Generating and Assessing Stellar Evolution Models

AST221 Final Project: Summary Report

By Ali Ahmed

1007192592

December 6th, 2021

AST221H5F: Astrophysics I - Planets, Sun, and Stars

Dr. Lea Hirsch

Introduction

Nuclear fusion operations transmute hydrogen into helium, allowing stars to illuminate. When you understand that it is turning mass to energy, this is hardly unexpected. Because a star does not have unlimited mass, it also does not have unlimited energy, and as a result, stars go through stellar evolution, which may be classified into a few stages. The main sequence phase of a star is the period in a star's lifetime when it is in the most steady and consistent state, this is when there is more than enough hydrogen fuel to burn in its core. The is because stars can be volatile and unpredictable shortly after forming from a disintegrating cloud of dust and gas, as well as later in life when they run out of fuel, but this is not the case during the main sequence. Astronomers discovered to distinguish the difference between steady stars and the more unpredictable, young, or old stars by graphing two fundamental observable properties as Luminosity and color against each other, making a plot known as a Hertzsprung-Russell (HR) Diagram. They determined that the positions of the steady and non-erratic stars throughout their lifetime were limited to a tiny section of this graph, along a specific range known as the main sequence. Another observation that was made as the basic stellar attribute that varied as one examined the main sequence section of the graph through one end toward the other was their stellar mass.

The purpose of this project is to comprehend the significance of the HR diagram in interpreting stellar evolution, to learn how observations of a star's physicochemical parameters define its placement on the HR diagram, and, more specifically, how the star's mass directly impacts its main-sequence lifetime.

Additionally, the main sequence lifetime of different masses of stars can also be determined using the following equation:

$$\tau \approx 10^{10} \left(\frac{M_{\odot}}{M}\right)^{2.5} yrs \ (1)$$

Where $M_{\odot}$ is the mass of the sun and M is the mass of a star whose main-sequence lifetime is to be calculated. However, this equation is not entirely correct and cannot be referred to as the hypothetical value to be calculated, since it has been constructed to calculate the sun's main sequence lifetime time accurately and has been constructed by the data collected from the observation of the sun. As a result, the equation may be used to approximate the lifetime of a specific star based on empirical data and interpretations of that data. Therefore it can be treated as more of an experimental value and also thus has the "≈" approximate sign. It may be used for comparison, to the data collected from the EZ-Web database, where the stellar properties have been calculated and stored using the known physics of stars and laws that govern the universe and thus may be referred to as the theoretical values for comparison.

Methods

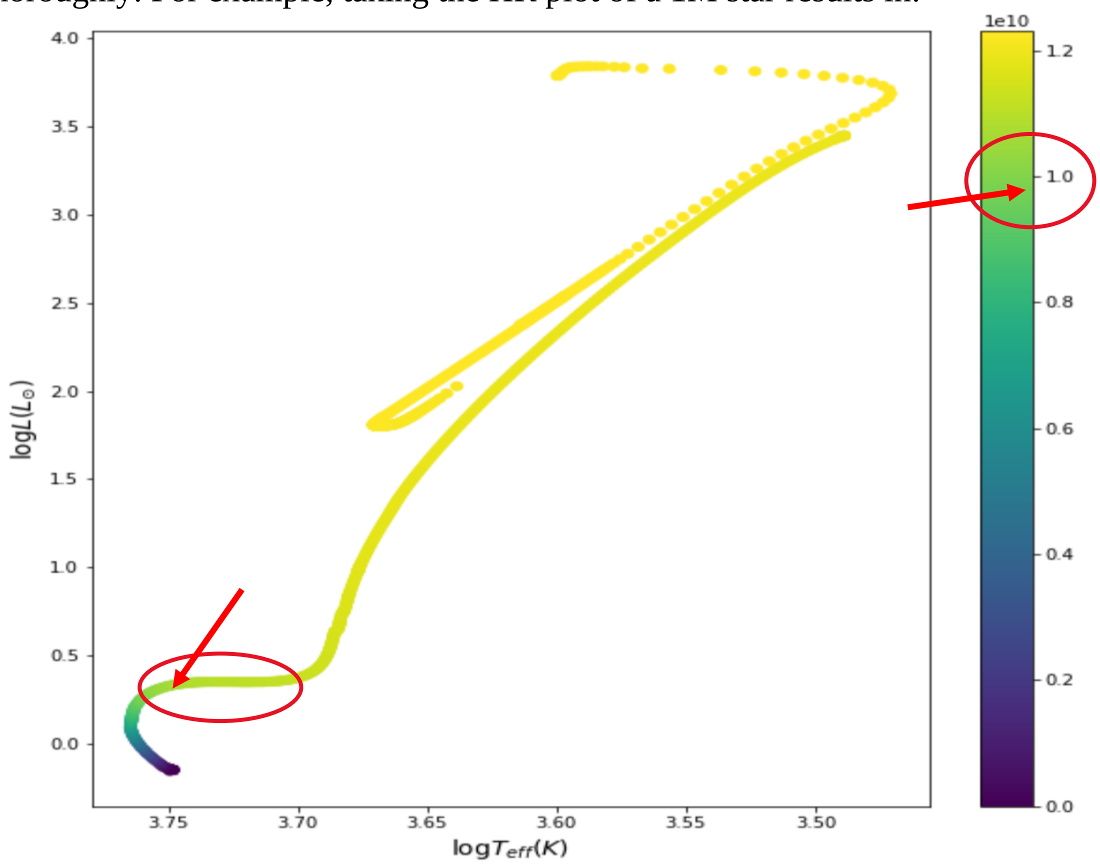
All data for the stellar properties for all different masses stars are retrieved and downloaded from the EZ-Web database. The files are extracted and saved in the same folder as the Jupyter Notebook, through which the data analysis is performed. (Please note: Jupyter Notebook for this project uses Python 3.6), (Please note the generated data for all stars' metallicity was kept at a value of 0.02).

1. Multiple python packages were imported such as NumPy, pandas, and matplotlib, which all contain libraries. These packages are used for scientific computing in Python as well as open-source data analysis.
2. All different masse's summary fields are imported, these files log bulk properties of the star over time. All different mass files are stored in their respective named variables (i.e. summary_1M, summary_2M) and named with the according column list (i.e. 'age(yr)', 'mass(solar)' etc.) (Please note the full column list is posted in appendix section).
    a. The variable code to store data for each file:
    ```
    summary_1M = pd.read_csv('summary_1M.txt', delim_whitespace=True, names=column_list)
    ```
    It is important to note that the summary file data must be stored in the same directory as the main ipynb file.
    b. A quick test can be run to see if the stored data is in the correct column as well as the right variable. This can be done by checking to see at how long the stellar evolution model ran, by printing the maximum age described in the summary file for any of the star's summaries:
    ```
    print('Maximum age of 1 solar mass star model = ', summary_1M['age(yr)'].max(), ' years')
    ```
        i. Maximum age of 1 solar mass star model = 12302476200.0 years
3. Then using the functions from the libraries imported, the luminosity and the effective temperature of the stars can be plotted, with the age bar to determine when it leaves the main sequence. To do this the following code below can be used as a general form (Sample code for 1M (Solar mass) star:

```
fig,ax = plt.subplots(figsize=(10,10))
#use c=summary['age(yr)'] to color the plot points by the age of the star.
myplot = ax.scatter(summary_1M['logTsurf(K)'], summary_1M['logL(solar)'], marker='o', c=summary_1M['age(yr)'])
plt.colorbar(myplot)

#Set my axis labels, and reverse the x axis so that temperature increases to the left.
ax.set_xlabel('$\log{T_{eff}} (K)$', fontsize=14)
ax.set_ylabel('$\log{L} (L_{\odot})$', fontsize=14)
ax.invert_xaxis()

#Note that both temp and luminosity are already in log space in our file, so I don't need to rescale the axes.
plt.show()

#For each star, determine (roughly) when it leaves the main sequence,
#ie when hydrogen fusion in the core is no longer the dominant energy source
tms_1M = 0.9E10
print("1M star main sequence lifespan (yrs) ~", "{:,}".format(tms_1M), "yrs")
```

As can be seen above, this will plot a graph with the effective temperature (Teff) on the x-axis and stellar luminosity on the y axis. (Both are stored as variables using the base 10 logarithm; thus, they are ‘logTsurf’ & ‘logL’). Additionally using the last parameter stated as ‘c’, the age of the star can be plotted as a color bar with the graph to make it easier to observe and notice when the star leaves the main sequence. Note that the masses of stars used in the files are all units in the solar mass, thus having the name 0.4M, 1M, or 6M). The graph is labeled accordingly with the respective axis names and the x-axis, which is the effective temperature is inverted such that it increases to the left as it would in an HR diagram. Another variable that can be seen on the bottom block of the code is ‘tms_1M’ is the time in the main sequence of the 1M (solar mass) star with the added print statement to effectively display the relevant stars main-sequence lifetime in years. Please note the following input in the last print statement: ‘"{:,}".format(tms_1M)’ is used to format the ‘tms_1M’ variable number into a decimal form with commas, making it easier to read.

