UNIVERSITY OF RUHUNA
Sri Lanka

PHY1b22: Elementary Physics Practical I

Project Report

The probability of habitual stars in the galaxy

V T Liyanage (SC/2019/10743)
liyanage10743@usci.ruh.ac.lk
+94767699397

B.Sc. Physical Science

Faculty of Science
University of Ruhuna

Date: 5/18/2021

Contents

Abstract

This project draws upon presenting the occurrence rates of exoplanets in the habitual zones of solar like dwarf stars and implementing methods to efficiently process the data collected from Kepler telescope and other imaging systems based on the NASA Exoplanet Archive, which is an astronomical exoplanet and stellar catalogue and data service that collects, combines and cross-correlates astronomical data and information. The stellar properties are used in developing diagrams by using them in calculations and as parameters in order to analyze and study the characteristics of host stars of the exoplanets that are discovered in our galaxy. Several models are being plotted based on different conditions to determine solar like habitual stars in accordance with analyses which produce meaningful information in order to conclude the results after comparing with the sun. The study outspreads to estimate habitual exoplanets based on planetary properties of the filtered solar like host stars.

Introduction

The universe is infinitely vast and when we reflect on its vastness the earth and the life on earth seems utterly insignificant. The prospect of habituality beyond earth in the universe has been puzzling scientists and astronomers and there have been various attempts to determine probabilities. With the dawn of space missions dedicated to the search of life outside our solar system, such as the NASA Kepler mission launched in 2009 the number of exoplanets detected has strongly risen. More than 4000 planets are now firmly identified and several thousand more need further confirmation. The NASA Exoplanet Archive, astronomical exoplanet catalog and data service that collects and serves public data contains datasets to support the search for and characterization of exoplanets and their host stars. It is part of the Infrared Processing and Analysis Center and is on the campus of the California Institute of Technology (Caltech) in Pasadena, CA. The archive is funded by NASA and was launched in early December 2011 by the NASA Exoplanet Science Institute as part of NASA's Exoplanet Exploration Program. The Exoplanet Archive contains objects discovered through all methods such as radial velocity, transits, microlensing, imaging, astrometry, eclipse timing variations etc. In order to investigate which stars and planets can be expected to have favorable conditions for life, the physical properties of stellar and planetary interiors relevant for habitability can be reviewed. Habitability or ability to harbor life, results from a complex network of interactions of a planet, the system it is a part of, and the star it orbits. The standard definition for a habitable planet is one that can sustain life for a significant period of time. As far as researchers know, this requires a planet to have liquid water. To detect this water from space, it must be on the planet's surface. The region around a star where liquid surface water can exist on a planet's surface is called the "Habitable Zone" (Goldilocks Zone) where conditions might be just right, neither too hot nor too cold for life. However, this definition is confined to our understanding of current and past life on Earth and the environments present on other planets. As researchers learn more and discover new environments in which life can sustain itself, the requirements for life on other planets may be redefined.

