
RRG_hist, XedgeG = np.histogram(RRG,bins=NPG)
cc1 , cc2, cc3 = 0, 0, 0
for i in DDG_hist:
    cc1 += 1
    cc2 = 0
    cc3 = 0
    for j in RRG_hist:
        cc2 += 1
        cc3 = 0
        for k in RDG_hist:
            cc3 += 1
            if ((cc1 == cc2) and (cc1 == cc3)):
                if (j!=0):
                    Total_hist_G.append((i+j-(2*k))/j)

plt.hist(Total_hist_G,bins=NPG, fill=False)
plt.grid()
plt.draw()
plt.savefig('Histogram_Total_Angular_two_point_correlation_function_for_Galaxies.pdf')
```



4.0.23 *Mean, Standard_deviation*

4.0.24 *Full data*

[38]:
```
M,S = np.mean(Total_hist),np.std(Total_hist)
```

4.0.25 *Stars*

[39]:
```
SM ,SS = np.mean(Total_hist),np.std(Total_hist)
```

4.0.26 *Galaxies*

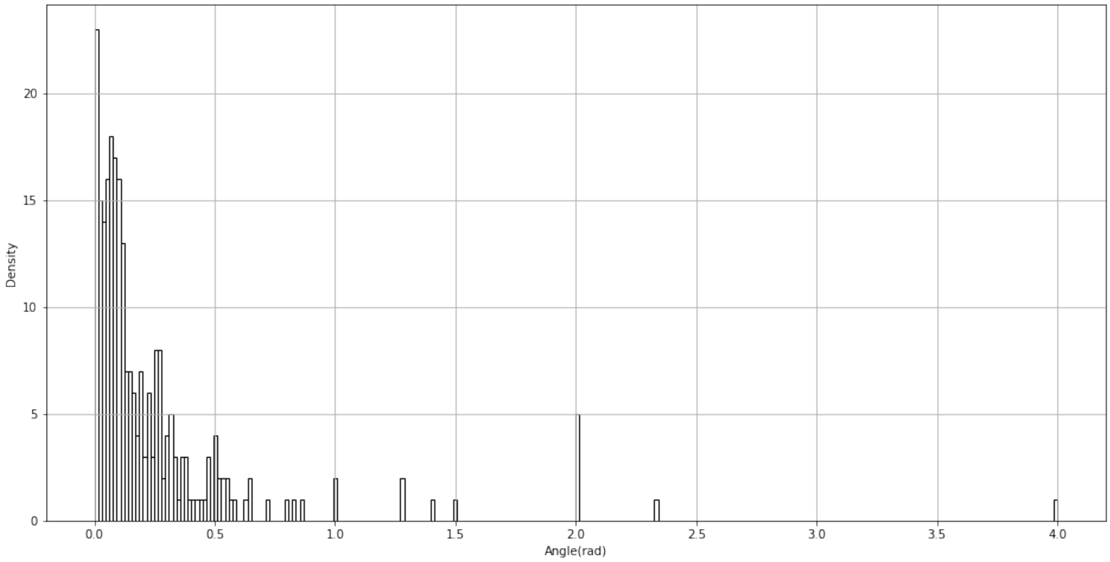
[40]:
```
GM, GS = np.mean(Total_hist),np.std(Total_hist)
```

4.0.27 *Absolute_Angular_two_point_correlation_function_full_data*

[29]:
```
Total_hist_ABS = [abs(i) for i in Total_hist]
```

4.0.28 *Visualization*

[30]:
```
plt.hist(Total_hist_ABS,bins=NP, fill=False)
plt.xlabel('Angle(rad)')
plt.ylabel('Density')
plt.grid()
plt.draw()
plt.savefig('Histogram_Angulartwopointcorrelation_ABS.pdf')
```

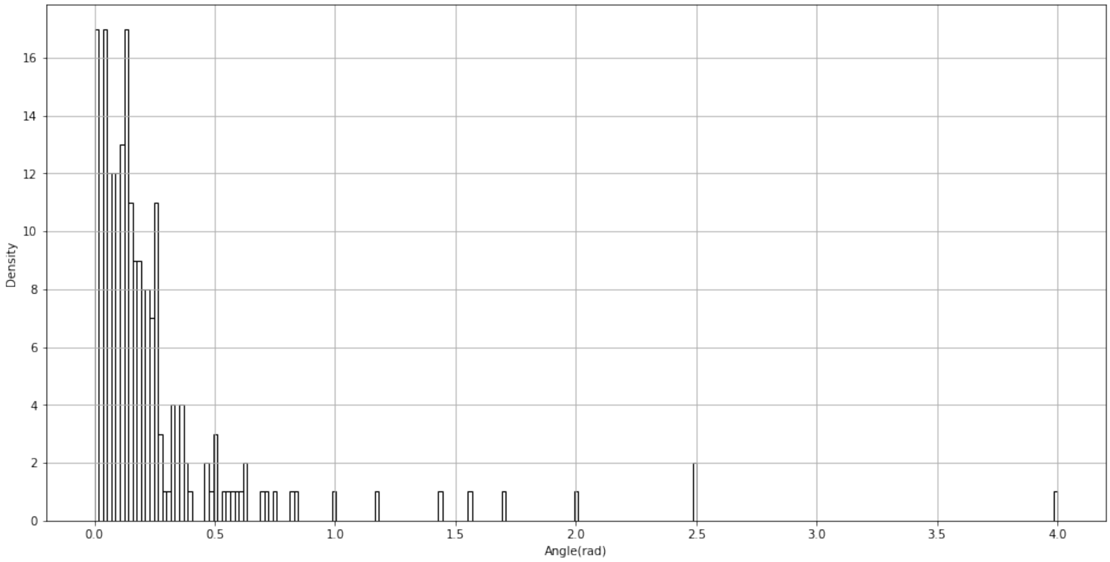


4.0.29 Absolute_Angular_two_point_correlation_function_Stars

[31]:
```
Total_hist_S_ABS = [abs(i) for i in Total_hist_S]
```

4.0.30 Visualization

[32]:
```
plt.hist(Total_hist_S_ABS,bins=NPS, fill=False)
