In [ ]:

```
#Importing  relevant modules for later as well as specifying directory
import astropy
import pandas as pd
import numpy as np
import matplotlib as plt
from matplotlib import figure
import matplotlib as plt
import os
import photutils
import astroplan
from astropy.io import ascii

# Set up matplotlib
%matplotlib inline
```

Below shows a bootstrapping technique to produce 95% upper and lower confidence intervals associated with a supernovae rate density estimation. Df refers to a pandas dataframe which contains the required values.The dataset should be taken from the BTS as a csv which is later turned into a dataframe so will contain the required headers and values such as peakabs which refers to the peak absolute magnitude a supernova was observed to have. T refers to the time over which the survey was taken,for example if the sample had supernovae that peaked bewteen 200 to 1500 days then this would yield a T = 3.56 years. n refers to a range of values of absolute magnitudes based on what was found within a dataset.

In [ ]:

```
```

In [ ]:

```
def bootstrap(n,Df,T):
    collective_cumulative_rates = []
    for i in range(0,4000):
        resampled_data = resample(Df,n_samples = len(Df) ,replace = True, stratify = Non
        sliced_data = resampled_data.drop(resampled_data[resampled_data.peakabs>n].index
        #Dataset now left with data cut off  by a particular apparent magnitude
        cumulative_rate = np.sum(sliced_data.rind)/T # Rate formula
        collective_cumulative_rates.append(cumulative_rate) #All 4000 cumulative rates c
    conf_int = np.percentile(collective_cumulative_rates,[2.5,97.5]) # 95% Upper and Low
    print(conf_int)

# for n in np.arange(-16.5,-20.6,-0.005):       <--------- example implentation, ensures
#    bootstrap_Ia(n,18.5)
```

In [ ]:

```
def Rind_PlotIa(Df,mlim,T,f_rec,f_ext,f_cl,f_sky): #Dataframe,limiting apparent Magnitud
    ######################################### CONSTANTS/APPLICATIONS


    Df["peakabs"] = pd.to_numeric(Df["peakabs"], downcast="float")
    f = f_rec*f_ext*f_cl*f_sky
    Mag1 = np.arange(-16.5,-20.6,-0.005) # Absolute Magnitude range


    ######################################### 1ST DATAFRAME
    muA = mlim - Df.peakabs #Essentially mu in the distance modulus formula (m-M)
    mu2A = 0.2*(muA+5-Df.A_V)
    dA= pow(10,mu2A) #Distance modulus
    dGpcA = dA*(1E-9) #Conversion Parsec to GigaParsec
    VolA = (4/3)*np.pi*pow(dGpcA,3) #Volume
    IVA = 1/VolA # Invese of Volume
    RindA = IVA*(1/f) #Individual Rate
    Df["RindA"] = RindA #Adding these individual Rates to Dataframe
    CumSumA = [] #Cumulative Rates
    for n in Mag1:
        yA = Df.drop(Df[Df.peakabs>n].index) #Cutting the dataframe by absolute magnitud
```

```
    bA = np.sum(yA.RindA)/T #Rate Formula
    CumSumA.append(bA)       #Cumulative Rates appended
################################### Bootstrapping
#Add the dataset from the saved upper and lower 95% confidence intervals as a datafr
#Then use this format to plot:
# plt.pyplot.fill_between(Mag1,Lower95%,Upper95%,label = "Title")


################################### Poisson Upper Limit




###################################      PLOTTING THINGS
plt.rcParams["figure.figsize"] = [10, 10]
plt.rcParams["figure.autolayout"] = True
plt.pyplot.figure()
plt.pyplot.plot(Mag1,CumSumA,color = "black",label = "SN Ia rate")
plt.pyplot.gca().invert_xaxis()
plt.pyplot.legend(loc = "lower left")
manager = plt.pyplot.get_current_fig_manager()
manager.full_screen_toggle()
plt.pyplot.yscale("log")
#plt.pyplot.ylim(0,1664)
plt.pyplot.xlim(-16,-20.7)
plt.pyplot.xlabel("Limiting Absolute Magnitude")
plt.pyplot.ylabel("Cumulative Sne Rate ($Gpc^{-3}$ $yr^{-1}$)")
#plt.pyplot.axhline(y=5.12E+02,color = "black")
#plt.pyplot.text(x= -18, y = 1.0E+03,s=r"SN Ia-91T Rate = $5.12 \times 10^{2}   \ Gp
fig = plt.pyplot.gcf()
fig.set_size_inches(8, 7)
plt.pyplot.show(block = True)
print("SN Rate Density is:",format(max(CumSumA),".2E"))
```