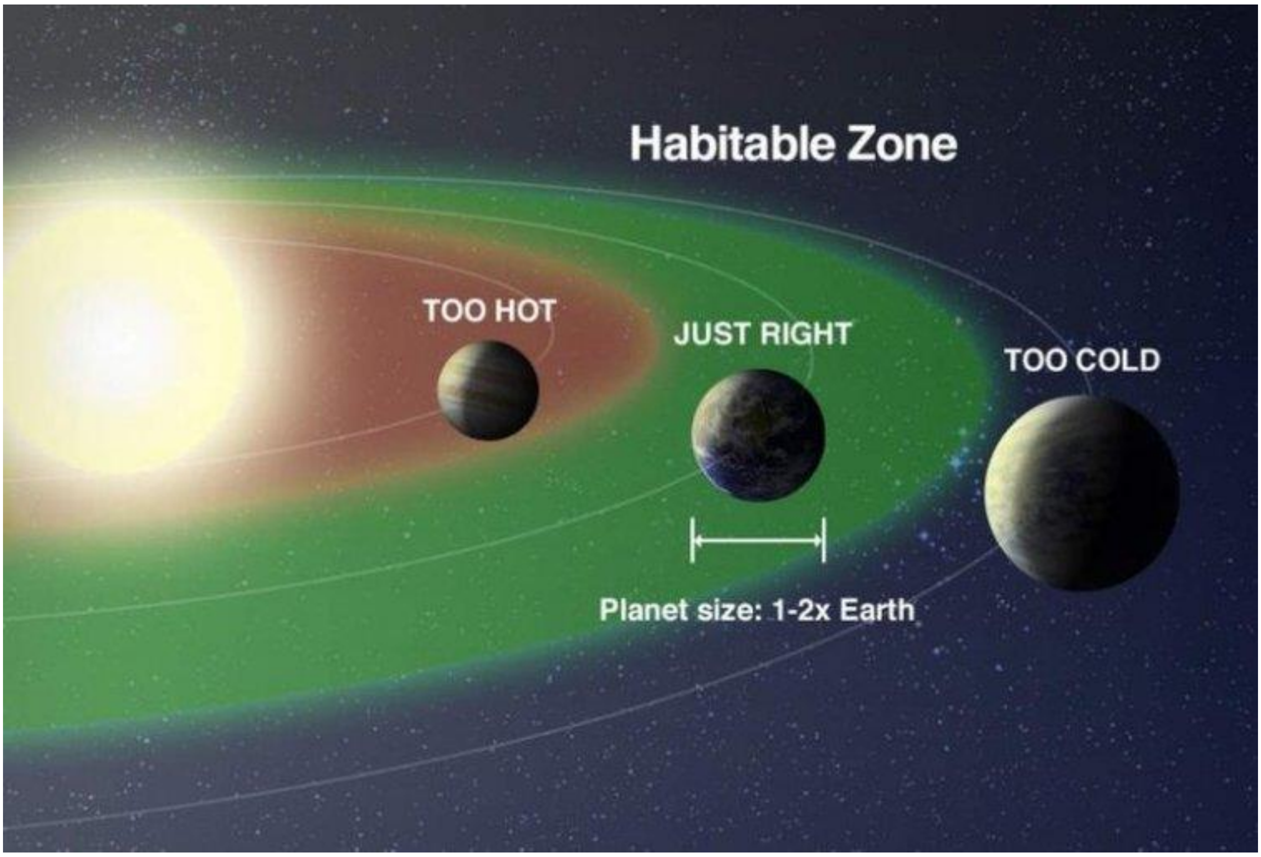

Figure 1

In the early 20th century, after investigating the effects of an object's temperature and of the color of its radiation, scientists reasoned that there should be a relationship between the temperature of a star and its luminosity. If all stars were alike, those with the same luminosity would have equal temperature and hotter stars would be brighter than cooler ones. In 1911, Ejnar Hertzsprung (Denmark), plotted a graph of star's magnitudes against their color. Independently in 1913, Henry Russell (USA), constructed a plot of stars' magnitudes against their spectral class, confirming that indeed, there did seem to be some sort of relationship between a star's luminosity and its temperature, and the stars fell into distinct groups. Such a plot was thereafter named the Hertzsprung-Russell or H-R diagrams. A star on a HR diagram is represented by a dot. Since many stars are usually represented on a HR diagram, there are many dots on the diagram, as shown below. The y axis on a HR diagram represents the star's luminosity and the x axis represents the temperature of the star. Also, the model can be developed with two vertical axes, Absolute magnitude and Luminosity and two horizontal axes Temperature and Spectral class of the star. Stars do not randomly scatter all over an H-R diagram. Instead, they lie on relevant regions according to the various stages of their evolution over time and is the most prominent feature that holds most of the stars which are called as dwarfs lie on a distinctive curved line called the main sequence while the giants and the super giants lie above the main sequence and white dwarfs lie below. HR diagram is one of the most important tools in the study of stellar evolution and analyses related to the stars. They contain an immense amount of information such as brightness, mass and radius as well.