4. To have an input for the variable: ‘tms_xM’ (where ‘x’ is the number of a solar mass of the star, i.e 0.2, 1, 8), the HR diagram plotted must be examined thoroughly. For example, taking the HR plot of a 1M star results in:




```
1M star main sequence lifespan (yrs) ~ 9,000,000,000.0 yrs
```

Since it is understood that a star spends most of its life on the main sequence, this can be used to estimate appropriately when the star leaves the main sequence from the above plot with the help of the color bar posted on the right. It can be observed that as the star moves up towards the right the color changes rapidly and gradually stabilize in the age bar color (moving from green to yellow). They are being referred to as circled in red as well as the arrows pointing to the estimate are with the color bar, thus the color from that region can be used to match with the bar on the right. Using this, the number matching according to color can be selected and stored in the 'tms_xM' variable (where 'x' is the number of solar masses of the star, in this case, x = 1). This is to be repeated for all different masses of stars selected.

5. Once all main sequence lifetimes for all different masses' stars have been estimated and stored in the 'tms_xM' variables, this data along with the star masses (in solar mass) can be used to plot main sequence age vs stellar mass and can be done so using the following code:

```
#Make a plot of this main sequence age vs stellar mass
mass = [0.2, 0.4, 0.6, 1.0, 2.0, 4.0, 6.0, 8.0, 10.0]
age_ms = [tms_0_2M, tms_0_4M, tms_0_6M, tms_1M, tms_2M, tms_4M, tms_6M, tms_8M, tms_10M]

fig,ax = plt.subplots(figsize=(10,10))
myplot = ax.plot(mass, age_ms, marker='o')

plt.title("Stellar main sequence age as a function of mass")
ax.set_xlabel('$Mass (M_{\odot})$', fontsize=14)
ax.set_ylabel('Main sequence age (yrs)', fontsize=14)

plt.show()
```

Two lists can be created as such above, with all different masses (units in solar masses) and the main sequence age with the objects being the 'tms_xM' (where 'x' is the number of solar masses of the star, i.e 0.2, 1, 8) as they were stored earlier in the HR plotting code. The appropriate titles can be applied to the graph. Please note that the x-axis must be the mass list since the main sequence lifetime depends on the star's mass and thus the star's mass is the independent variable. This graph can be used to make observations regarding the effect of stellar mass and the star's main sequence phase time.

Results

Star masses used for this project: 0.2, 0.4, 0.6, 1.0, 2.0, 4.0, 6.0, 8.0 and 10.0 solar masses

The following are the HR diagrams for the above masses listed as well as their main sequence phase time stated below:

0.2M Star:

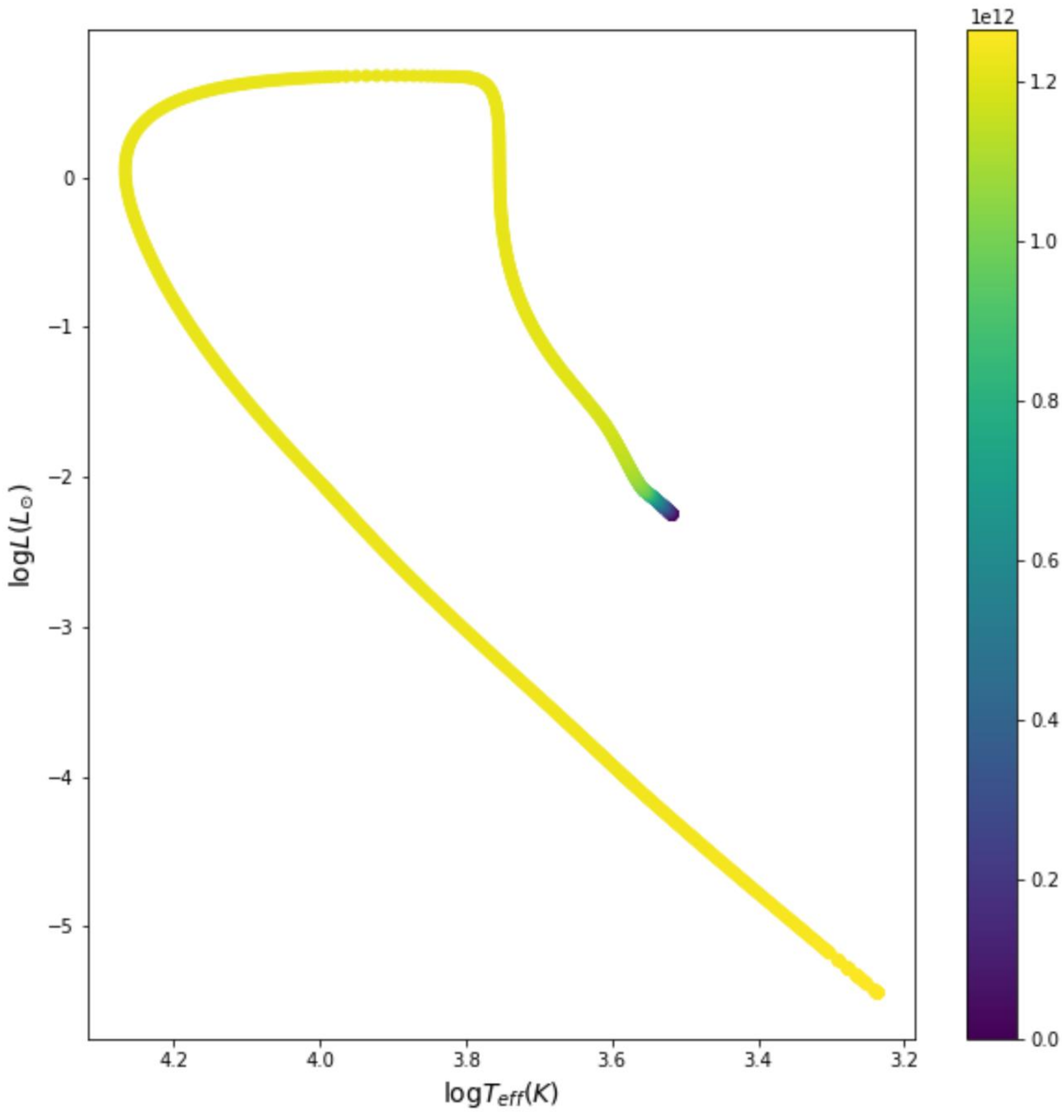


Figure 1.1: HR plot of 0.2M (solar mass) star

0.2M star main sequence lifespan (yrs) ~ 800,000,000,000.0 yrs

0.4M Star:

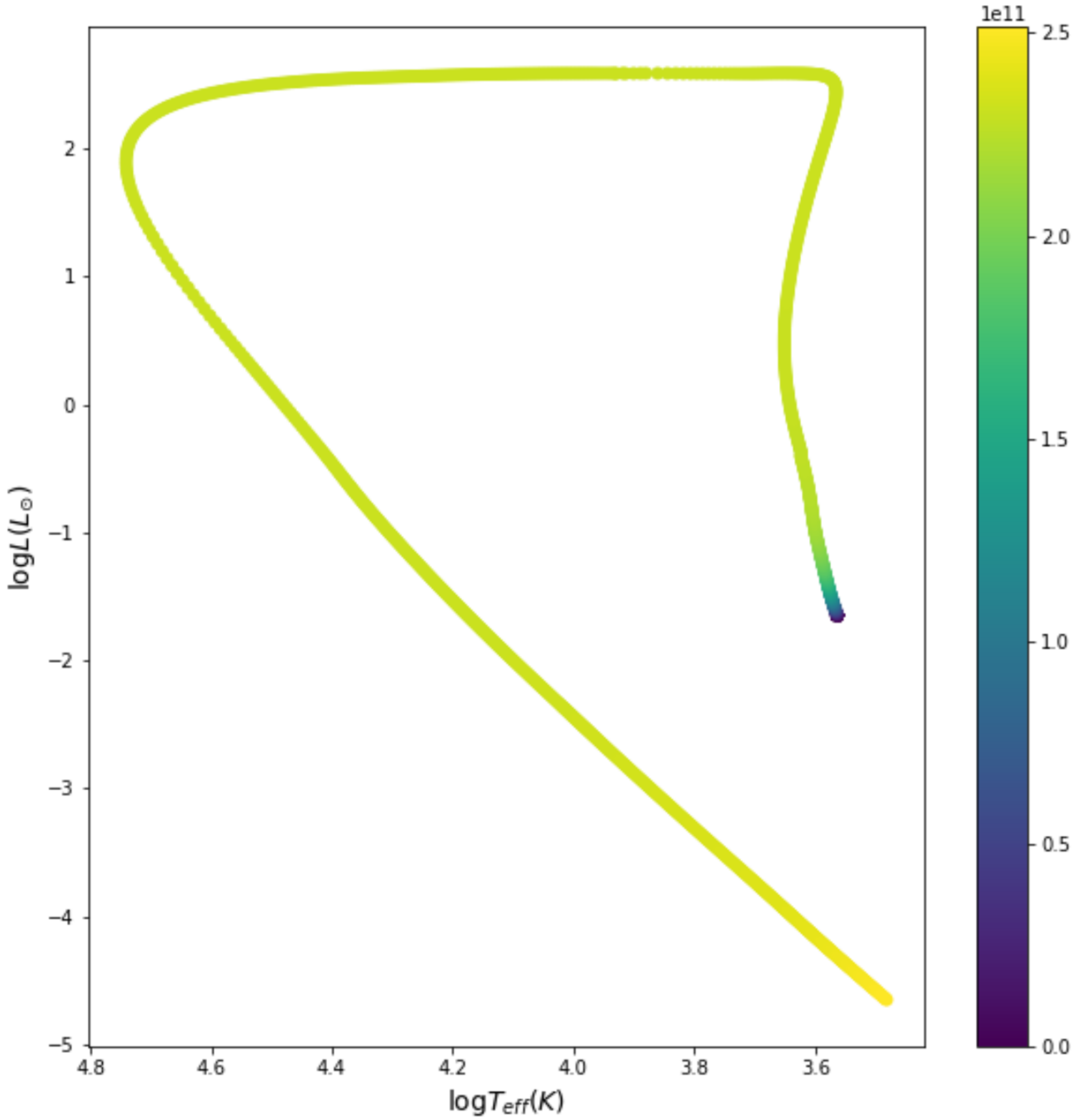


Figure 1.2: HR plot of 0.4M (solar mass) star

0.4M star main sequence lifespan (yrs) ~ 150,000,000,000.0 yrs

0.6M Star:

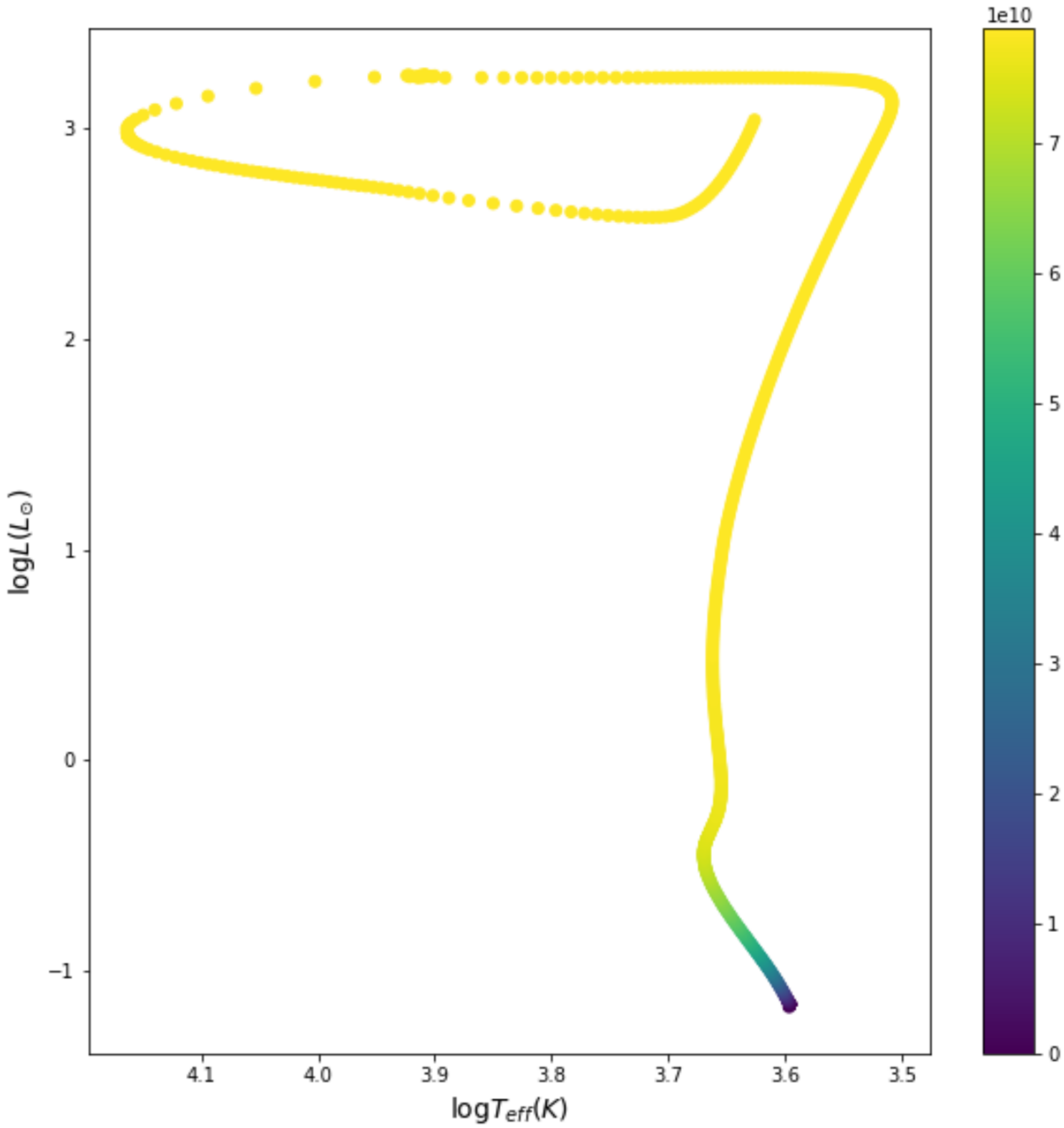


Figure 1.3: HR plot of 0.6M (solar mass) star

0.6M star main sequence lifespan (yrs) ~ 50,000,000,000.0 yrs

1M Star:

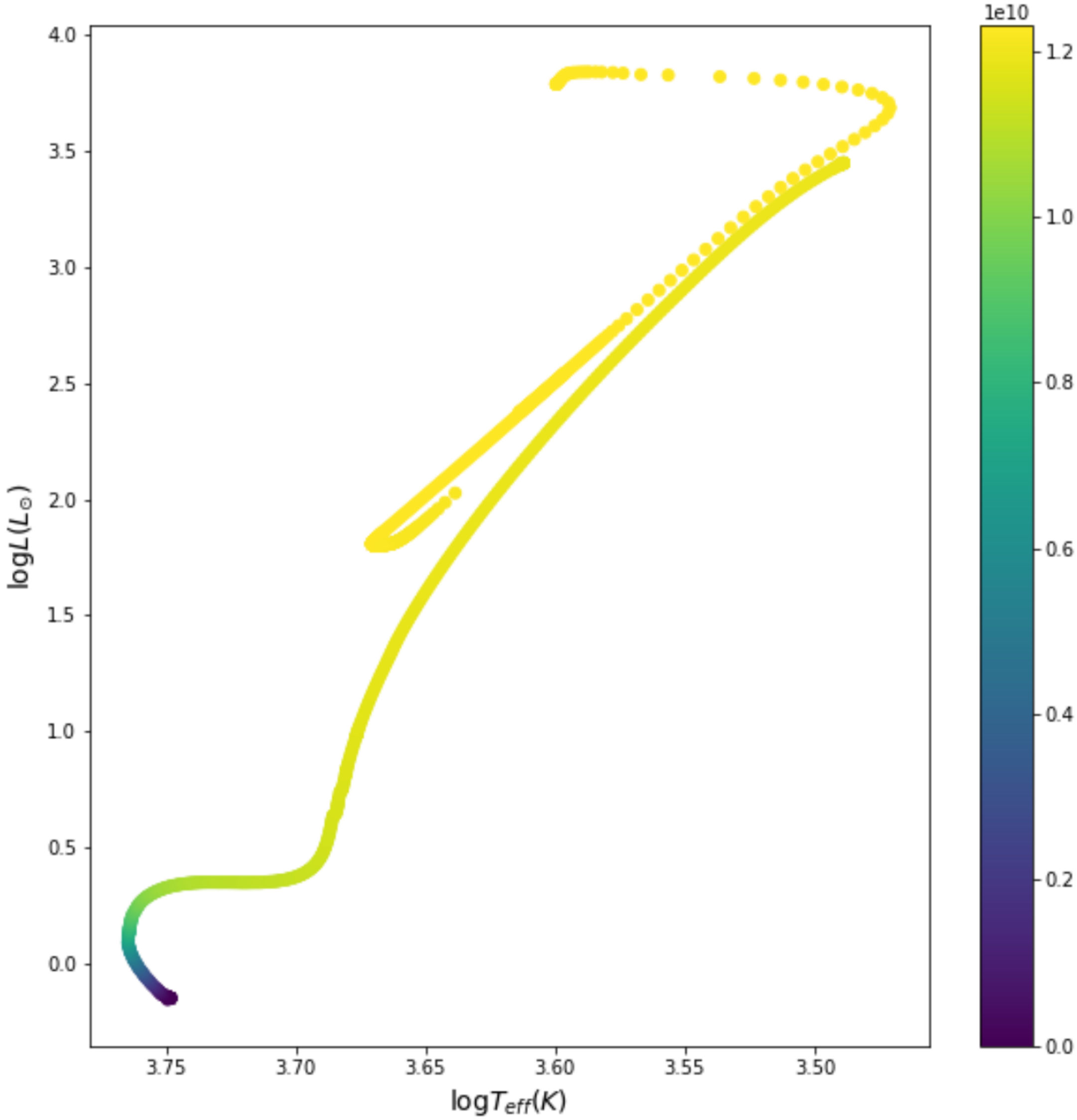


Figure 1.4: HR plot of 1M (solar mass) star

1M star main sequence lifespan (yrs) ~ 9,000,000,000.0 yrs

2M Star:

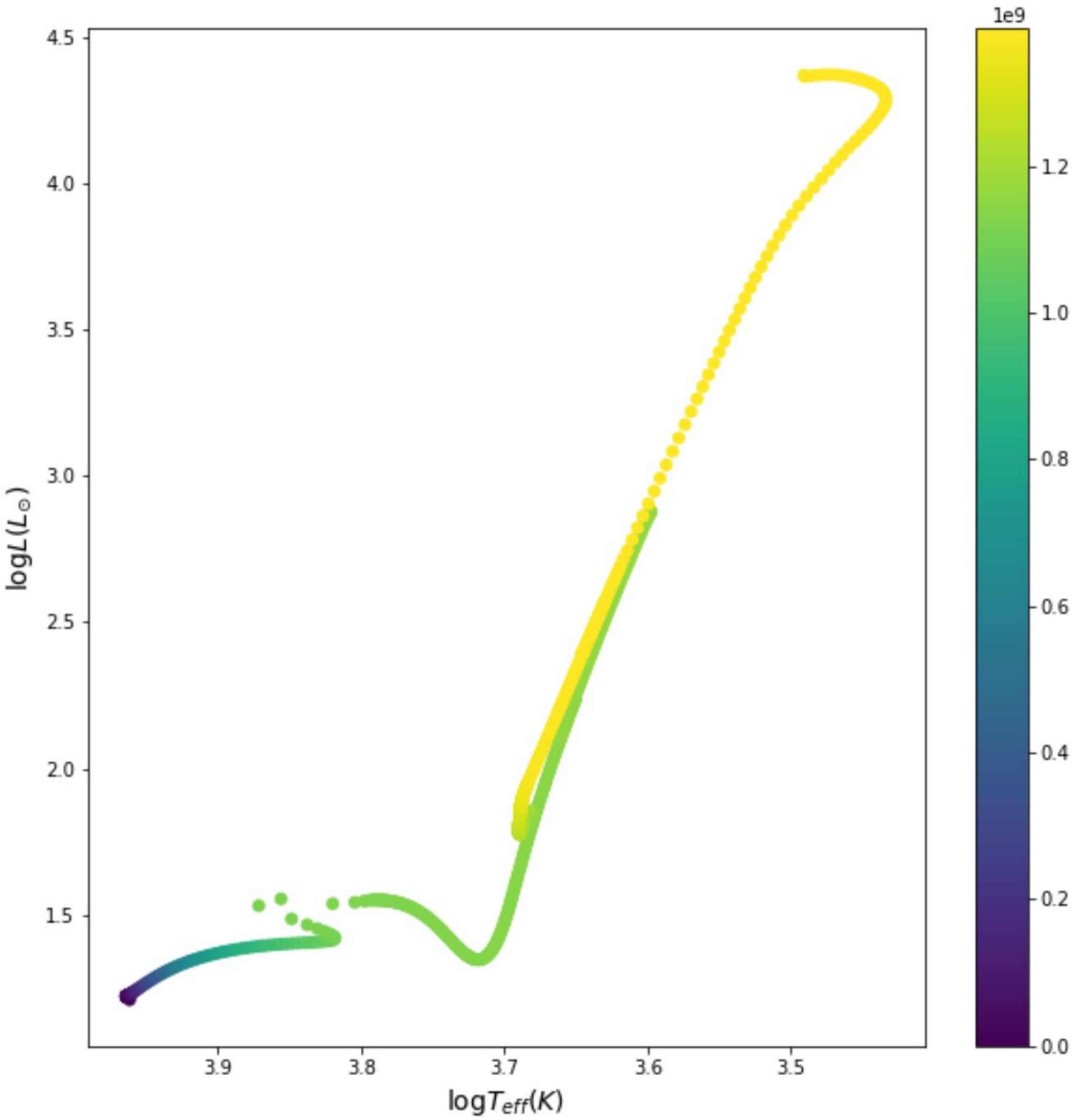


Figure 1.5: HR plot of 2M (solar mass) star

2M star main sequence lifespan (yrs) ~ 1,150,000,000.0 yrs

4M Star:

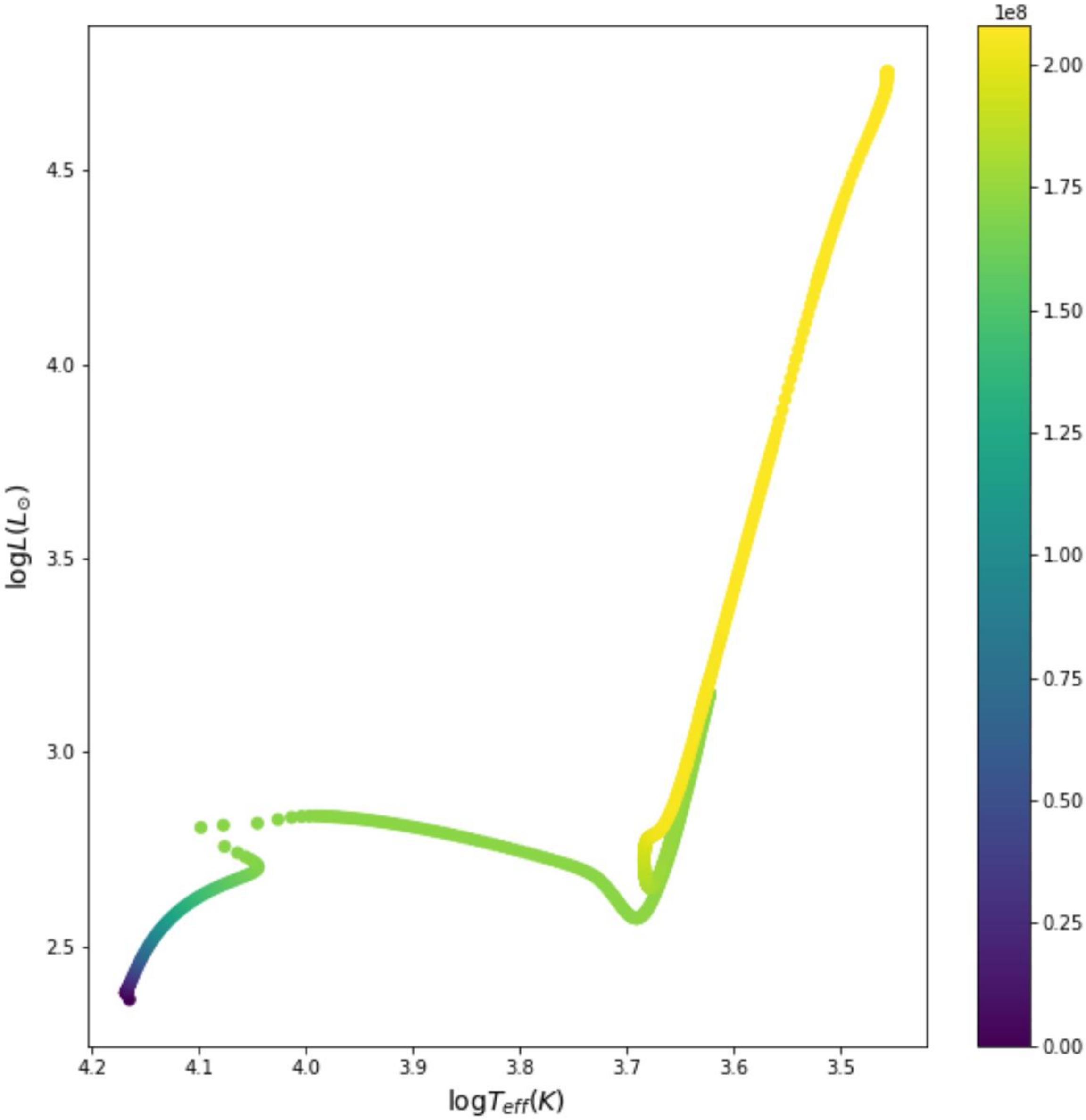


Figure 1.6: HR plot of 4M (solar mass) star

4M star main sequence lifespan (yrs) ~ 175,000,000.0 yrs

6M Star:

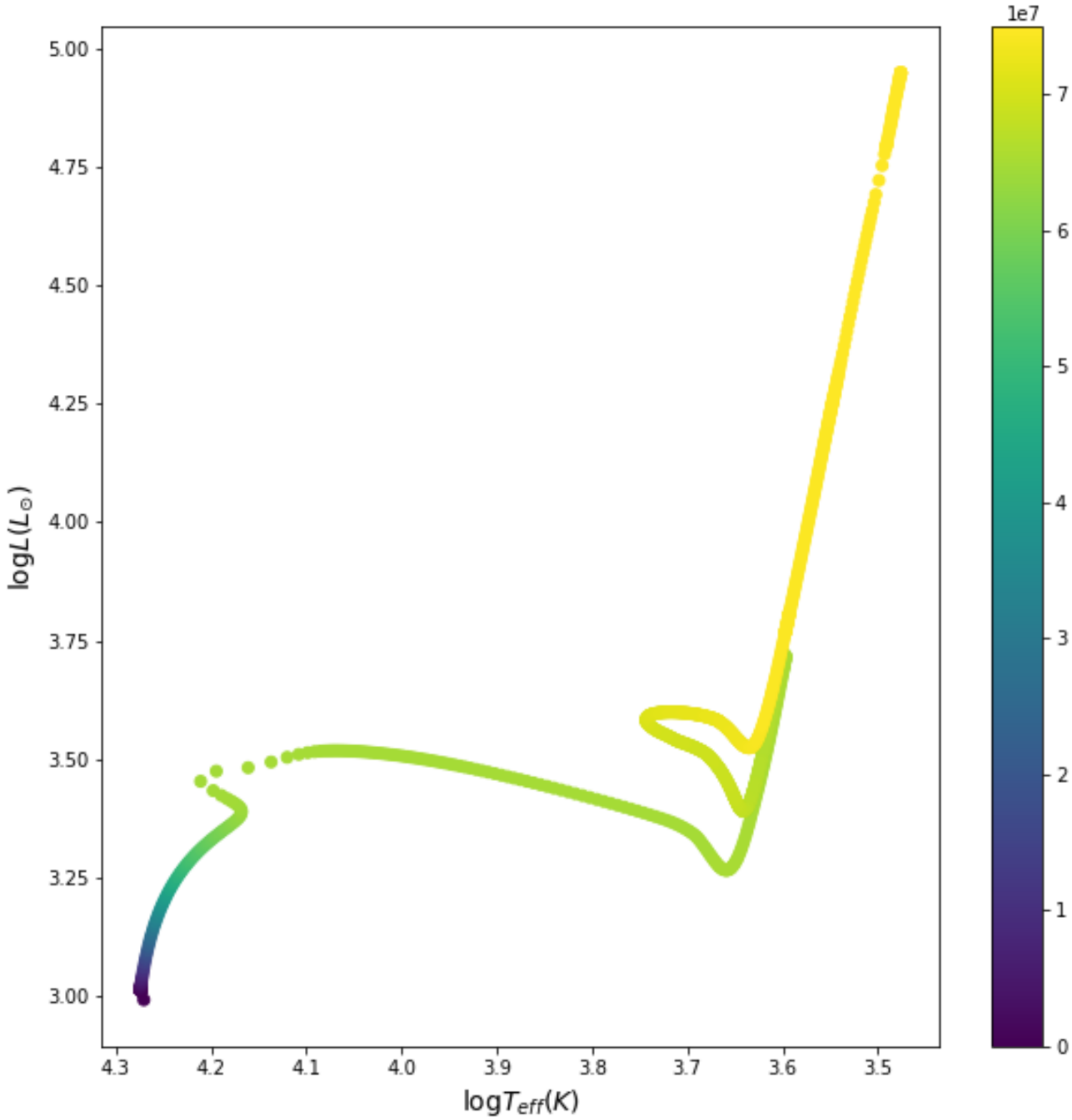


Figure 1.7: HR plot of 6M (solar mass) star

6M star main sequence lifespan (yrs) ~ 62,000,000.0 yrs

8M Star:

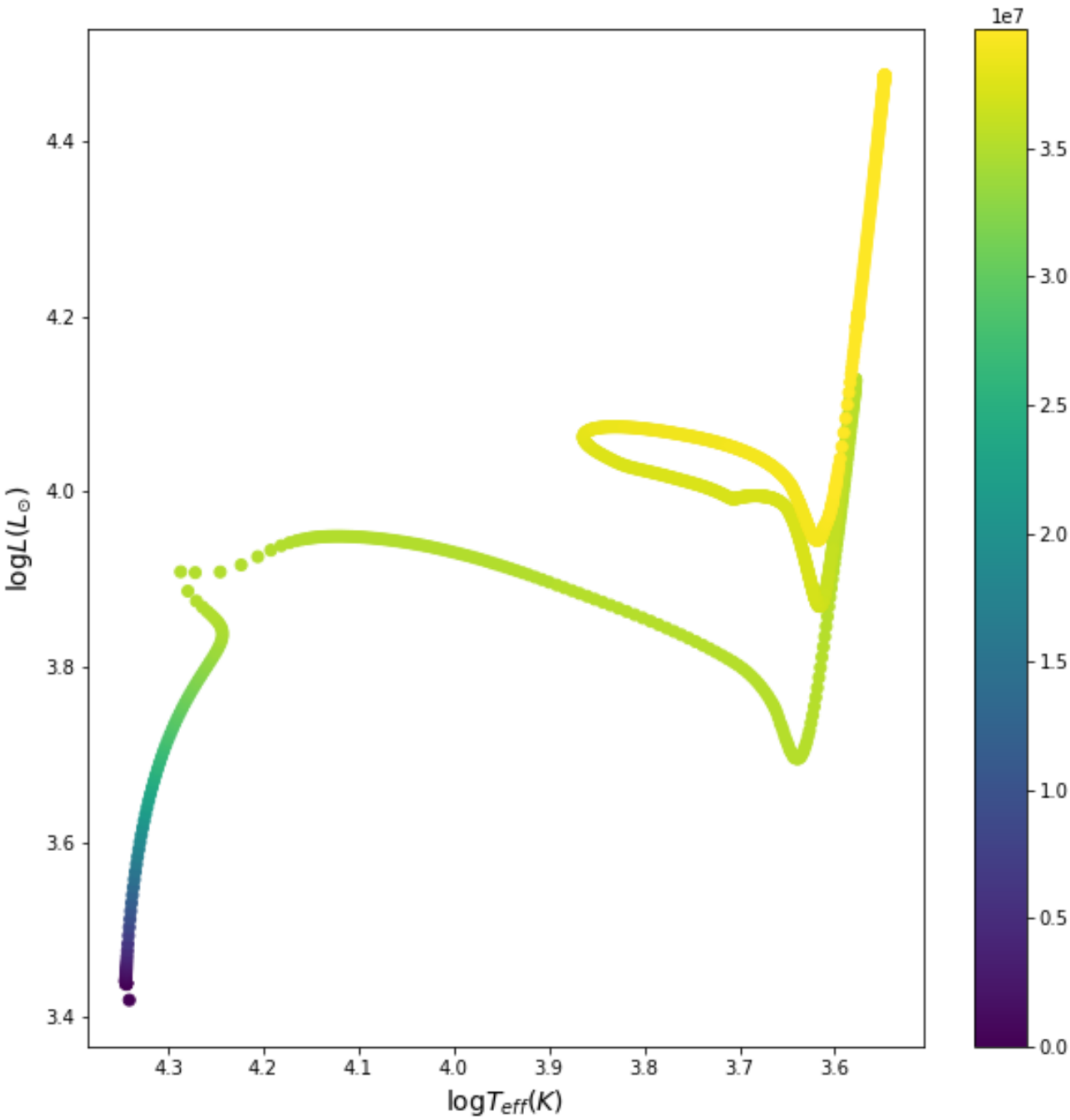


Figure 1.8: HR plot of 8M (solar mass) star

8M star main sequence lifespan (yrs) ~ 32,500,000.0 yrs

10M Star:

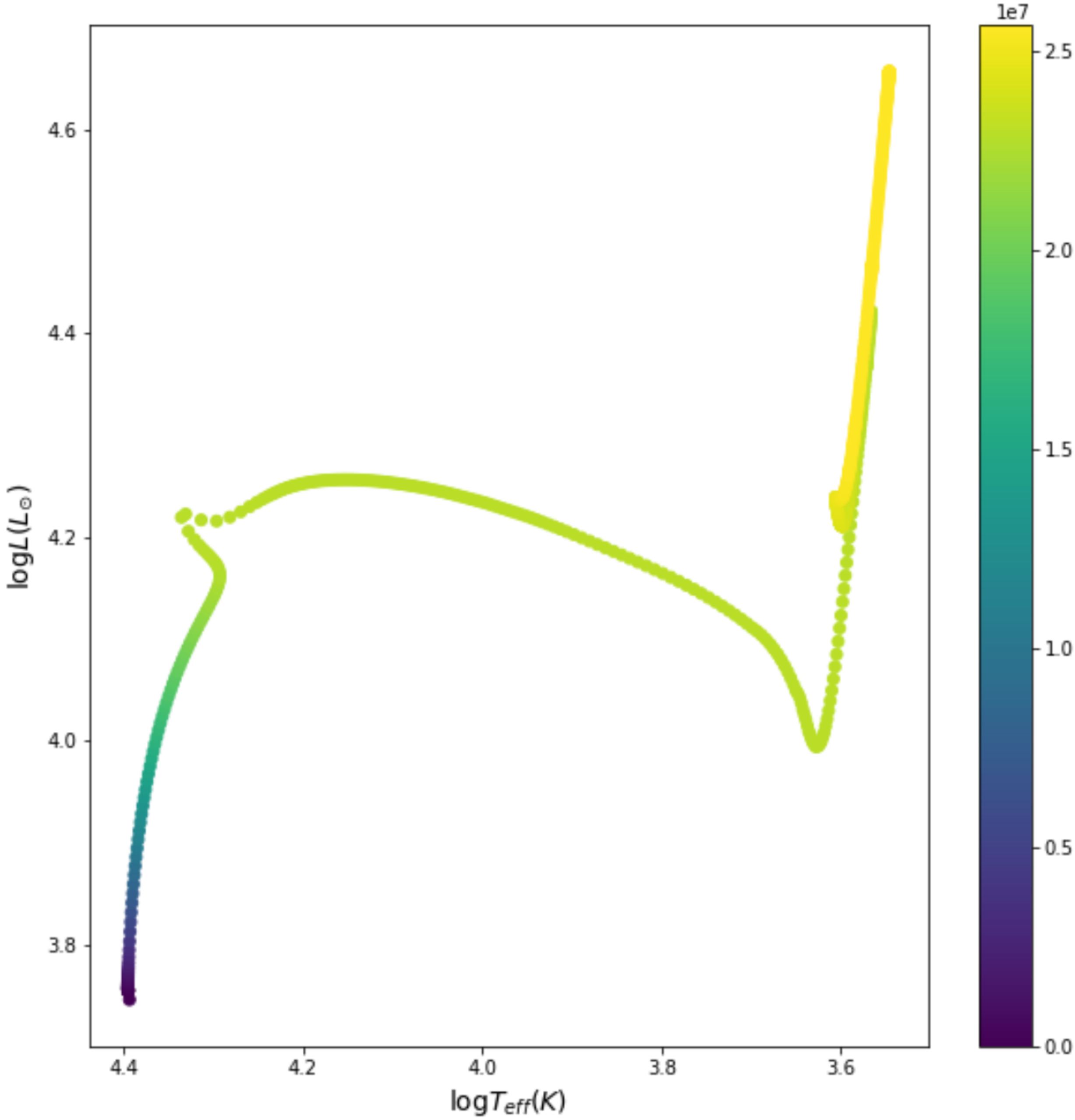


Figure 1.9: HR plot of 10M (solar mass) star

10M star main sequence lifespan (yrs) ~ 21,000,000.0 yrs

Table comparison for the stellar mass (solar mass), main sequence age from the HR plot estimate, and the approximate empirical value calculated using equation (1):

Stellar-mass ($M_{\odot}$)	**HR Diagram estimate (years)**	**Equation (1) – Calculated (years)**
0.2	800,000,000,000.00	559,016,994,374.95
0.4	150,000,000,000.00	98,821,176,880.26
0.6	50,000,000,000.00	35,860,956,909.33
1	9,000,000,000.00	10,000,000,000.00
2	1,150,000,000.00	1,767,766,952.97
4	175,000,000.00	312,500,000.00
6	62,000,000.00	113,402,302.91
8	32,500,000.00	55,242,717.28
10	21,000,000.00	31,622,776.60

Note: An interesting observation can be made here about the main sequence ages in the two columns above. For stars with mass $< 1\ M_{\odot}$, the HR diagram estimate is greater than the calculated value from equation (1). Similarly, for stars with mass $\geq 1\ M_{\odot}$, the HR diagram estimate is less than the calculated value from equation (1).

