

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
from sklearn.linear_model import LinearRegression
def main():
    problem_3()
    problem_4()

def problem_4():
    # Import df
    gcp = pd.read_csv('GCP.csv')
    #print(gcp.head())

    # Scale to ppm
    gcp['F_land'] = gcp['F_land'] / 2.13
    gcp['F_LUC'] = gcp['F_LUC'] / 2.13
    gcp['E'] = gcp['E'] / 2.13

    # PART A: Calculate linear fit
    x = np.array(gcp['cCO2'].values).reshape((-1, 1))
    y_land = np.array(gcp['F_land'].values)
    y_luc = np.array(gcp['F_LUC'].values)

    # fit linear model
    model_land = LinearRegression().fit(x, y_land)
    model_luc = LinearRegression().fit(x, y_luc)

    r_sq_land = model_land.score(x, y_land)
    print(f"R^2 land: {r_sq_land}")
    r_sq_luc = model_luc.score(x, y_luc)
    print(f"R^2 luc: {r_sq_luc}")

    plt.subplot(211)
    plt.plot(gcp['cCO2'], gcp['F_land'], 'o', c = '#66bf7d', mec =
'#404542')
    plt.plot(x, model_land.predict(x), c = 'black', label = f"Linear
fit ($R^2$ = {round(r_sq_land, 2)})") #type:ignore
    plt.ylabel('ppm') #Gt C/yr')
    plt.title('Carbon uptake from terrestrial processes')
    plt.legend()
    plt.subplot(212)
    plt.plot(gcp['cCO2'], gcp['F_LUC'], 'o', color = '#dcc37c', mec =
'#404542')
    plt.plot(x, model_luc.predict(x), c = 'black', label = f"Linear
fit ($R^2$ = {round(r_sq_luc, 2)})") #type:ignore
    plt.xlabel('Average Annual $CO_2$ concentration (ppm)')
    plt.legend()

```

```

plt.ylabel('ppm')#Gt C/yr')
plt.title('Carbon emissions from land use change')
plt.tight_layout()
plt.savefig('flux.png')
plt.close()

# Print necessary stuff for answering questions
print(f"F_land = {round(model_land.coef_[0],4)}x+
{round(model_land.intercept_,4)}")
print(f"F_luc = {round(model_luc.coef_[0],4)}x+
{round(model_luc.intercept_,4)}")
print(f"Emissions from 2015: {round(gcp[gcp['year']==2015]
['E'].values[0],4)} ppm") #type:ignore

def problem_3():
    # Problem 3

    def blackbody(wavelength, temp):
        h = 6.626e-34 # Planck constant (J s)
        k_b = 1.3806e-23 # Boltzmann constant (J/K)
        c = 3e8 # speed of light (m/s)

        numerator = 2 * math.pi * h * c * c
        denominator = pow(wavelength,5) * math.exp((h*c)/(wavelength *
k_b * temp) -1)
        return numerator / denominator

    # Wavelengths to calculate
    w = [0.1, 0.105, 0.11, 0.12, 0.13, 0.15, 0.18, 0.19, 0.2, 0.3,
0.4, 0.5, 0.6, 0.7, 0.8, 0.9,
        1, 1.5, 2, 1.5, 3, 4, 5, 6, 7, 8, 8.5, 9, 9.5, 10, 11, 12,
13, 14, 15, 16, 17, 18, 19, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65,
70, 75, 80, 85, 90, 95, 100]

    w = [i*1e-6 for i in w] # convert from micrometer to meter

    t_sun = 5780 # Kelvin
    t_atm = 255 # Kelvin

    rad_sun = []
    rad_atm = []
    for wavelength in w:
        rad_sun.append(blackbody(wavelength, t_sun))
        rad_atm.append(blackbody(wavelength, t_atm))

    #plt.figure(figsize = (10,3))
    plt.semilogx(w, rad_sun, '-', c = 'orange', label = 'Sun')
    plt.xlabel('Wavelength (m)')

```

```
plt.ylabel('Blackbody radiation (W/m^2)')
plt.title('Emission spectrum of the sun (5,780 K)')
plt.savefig('rad_sun.png')
plt.close()
#plt.subplots(212)
plt.semilogx(w, rad_atm)
plt.xlabel('Wavelength (m)')
plt.ylabel('Blackbody radiation (W/m^2)')
plt.title('Emission spectrum of the atmosphere (255 K)')
plt.tight_layout()
plt.savefig('rad_atm.png')
plt.close()
```

```
if __name__ == '__main__':
    main()
```