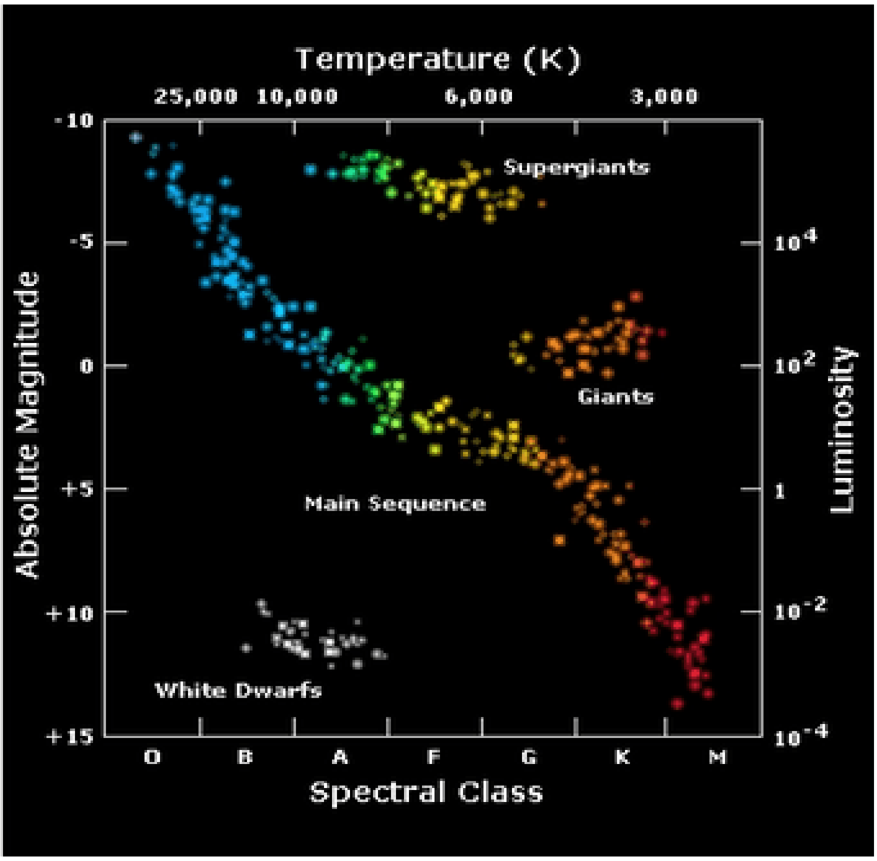

Figure 2

These models can be constructed with the stellar properties taken from the data set in Python modules using a special tool called Jupyter Notebook with annotated python code which is an open-source web application that allows data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning etc. The usage of Jupyter Notebooks has become quite popular in radio astronomy and marks a distinct paradigm shift in the way that astronomers all over the world are collaborating on large science projects providing interactive demos and simulations. It has enabled astrophysical analysis using data proximate computing platforms. On the other hand, Python is a great language for science, and specifically for astronomy. The various packages such as NumPy, SciPy, Scikit-Image and Astropy are all a great testament to the suitability of Python for astronomy, and there are plenty of use cases.

Luminosity – Absolute Magnitude

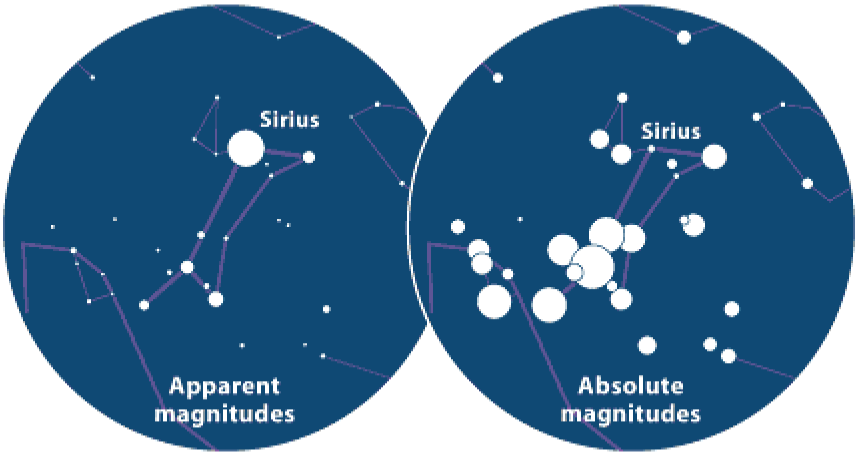

Figure 3

Luminosity, in astronomy, the amount of light emitted by an object in a unit of time. The luminosity of the Sun is 3.846 × 1026 watts (or 3.846 × 1033 ergs per second). Luminosity is an absolute measure of radiant power and its value is independent of an observer's distance from an object. Astronomers usually refer to the luminosity of an object in terms of solar luminosities, with one solar luminosity being equal to the luminosity of the Sun.

Since bright stars can look faint from far away, or dim stars can look bright if they are close, astronomers developed a method to find their true brightness, this is referred to as Absolute Magnitude. They compare how bright they would be if they were all measured at the same distance. It is defined as the apparent magnitude the star would have if it was placed 10 pc away from earth. (1 parsec = 3.26156 Light Years)

$$m - M = 5 \log (d/10)$$

m – Apparent magnitude

M – Absolute Magnitude

d – Distance between the star and the earth in parsecs

Apparent magnitude of the sun is -26.74 and the Absolute magnitude of the sun is +4.83

The distance between the sun and the earth is 1 AU (=4.84814e-6 pc)

Effective Temperature - Spectral class

The temperature of a star is found by using information from a telescope. The temperature of a star can be measured by find the difference between two colors. When stars are viewed with a telescope, their colors are record by using filters. They measure the amount of color the star gives off (measured in the form of a magnitude). When we find the difference between two colors like b and v, we can find the temperature based on the principal of blackbody radiation.

Hot stars appear bluer than cooler stars. Cooler stars are redder than hotter stars. The ``B-V color index'' is a way of quantifying this using two different filters; one a blue (B) filter that only lets a narrow range of colors or wavelengths through centered on the blue colors, and a ``visual'' (V) filter that only lets the wavelengths close to the green-yellow band through.

Spectral class indicates the relative abundance of the different elements in stars and correlates with temperature. Stellar classification, scheme for assigning stars to types according to their temperatures as estimated from their spectra. The generally accepted system of stellar classification is a combination of two classification schemes, the Harvard system which is based on the surface temperature of the star and the MK system which is based on the luminosity of the star.

Figure 4

Earth like Exoplanets

Equilibrium Temperature

According to a study done by Kaltenegger and Sasselov, for planets similar to Earth in size, the limits of the habitability zone can be defined by a planet's equilibrium temperature falling between 175K and 270K. This range considers the temperature needed for liquid water to exist on the surface, given atmospheric conditions, such as the greenhouse effect, that would exist on a theoretically habitable exoplanet.