plt.xlabel('Angle(rad)')
plt.ylabel('Density')
plt.grid()
plt.draw()
plt.savefig('Histogram_Angulartwopointcorrelation_ABS_for_Stars.pdf')
```

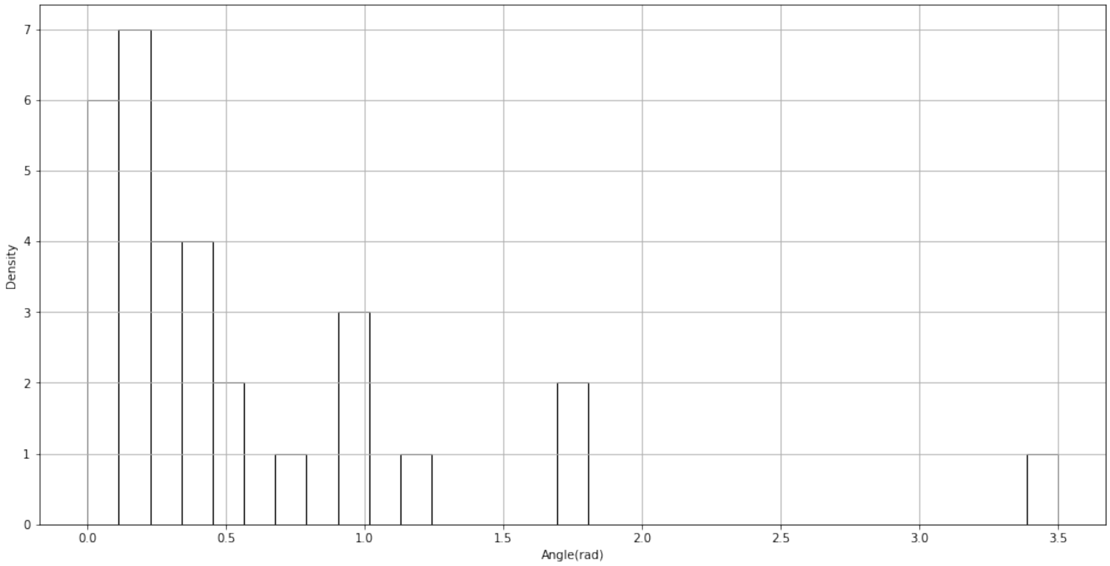


4.0.31 Absolute_Angular_two_point_correlation_function_Galaxies

[33]:
```
Total_hist_G_ABS = [abs(i) for i in Total_hist_G]
```

4.0.32 Visualization

[34]:
```
plt.hist(Total_hist_G_ABS,bins=NPG, fill=False)
plt.xlabel('Angle(rad)')
plt.ylabel('Density')
plt.grid()
plt.draw()
plt.savefig('Histogram_Angulartwopointcorrelation_ABS_for_Galaxies.pdf')
```



4.0.33 Mean, Standard deviation

FULL_DATA

[41]:
```
ABS_M, ABS_S = np.mean(Total_hist_ABS),np.std(Total_hist_ABS)
```

STARS

[42]:
```
ABS_SM, ABS_SS = np.mean(Total_hist_S_ABS),np.std(Total_hist_S_ABS)
```

GALAXIES

[43]:
```
ABS_GM, ABS_GS = np.mean(Total_hist_G_ABS),np.std(Total_hist_G_ABS)
```

[94]:
```
RES=["ABSOLUTE TWO Point angular correlation function All data :\n","\n",
     "Full_Mean:",ABS_M,"        Full_Standard_Deviation:",ABS_S,"\n",
     "Stars_Mean:",ABS_SM,"      Stars_Standard_Deviation:",ABS_SS, "\n",
     "Galaxies_Mean:",ABS_GM,"   Galaxies_Standard_Deviation:",ABS_GS, "\n", "\n", "\n",
     "ORDINARY TWO Point angular correlation function All data:\n","\n",
     "Full_Mean:",M,"       Full_Standard_Deviation:",S, "\n",
     "Stars_Mean:",SM,"      Stars_Standard_Deviation:",SS, "\n",
     "Galaxies_Mean:",GM,"   Galaxies_Standard_Deviation:",GS ,"\n"]