Using the collected data, the stellar main sequence age is plotted as a function of the star's mass:

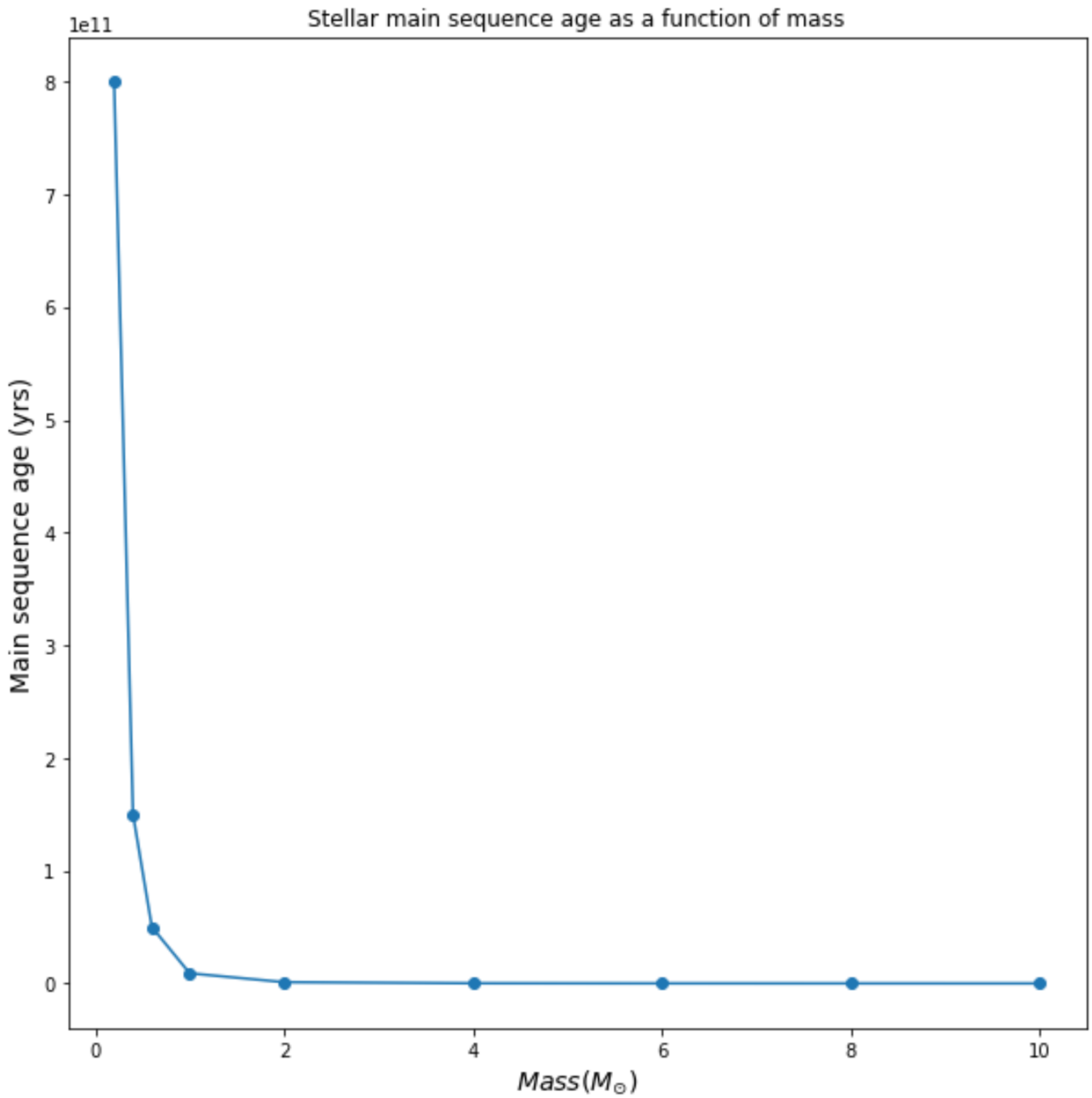


Figure 2: Plot of the stellar main sequence age as a function of the stellar mass

This type of relationship between two variables is inversely proportional. It can be observed that as the mass of the star increases it spends lesser time in its main sequence phase, more on in this in the discussion section.

Discussion and Conclusion

Using the theoretical data collected from the EZ-Web database, HR diagrams were plotted for different masses stars to determine how long each of the star's main sequence ages lasted. Using this data, the plot of stellar main sequence age as a function of the stellar mass was plotted, and it was it is observed that the relation between the two variables is inversely proportional, indicating that stars with more mass (bigger stars) spend less time during their main sequence phase, thus less massive stars (small stars) spend more time during their main sequence phase. The mass of a star is crucial because it dictates how long it can stabilize and be stable. Examining HR graphs, particularly of differing mass stars, confirms this. The most massive, brightest stars will only be in the main sequence for a few million years, whereas the least massive, faintest stars will be in the main sequence for hundreds of billions of years.

This raises the question as to why massive stars spend less time on the main sequence when they have more material to burn, and the conclusion above can be counterintuitive. Massive stars have much more hydrogen, but they exhaust it so rapidly that their lifetimes are substantially shorter than the low mass stars. A simple breakdown of this can be thought of as, since more massive stars have more mass, there will have more gravity, this will result in higher internal pressure which in turn will increase the temperature making it hotter, thus making it burn faster.

References

Data were retrieved from the following website database:

Link: http://www.astro.wisc.edu/~townsend/static.php?ref=ez-web

Townsend, Rich. "EZ-Web." *Mad Star - EZ-Web*, 4 Dec. 2021, http://www.astro.wisc.edu/~townsend/static.php?ref=ez-web.

Appendix

Full source code used for data analysis:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.interpolate import make_interp_spline, BSpline
%pylab inline
```

```
#import summary files for different masses stars
#I've used ezweb to generate stellar evolution and structure models for 9 different solar mass stars.
#The file 'summary_XXX.txt' logs bulk properties of the star over time.

column_list = ['step_number','age(yr)','mass(solar)','logL(solar)','logR(solar)', 'logTsurf(K)','logTcent(K)',
               'logrho(kg/m3)','logP(N/m2)','degen','X_hydrogen','Y_helium','X_carbon', 'X_nitrogen', 'X_oxygen',
               't_dyn(sec)', 't_KH(yr)', 't_nuc(yr)', 'lum_pp(solar)', 'lum_CNO(solar)','lum_3alpha(solar)',
               'lum_metals(solar)', 'lum_neutrinos(solar)', 'M_Hecore(solar)', 'M_Ccore(solar)','M_Ocore(solar)',
               'R_Hecore(solar)', 'R_Ccore(solar)', 'R_Ocore(solar)']

summary_0_2M = pd.read_csv('summary_0.2M.txt', delim_whitespace=True, names=column_list)
summary_0_4M = pd.read_csv('summary_0.4M.txt', delim_whitespace=True, names=column_list)
summary_0_6M = pd.read_csv('summary_0.6M.txt', delim_whitespace=True, names=column_list)
summary_1M = pd.read_csv('summary_1M.txt', delim_whitespace=True, names=column_list)
summary_2M = pd.read_csv('summary_2M.txt', delim_whitespace=True, names=column_list)
summary_4M = pd.read_csv('summary_4M.txt', delim_whitespace=True, names=column_list)
summary_6M = pd.read_csv('summary_6M.txt', delim_whitespace=True, names=column_list)
summary_8M = pd.read_csv('summary_8M.txt', delim_whitespace=True, names=column_list)
summary_10M = pd.read_csv('summary_10M.txt', delim_whitespace=True, names=column_list)
```

```
#We can also take a look at how long the stellar evolution model ran
#by checking the maximum age described in the summary file for any of the stars listed above.

print('Maximum age of 1 solar mass star model = ', summary_1M['age(yr)'].max(), ' years')
```

```
fig,ax = plt.subplots(figsize=(10,10))
#use c=summary['age(yr)'] to color the plot points by the age of the star.
myplot = ax.scatter(summary_0_2M['logTsurf(K)'], summary_0_2M['logL(solar)'], marker='o', c=summary_0_2M['age(yr)'])
plt.colorbar(myplot)

#Set my axis labels, and reverse the x axis so that temperature increases to the left.
ax.set_xlabel('$\log{T_{eff}} (K)$', fontsize=14)
ax.set_ylabel('$\log{L} (L_{\odot})$', fontsize=14)
ax.invert_xaxis()

#Note that both temp and luminosity are already in log space in our file, so I don't need to rescale the axes.
plt.show()

#For each star, determine (roughly) when it leaves the main sequence,
#ie when hydrogen fusion in the core is no longer the dominant energy source
tms_0_2M = 0.8E12
print("0.2M star main sequence lifespan (yrs) ~", f'{tms_0_2M:,}', "yrs")
```