Exoplanet Size

Size is often thought to be a significant factor, as planets of Earth's size are thought more likely to be terrestrial in nature and be capable of retaining an Earth-like atmosphere. The list includes planets within the range of 0.8–1.9 Earth masses, below which are generally classed as sub-Earth and above classed as super-Earth. In addition, only planets known to fall within the range of 0.5–2.0 Earth radius (between half and twice the radius of the Earth) are included. In contrast, the Earth Similarity Index uses both mass and radius as criteria.

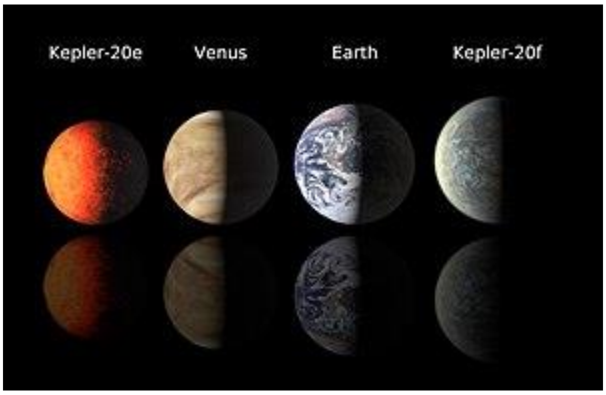

Figure 5

Eccentricity

The eccentricity of a planet's orbit can be crucial for determining habitability. If a planet's orbit is too eccentric it can lead to planets that cannot maintain liquid water year-round as they move farther and closer to their host star. Linsenmeier et al. (2015) studied the influence of both obliquity and eccentricity for ocean covered planets orbiting a Sun-like star on a 365-day orbit and a 24-hour day, like Earth. They found that planets with eccentricities higher than 0.2 can only sustain surface liquid water for a part of the year.

Methodology

The relationship between the stellar properties, effective temperature and luminosity were represented by plotting an H-R diagram with Python programming modules using Jupyter Notebook application. The required data for the parameters were directly used from the data set.

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np

total_stars_no_dup = dataset.copy()
print(len(total_stars_no_dup))
total_stars_no_dup.drop_duplicates(subset ="hostname",keep = False, inplace = True)

# Format the points
transparency = 0.6
size = 5

# Draws the scatter plot

plot = plt.scatter(dataset.st_teff, dataset.st_lum, s=size, edgecolors='none', alpha=transparency)
plt.xlim(10000,2000)
plt.title("H-R Diagram")
plt.ylabel("Luminosity [log (solar)]")
plt.xlabel("Temperature [K]");
```

```
8068
```

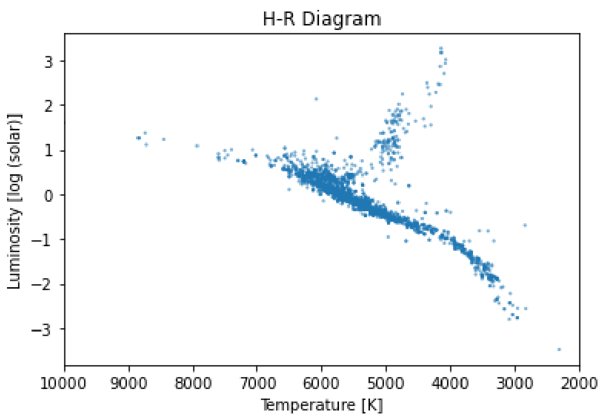

Figure 6

Then the solar like stars were identified based on the effective temperature and luminosity of all stars by filtering them through the conditions that need to be fulfilled in order to be a sun like star.

```python
def filtered_star_assigner (row):
   if row['st_teff'] >= 5500 and row['st_teff'] <= 6300 and row['st_lum'] >= -0.5 and row['st_lum'] <= 0.5:
      return 'red'
   return 'blue'

def filtered_star_assigner_value (row, column):
    if row[column] == 'red':
        return 1
    elif row[column] == 'blue':
        return 0
    elif row[column] == 'green':
        return 2

dataset['filtered_color'] = dataset.apply (lambda row: filtered_star_assigner(row), axis=1)
dataset['filtered_color_value'] = dataset.apply (lambda row: filtered_star_assigner_value(row, 'filtered_color'), axis=1)
dataset = dataset.sort_values('filtered_color_value')

display_dataset = dataset[(dataset['filtered_color'] == 'red')].copy()
display_dataset.rename(columns = {'st_teff':'Temperature', 'st_lum':'Luminosity'}, inplace = True)
display_dataset.drop_duplicates(subset ="hostname",keep = False, inplace = True)
```