```

[95]:
```
Result = open("Results.txt",'w')
for i in RES:
```

```
    Result.write(str(i))
Result.close()
```

5 WE HAD RUN THE PREVIOUS CODE 19 TIMES

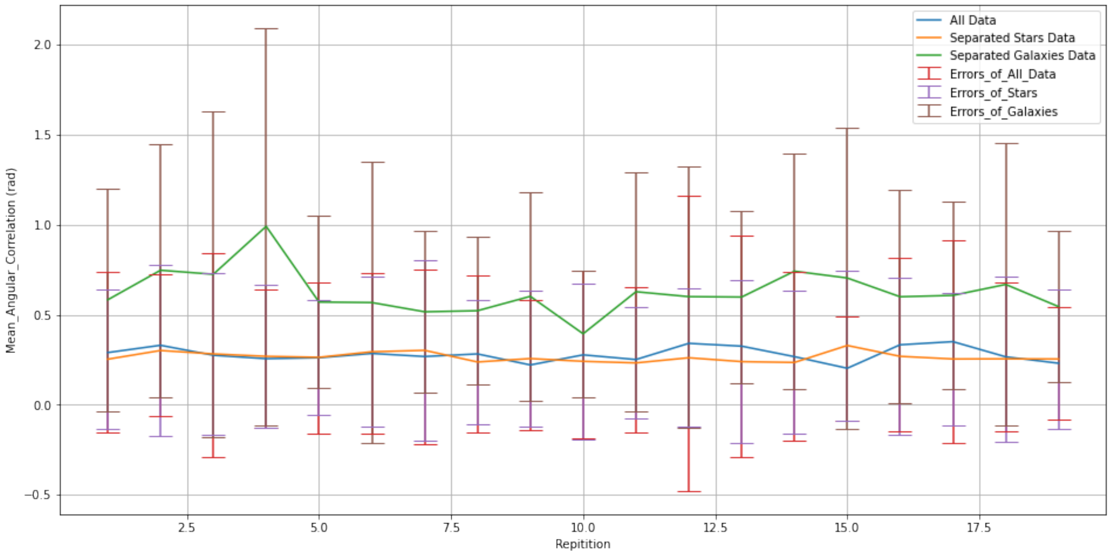
5.0.1 *Extracting Data from Result.txt Files:*

[5]:
```
Number_of_Times= [i for i in range(1,20)]
Total_Std_hist=[0.446715879424833,0.39570794635384876,0.566740371900381,0.
 ↪3819033191708908,0.4209950307398872,0.445885859004162,0.48425892245456864,0.
 ↪4360396396331224,0.36384177813995405,0.46680037565168486,0.4021768289887495,0.
 ↪8208605608097201,0.616211223827725,0.4695642754174025,0.28960042705877,0.
 ↪4809582690892273,0.5604259860348734,0.41512608766092207,0.3100496942852065]
Stars_Std_hist=[0.3903675875246,0.4762697660277469,0.4514289383598142,0.
 ↪3949590084609015,0.3208057569793171,0.417604881489661,0.5026492918231493,0.
 ↪3466341759747653,0.37798662351014345,0.43476605539877444,0.3086336552482156,0.
 ↪3844263457802367,0.4527321644196628,0.3985728940371774,0.4164853700855077,0.
 ↪4351331662221394,0.3698011634062654,0.4582661822531075,0.38792431824665924]
Galaxies_Std_hist=[0.6202785470291529,0.7038201207230281,0.9059862280814963,1.
 ↪1034122476133104,0.47980680267507625,0.781326696854516,0.44678094236581645,0.
 ↪4112886490959973,0.5785587050194914,0.353342921929238,0.6617217864135273,0.
 ↪7260956829104465,0.4755805038917889,0.6542420958759484,0.8367134217810784,0.