```
fig,ax = plt.subplots(figsize=(10,10))
#use c=summary['age(yr)'] to color the plot points by the age of the star.
myplot = ax.scatter(summary_0_4M['logTsurf(K)'], summary_0_4M['logL(solar)'], marker='o', c=summary_0_4M['age(yr)'])
plt.colorbar(myplot)

#Set my axis labels, and reverse the x axis so that temperature increases to the left.
ax.set_xlabel('$\log{T_{eff}} (K)$', fontsize=14)
ax.set_ylabel('$\log{L} (L_{\odot})$', fontsize=14)
ax.invert_xaxis()

#Note that both temp and luminosity are already in log space in our file, so I don't need to rescale the axes.
plt.show()

#For each star, determine (roughly) when it leaves the main sequence,
#ie when hydrogen fusion in the core is no longer the dominant energy source
tms_0_4M = 1.5E11
print("0.4M star main sequence lifespan (yrs) ~", f'{tms_0_4M:,}', "yrs")
```

```python
fig,ax = plt.subplots(figsize=(10,10))
#use c=summary['age(yr)'] to color the plot points by the age of the star.
myplot = ax.scatter(summary_0_6M['logTsurf(K)'], summary_0_6M['logL(solar)'], marker='o', c=summary_0_6M['age(yr)'])
plt.colorbar(myplot)

#Set my axis labels, and reverse the x axis so that temperature increases to the left.
ax.set_xlabel('$\log{T_{eff}} (K)$', fontsize=14)
ax.set_ylabel('$\log{L} (L_{\odot})$', fontsize=14)
ax.invert_xaxis()

#Note that both temp and luminosity are already in log space in our file, so I don't need to rescale the axes.
plt.show()

#For each star, determine (roughly) when it leaves the main sequence,
#ie when hydrogen fusion in the core is no longer the dominant energy source
tms_0_6M = 5E10
print("0.6M star main sequence lifespan (yrs) ~", "{:,}".format(tms_0_6M), "yrs")
```

```python
fig,ax = plt.subplots(figsize=(10,10))
#use c=summary['age(yr)'] to color the plot points by the age of the star.
myplot = ax.scatter(summary_1M['logTsurf(K)'], summary_1M['logL(solar)'], marker='o', c=summary_1M['age(yr)'])
plt.colorbar(myplot)

#Set my axis labels, and reverse the x axis so that temperature increases to the left.
ax.set_xlabel('$\log{T_{eff}} (K)$', fontsize=14)
ax.set_ylabel('$\log{L} (L_{\odot})$', fontsize=14)
ax.invert_xaxis()

#Note that both temp and luminosity are already in log space in our file, so I don't need to rescale the axes.
plt.show()

#For each star, determine (roughly) when it leaves the main sequence,
#ie when hydrogen fusion in the core is no longer the dominant energy source
tms_1M = 0.9E10
print("1M star main sequence lifespan (yrs) ~", "{:,}".format(tms_1M), "yrs")
```

```python
fig,ax = plt.subplots(figsize=(10,10))
#use c=summary['age(yr)'] to color the plot points by the age of the star.
myplot = ax.scatter(summary_2M['logTsurf(K)'], summary_2M['logL(solar)'], marker='o', c=summary_2M['age(yr)'])
plt.colorbar(myplot)

#Set my axis labels, and reverse the x axis so that temperature increases to the left.
ax.set_xlabel('$\log{T_{eff}} (K)$', fontsize=14)
ax.set_ylabel('$\log{L} (L_{\odot})$', fontsize=14)
ax.invert_xaxis()

#Note that both temp and luminosity are already in log space in our file, so I don't need to rescale the axes.
plt.show()

#For each star, determine (roughly) when it leaves the main sequence,
#ie when hydrogen fusion in the core is no longer the dominant energy source
tms_2M = 1.15E9
print("2M star main sequence lifespan (yrs) ~", "{:,}".format(tms_2M), "yrs")
```

```python
fig,ax = plt.subplots(figsize=(10,10))
#use c=summary['age(yr)'] to color the plot points by the age of the star.
myplot = ax.scatter(summary_4M['logTsurf(K)'], summary_4M['logL(solar)'], marker='o', c=summary_4M['age(yr)'])
plt.colorbar(myplot)

#Set my axis labels, and reverse the x axis so that temperature increases to the left.
ax.set_xlabel('$\log{T_{eff}} (K)$', fontsize=14)
ax.set_ylabel('$\log{L} (L_{\odot})$', fontsize=14)
ax.invert_xaxis()

#Note that both temp and luminosity are already in log space in our file, so I don't need to rescale the axes.
plt.show()

#For each star, determine (roughly) when it leaves the main sequence,
#ie when hydrogen fusion in the core is no longer the dominant energy source
tms_4M = 1.75E8
print("4M star main sequence lifespan (yrs) ~", "{:,}".format(tms_4M), "yrs")
```

```python
fig,ax = plt.subplots(figsize=(10,10))
#use c=summary['age(yr)'] to color the plot points by the age of the star.
myplot = ax.scatter(summary_6M['logTsurf(K)'], summary_6M['logL(solar)'], marker='o', c=summary_6M['age(yr)'])
plt.colorbar(myplot)

#Set my axis labels, and reverse the x axis so that temperature increases to the left.
ax.set_xlabel('$\log{T_{eff}} (K)$', fontsize=14)
ax.set_ylabel('$\log{L} (L_{\odot})$', fontsize=14)
ax.invert_xaxis()

#Note that both temp and luminosity are already in log space in our file, so I don't need to rescale the axes.
plt.show()

#For each star, determine (roughly) when it leaves the main sequence,
#ie when hydrogen fusion in the core is no longer the dominant energy source
tms_6M = 6.2E7
print("6M star main sequence lifespan (yrs) ~", "{:,}".format(tms_6M), "yrs")
```

```python
fig,ax = plt.subplots(figsize=(10,10))
#use c=summary['age(yr)'] to color the plot points by the age of the star.
myplot = ax.scatter(summary_8M['logTsurf(K)'], summary_8M['logL(solar)'], marker='o', c=summary_8M['age(yr)'])
plt.colorbar(myplot)

#Set my axis labels, and reverse the x axis so that temperature increases to the left.
ax.set_xlabel('$\log{T_{eff}} (K)$', fontsize=14)
ax.set_ylabel('$\log{L} (L_{\odot})$', fontsize=14)
ax.invert_xaxis()

#Note that both temp and luminosity are already in log space in our file, so I don't need to rescale the axes.
plt.show()

#For each star, determine (roughly) when it leaves the main sequence,
#ie when hydrogen fusion in the core is no longer the dominant energy source
tms_8M = 3.25E7
print("8M star main sequence lifespan (yrs) ~", "{:,}".format(tms_8M), "yrs")
```

```python
fig,ax = plt.subplots(figsize=(10,10))
#use c=summary['age(yr)'] to color the plot points by the age of the star.
myplot = ax.scatter(summary_10M['logTsurf(K)'], summary_10M['logL(solar)'], marker='o', c=summary_10M['age(yr)'])
plt.colorbar(myplot)

#Set my axis labels, and reverse the x axis so that temperature increases to the left.
ax.set_xlabel('$\log{T_{eff}} (K)$', fontsize=14)
ax.set_ylabel('$\log{L} (L_{\odot})$', fontsize=14)
ax.invert_xaxis()

#Note that both temp and luminosity are already in log space in our file, so I don't need to rescale the axes.
plt.show()

#For each star, determine (roughly) when it leaves the main sequence,
#ie when hydrogen fusion in the core is no longer the dominant energy source
tms_10M = 2.1E7
print("10M star main sequence lifespan (yrs) ~", "{:,}".format(tms_10M), "yrs")
```

```python
#Make a plot of this main sequence age vs stellar mass
mass = [0.2, 0.4, 0.6, 1.0, 2.0, 4.0, 6.0, 8.0, 10.0]
age_ms = [tms_0_2M, tms_0_4M, tms_0_6M, tms_1M, tms_2M, tms_4M, tms_6M, tms_8M, tms_10M]

fig,ax = plt.subplots(figsize=(10,10))
myplot = ax.plot(mass, age_ms, marker='o')

plt.title("Stellar main sequence age as a function of mass")
ax.set_xlabel('$Mass (M_{\odot})$', fontsize=14)
ax.set_ylabel('Main sequence age (yrs)', fontsize=14)

plt.show()
```