```python
# Draws the scatter plot of filtered stars
plot = plt.scatter(dataset.st_teff, dataset.st_lum, c = dataset.filtered_color, s=size, alpha=transparency)
plt.xlim(10000,2000)
plt.title("H-R Diagram (Filtered Stars)")
plt.ylabel("Luminosity [log (solar)]")
plt.xlabel("Temperature [K]");
# with pd.option_context('display.max_rows', None, 'display.max_columns', None):  # more options can be specified also
#     display(display_dataset[['Temperature', 'Luminocity']])
```

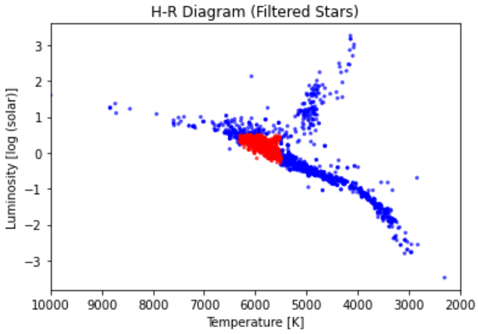

Figure 7

An improved model of H-R diagram was constructed using multiple axes with Luminosity and Absolute Magnitude as vertical axes and Effective Temperature and Spectral class as the horizontal axes. The relationship between the Absolute Magnitude and Luminosity was derived in the following equation by substituting the values for Sun's Luminosity and Absolute Magnitude.

$$M = -2.5 \log (L) + 4.83$$

M – Absolute Magnitude

L – Luminosity

In [181]:
```python
import matplotlib.ticker as ticker

filtered_dataset = dataset[dataset['st_teff'].between(2000, 10000)]

fig, ax1 = plt.subplots(1, 1)
ax2 = ax1.twinx()
ax1.scatter(filtered_dataset.st_teff, -2.5 * filtered_dataset.st_lum + 4.83, alpha=transparency/2)
ax2.scatter(filtered_dataset.st_teff, np.power(10, filtered_dataset.st_lum), alpha=transparency/2)
ax2.set_yscale('log')
plt.xlim(11000,1000)
ax1.invert_yaxis()

ax3 = ax1.twiny()
plt.xlim(11000,1000)
ax3.xaxis.set_major_formatter(ticker.NullFormatter())
ax3.xaxis.set_minor_locator(ticker.FixedLocator([2400,3700,5200,5999,7000,10001]))
ax3.xaxis.set_minor_formatter(ticker.FixedFormatter(["M0","K0","G0","F0","A0","B0"]))

plt.title("H-R Diagram")
ax1.set_ylabel("Absolute Magnitude")
ax2.set_ylabel("Luminosity [log (solar)]")
ax3.set_xlabel("Spectral Class")
ax1.set_xlabel("Temperature [K]")
```

Out[181]: Text(0.5, 0, 'Temperature [K]')

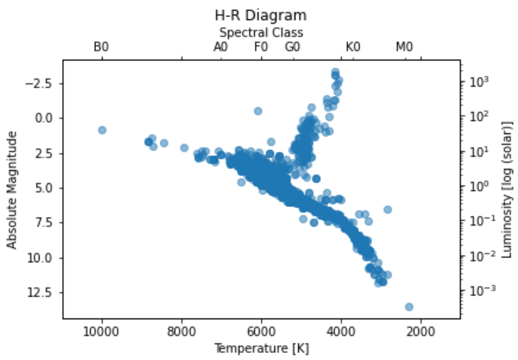

Figure 8

In [182]:
```python
fig, ax1 = plt.subplots(1, 1)
ax2 = ax1.twinx()
ax1.scatter(filtered_dataset.st_teff, -2.5 * filtered_dataset.st_lum + 4.83, c = filtered_dataset.filtered_color, alpha=transpare
ax2.scatter(filtered_dataset.st_teff, np.power(10, filtered_dataset.st_lum), c = filtered_dataset.filtered_color, alpha=transpare
ax2.set_yscale('log')
plt.xlim(11000,1000)
ax1.invert_yaxis()

ax3 = ax1.twiny()
plt.xlim(11000,1000)
ax3.xaxis.set_major_formatter(ticker.NullFormatter())
ax3.xaxis.set_minor_locator(ticker.FixedLocator([2400,3700,5200,5999,7000,10001]))
ax3.xaxis.set_minor_formatter(ticker.FixedFormatter(["M0","K0","G0","F0","A0","B0"]))

plt.title("H-R Diagram (Filtered Stars)")
ax1.set_ylabel("Absolute Magnitude")
ax2.set_ylabel("Luminosity [log (solar)]")
ax3.set_xlabel("Spectral Class")
ax1.set_xlabel("Temperature [K]")
```

Out[182]: Text(0.5, 0, 'Temperature [K]')

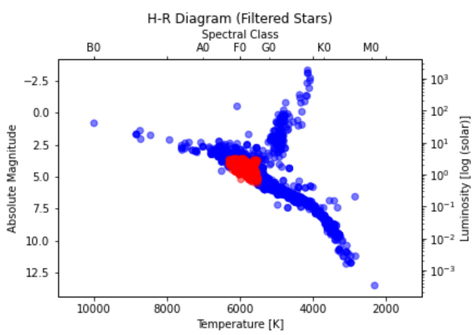

Figure 9

According to the data set most records are missing in the spectral class column. Therefore, more precise module was developed without considering the spectral class property.