 ↪5928235855951426,0.5229951293606696,0.7848226645249823,0.41957667712774416]
Total_Mean_hist=[0.28927677191396867,0.3298597069128274,0.27446727452251496,0.
 ↪2554504875137826,0.2605489540540603,0.2843205769357354,0.26762973168241044,0.
 ↪28179849014064884,0.22067053618864593,0.27664388895663594,0.2506822545370719,0.
 ↪34064089967225913,0.3250425771853553,0.2666683191393442,0.20219134940375405,0.
 ↪3326836081765062,0.3501460398132735,0.265737446934609,0.23017541856444337]
Stars_Mean_hist=[0.25255123879897673,0.3009029142125835,0.28187645643267867,0.
 ↪2685779344445992,0.26311983321726984,0.2938484382095898,0.3017999874394887,0.
 ↪2371154107601975,0.2560071713359439,0.24077839156108205,0.2319980710790171,0.
 ↪26049784865973497,0.23908776117864253,0.23460442888954813,0.32880509405114833,0.
 ↪26853292264820283,0.25381459222728997,0.25453178246910424,0.2538065691520102]
Galaxies_Mean_hist=[0.5824369981392975,0.7468513080739029,0.7245494963663682,0.
 ↪9908493587924502,0.5701213015548535,0.5673313801351904,0.5159954744026761,0.
 ↪5219267781183363,0.6018660797527249,0.39380640400386824,0.6274991276882772,0.
 ↪600159538132886,0.5979188145204716,0.7416099971690316,0.7037824153869325,0.
 ↪5996153469544401,0.6071642875649206,0.6679327707399317,0.5447260475763465]
```

5.0.2 *Visualization of Results:*

[6]:
```
plt.plot(Number_of_Times,Total_Mean_hist,label='All Data')
plt.plot(Number_of_Times,Stars_Mean_hist,label='Separated Stars Data')
plt.plot(Number_of_Times,Galaxies_Mean_hist,label='Separated Galaxies Data')
plt.errorbar(Number_of_Times, Total_Mean_hist, yerr=Total_Std_hist,␣
 ↪capsize=10,label='Errors_of_All_Data',fmt='none')
plt.errorbar(Number_of_Times, Stars_Mean_hist, yerr=Stars_Std_hist,␣
 ↪capsize=10,label='Errors_of_Stars',fmt='none')
plt.errorbar(Number_of_Times, Galaxies_Mean_hist, yerr=Galaxies_Std_hist,␣
 ↪capsize=10,label='Errors_of_Galaxies',fmt='none')
plt.xlabel('Repitition')
plt.ylabel('Mean_Angular_Correlation (rad)')
plt.grid()
plt.legend()
```

```
plt.draw()
plt.savefig('Final_Result_Ultimate.pdf')
```



6 RESULTS & CONCLUSION

- **Results :**
  1. The mean value of $\omega(\theta)$ will be same (around zero), because of a degeneracy that depends on the direction of the angle. Alternatively, an absolute $\omega(|\theta|)$ replaced the original. If we set θ's in absolute values then the symmetry breaks down, and the true mean θ appears.
  2. $\omega(\theta)_i$ are constant functions with fluctuations, due to the errors.
  3. The Mean angle is 0.2791 rad in any object, 0.2643 rad in Stars, 0.6266 rad in Galaxies.
  4. The errors of calculations of θ are least in Stars, slightly higher in overall data, Huge in Galaxies.
- **Conclusion :**
  1. Isotropic distribution of Astral Objects is a must in order to get correct results.
  2. It is trivial to affirm that Galaxies are less abundant in a sky view than stars, which was the case in our data.
  3. The result reflects a physical reality, stars are spread over space in clusters of different forms, they attract each other, what justifies the smallness of θ in this case.
  4. θ in Galaxies is really big, which reveals the weak independance of galaxies from eachother gravitational attraction.
  5. The standard deviations are quite large for all the data, e which unveils the stockastic distribution of astral objects. Also, we were constrained by the number of Random virtual objects, because of our limited ability of calculations (laptops).
  6. errors are Larger in Galaxies because:
     a) Sometimes they behave like pointlike sources exactly like stars when they're very far from us.
     b) They're fewer, then the related data is few, and therefore the standard deviations will be larger.

c) The Illuminating zones in old galaxies shrinks to the center which imitates the Signals of Stars.

7 BIBLIOGRAPHY

BIAS AND VARIANCE OF ANGULAR CORRELATION FUNCTIONS, STEPHEN D.LANDY and Alexander S.SZALAY,The Astrophysical Journal, 412:64-71,1993 July 20.

The SDSS galaxy angular two-point correlation function, Y. Wang, R. J. Brunner and J. C. Dolence, doi:10.1093/mnras/stt450.

https://www.astromatic.net/software/sextractor.