```
In [183]: fig, ax1 = plt.subplots(1, 1)
          ax2 = ax1.twinx()
          ax1.scatter(filtered_dataset.st_teff, -2.5 * filtered_dataset.st_lum + 4.83, alpha=transparency/2)
          ax2.scatter(filtered_dataset.st_teff, np.power(10, filtered_dataset.st_lum), alpha=transparency/2)
          ax2.set_yscale('log')
          plt.xlim(11000,1000)
          ax1.invert_yaxis()

          plt.title("H-R Diagram")
          ax1.set_ylabel("Absolute Magnitude")
          ax2.set_ylabel("Luminosity [log (solar)]")
          ax1.set_xlabel("Temperature [K]")

Out[183]: Text(0.5, 0, 'Temperature [K]')
```

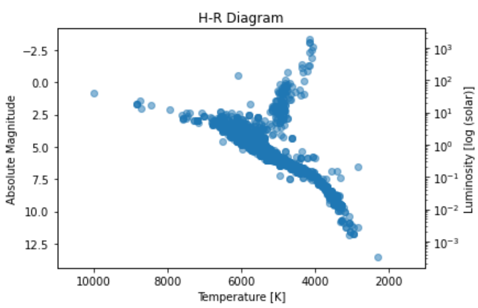

Figure 10

Then the same model was developed to represent filtered solar like stars as well and it was used to determine the solar like stars count using the diagram.

```
filtered_dataset_plot = filtered_dataset.copy()

fig, ax1 = plt.subplots(1, 1)
ax2 = ax1.twinx()
ax1.scatter(filtered_dataset_plot.st_teff, -2.5 * filtered_dataset_plot.st_lum + 4.83, c = filtered_dataset_plot.filtered_color,
ax2.scatter(filtered_dataset_plot.st_teff, np.power(10, filtered_dataset_plot.st_lum), c = filtered_dataset_plot.filtered_color,
ax2.set_yscale('log')
plt.xlim(11000,1000)
ax1.invert_yaxis()

plt.title("H-R Diagram (Filtered Stars)")
ax1.set_ylabel("Absolute Magnitude")
ax2.set_ylabel("Luminosity [log (solar)]")
ax1.set_xlabel("Temperature [K]")
print("Filtered stars list count: " + str(len(display_dataset)))
print("Total stars count: " + str(len(total_stars_no_dup)))
with pd.option_context('display.max_rows', None, 'display.max_columns', None):  # more options can be specified also
    display(display_dataset[['Temperature', 'Luminosity']])
```

Figure 11

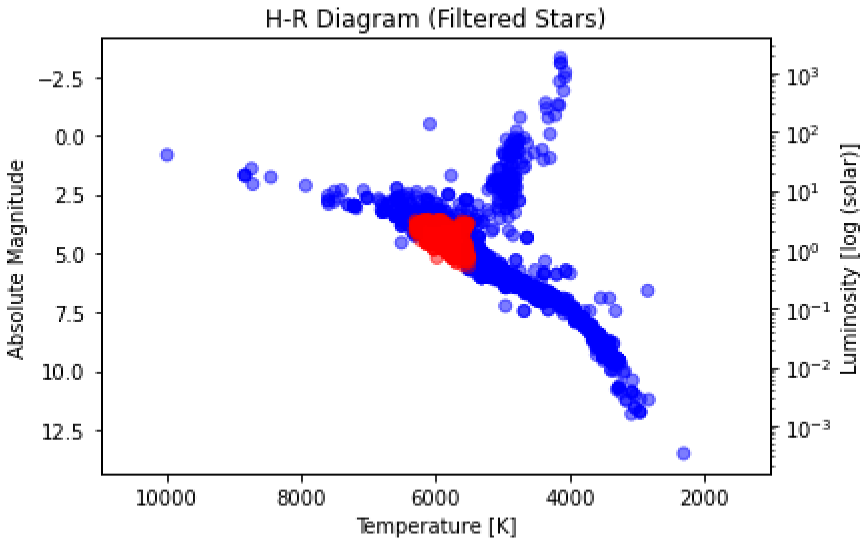

Figure 12

```
Filtered stars list count: 228
Total stars count: 8068
```

	Temperature	Luminosity
2861	5573.35	-0.060
1359	6076.00	0.346
1322	5883.00	-0.010
2776	5505.60	-0.289
1186	5995.71	0.213
1121	5533.00	-0.001
1124	5783.00	0.301
2812	6046.57	0.462
3201	5681.00	-0.217

Figure 13

Also, further analysis was done to determine the possibility of existing earth like exoplanets orbiting the sun like stars. In this case, planetary properties such as equilibrium temperature, size of the planet and eccentricity were considered to identify potential exoplanets similar to the earth.

```
In [185]: def pl_filtered_star_assigner (row):
              if row['filtered_color'] == 'red':
          #         if row['pl_massj'] >= 0.8 and row['pl_massj'] <= 1.9 and row['pl_rade'] >= 0.5 and row['pl_rade'] <= 2.0 and row['plt_eqt']
                  if ((row['pl_masse'] >= 0.8 and row['pl_masse'] <= 1.9) and (row['pl_rade'] >= 0.5 and row['pl_rade'] <= 2.0)) or (row['p
                      return 'green'
                  else:
                      return 'red'
              else:
                  return 'blue'

          # dataset.drop_duplicates(subset ="pl_hostname",keep = False, inplace = True)


          dataset['pl_filtered_color'] = dataset.apply (lambda row: pl_filtered_star_assigner(row), axis=1)
          dataset['pl_filtered_color_value'] = dataset.apply (lambda row: filtered_star_assigner_value(row, 'pl_filtered_color'), axis=1)
          dataset = dataset.sort_values('pl_filtered_color_value')

          filtered_dataset = dataset[dataset['st_teff'].between(2000, 10000)]

          fig, ax1 = plt.subplots(1, 1)
          ax2 = ax1.twinx()
          ax1.scatter(filtered_dataset.st_teff, -2.5 * filtered_dataset.st_lum + 4.83, c = filtered_dataset.pl_filtered_color, alpha=transp
          ax2.scatter(filtered_dataset.st_teff, np.power(10, filtered_dataset.st_lum), c = filtered_dataset.pl_filtered_color, alpha=0)
          ax2.set_yscale('log')
          plt.xlim(11000,1000)
          ax1.invert_yaxis()

          plt.title("H-R Diagram (Filtered Stars with Earth like Exoplanets)")
          ax1.set_ylabel("Absolute Magnitude")
          ax2.set_ylabel("Luminosity [log (solar)]")
          ax1.set_xlabel("Temperature [K]")



          display_dataset = filtered_dataset[(filtered_dataset['filtered_color'] == 'red')].copy()
          display_dataset.rename(columns = {'st_teff':'Temperature', 'st_lum':'Luminosity'}, inplace = True)
          display_dataset.drop_duplicates(subset ="pl_name",keep = False, inplace = True)
          print("List count: " + str(len(display_dataset)))
          print("Filtered stars with exoplanets list count: " + str(len(filtered_dataset[(filtered_dataset['pl_filtered_color'] == 'green')
          print("Total stars count: " + str(len(total_stars_no_dup)))
          with pd.option_context('display.max_rows', None, 'display.max_columns', None):  # more options can be specified also
              display(display_dataset[['Temperature', 'Luminosity']])
```

Figure 14

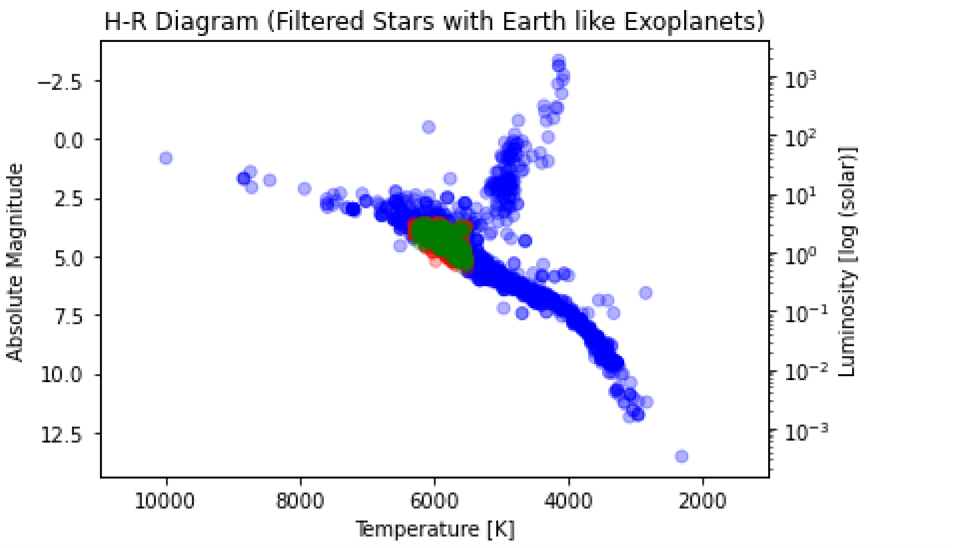

Figure 15

```
In [185]: def pl_filtered_star_assigner (row):
              if row['filtered_color'] == 'red':
          #       if row['pl_massj'] >= 0.8 and row['pl_massj'] <= 1.9 and row['pl_rade'] >= 0.5 and row['pl_rade'] <= 2.0 and row['plt_eqt']
                  if ((row['pl_masse'] >= 0.8 and row['pl_masse'] <= 1.9) and (row['pl_rade'] >= 0.5 and row['pl_rade'] <= 2.0)) or (row['p
                      return 'green'
                  else:
                      return 'red'
              else:
                  return 'blue'

          # dataset.drop_duplicates(subset ="pl_hostname",keep = False, inplace = True)


          dataset['pl_filtered_color'] = dataset.apply (lambda row: pl_filtered_star_assigner(row), axis=1)
          dataset['pl_filtered_color_value'] = dataset.apply (lambda row: filtered_star_assigner_value(row, 'pl_filtered_color'), axis=1)
          dataset = dataset.sort_values('pl_filtered_color_value')

          filtered_dataset = dataset[dataset['st_teff'].between(2000, 10000)]

          fig, ax1 = plt.subplots(1, 1)
          ax2 = ax1.twinx()
          ax1.scatter(filtered_dataset.st_teff, -2.5 * filtered_dataset.st_lum + 4.83, c = filtered_dataset.pl_filtered_color, alpha=transp
          ax2.scatter(filtered_dataset.st_teff, np.power(10, filtered_dataset.st_lum), c = filtered_dataset.pl_filtered_color, alpha=0)
          ax2.set_yscale('log')
          plt.xlim(11000,1000)
          ax1.invert_yaxis()

          plt.title("H-R Diagram (Filtered Stars with Earth like Exoplanets)")
          ax1.set_ylabel("Absolute Magnitude")
          ax2.set_ylabel("Luminosity [log (solar)]")
          ax1.set_xlabel("Temperature [K]")



          display_dataset = filtered_dataset[(filtered_dataset['filtered_color'] == 'red')].copy()
          display_dataset.rename(columns = {'st_teff':'Temperature', 'st_lum':'Luminosity'}, inplace = True)
          display_dataset.drop_duplicates(subset ="pl_name",keep = False, inplace = True)
          print("List count: " + str(len(display_dataset)))
          print("Filtered stars with exoplanets list count: " + str(len(filtered_dataset[(filtered_dataset['pl_filtered_color'] == 'green')
          print("Total stars count: " + str(len(total_stars_no_dup)))
          with pd.option_context('display.max_rows', None, 'display.max_columns', None):  # more options can be specified also
              display(display_dataset[['Temperature', 'Luminosity']])
```

Figure 16

Finally, the probability of habitual stars in the galaxy and the probability of habitual stars with habitual exoplanets were estimated using the counts obtained using the models.

```
List count: 228
Filtered stars with exoplanets list count: 221
Total stars count: 8068
```

	Temperature	Luminosity
1999	6139.00	0.291
2029	5566.00	0.141
2007	5582.00	0.293
2010	5582.00	0.293
197	5529.00	-0.131
2202	5748.00	0.030
2206	5524.00	-0.002
7967	5695.00	-0.055
2408	5927.14	0.055

Figure 17

Conclusion

The probability of habitual stars in the galaxy = 228/8068 = 0.0283

The probability of habitual stars with habitual exoplanets = 221/8068 = 0.0274

Discussion

In this project, when estimating the sun like stars and earth like exoplanets the values of the stellar properties and the planetary properties were filtered through the conditions using the standards set by the previous research and studies.

In the dataset used for these analyses only the data of the confirmed exoplanets and their host stars are presented, and there are more candidates which need further clarifications. Therefore, the future analyses might draw different conclusions with the updated datasets.

Since the concept of habitability is defined according to the life on earth and its potential, the actual extraterrestrial life and their habitability might be different.

The data of the exoplanets discovered so far contains only the data inside our galaxy as the other galaxies are too distant to be explored with the technology and the discovery methods we can use in the present time.

Even though, the planet mass plays an important role in comparing exoplanets with the earth, most records of the dataset do not contain the planet mass data. When considering the other planetary properties as well most records do not contain all the planetary properties that are being considered.

According to the results it can be identified that most solar like stars have at least one exoplanet orbiting around it with an earth like property.

Future work

Further analyses on probability of life beyond earth could be done using the Drakes Equation, estimating its parameters. Completed tasks can cover some of the parameters such as, and other parameters require further studies. Additionally, a deeper evaluation and studying would be beneficial about the habitual exoplanets discovered through filtering. Furthermore, we could try to incorporate tidal locking into our determination of what could be a habitable work. We could also try to get our data to be more accurate by cross referencing the NASA exoplanet database with other databases to fill in any gaps that there were.

References

http://www2.lowell.edu/users/jch/workshop/drs/drs-p1.html

https://en.wikipedia.org/wiki/Solar_analog

https://cdn.technologynetworks.com/ep/pdfs/determining-the-habitability-of-exoplanets.pdf

https://en.wikipedia.org/wiki/Earth_analog

https://www.nasa.gov/ames/kepler/nasas-kepler-discovers-first-earth-size-planet-in-the-habitable-zone-of-another-star

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4375622/

https://exoplanets.nasa.gov/search-for-life/habitable-zone/

https://www.tandfonline.com/doi/full/10.1080/23746149.2019.1630316

https://iopscience.iop.org/article/10.3847/1538-4357/ab3be8

http://bl.ocks.org/nbremer/eb0d1fd4118b731d069e2ff98dfadc47

https://astronomy.swin.edu.au/cosmos/a/Absolute+Magnitude

https://www.atnf.csiro.au/outreach/education/senior/cosmicengine/stars_hrdiagram.html

Acknowledgement

I would like to give my special thanks to Dr. K V S Prasad for the advice given throughout this project and the time he spent on supporting. Also, I would like to thank Dr. Manjula for the understanding and encouraging and Prof. Kanthi Yapa for the guidance on doing these kinds of projects.