Math 336 Midterm

Cole Werry

March 23st 2024

1. [25 points]
(a) [10 points] Figure 1 shows a simple pendulum with mass m, string length l, and the Earth gravitational acceleration g. Use dimensional analysis method to determine the period τ of the pendulum as a function of l, g, determined up to a constant α, i.e.,$\tau = \alpha l^a g^b$.

Proof. The book tells us that: In general, a physical quantity X can be written as the product of powers of all the k relevant quantities $X_1, X_2, \dots, X_k$ as follows:

$$X = \alpha X_1^{n_1} X_2^{n_2} X_3^{n_3} \dots X_k^{n_k},$$

where α is a dimensionless constant. Thus we can assume that the equation for this pendulum with mass m, string length l, and the Earth gravitational acceleration g will look something like:

$$\tau = \alpha m^{n_1} l^{n_2} g^{n_3} \text{ or } \tau = \alpha m^a l^b g^c$$

Using dimensional analysis

$$[\tau] = [\alpha][m]^a[l]^b[g]^c = M^a L^b (LT^{-2}) = M^a L^{b+c} T^{-2c}$$

Because we are looking for the period $[\tau] = T$

$$M^0 L^0 T^1 = M^a L^{b+c} T^{-2c}$$

This produces a system of equations for the exponents which I will use python to solve.

$$a = 0,$$
$$b + c = 0,$$
$$-2c = 1$$

```
#Python code for System of equations for pendulum problem
import sympy as sp

a, b, c = sp.symbols('a b c')


equations = [
    sp.Eq(a, 0),
    sp.Eq(b + c, 0),
    sp.Eq(-2*c, 1)
]

solution = sp.solve(equations, (a, b, c))


print("Solution:")
```

```
print("a =", solution[a])
print("b =", solution[b])
print("c =", solution[c])
```

This code tells me that $a = 0, b = \frac{1}{2}, andc = -\frac{1}{2}$. Plugging in these values to our original equation

$$\tau = \alpha m^0 l^{\frac{1}{2}} g^{-\frac{1}{2}} = \alpha\sqrt{\frac{l}{g}}$$

□

(b) [7 points] Use the conservation law of energy to find an approximate value of α in Part (a) under the condition of $\sin(x) \approx x$ when x, measured in radian (not degree), is small and close to zero.

Proof. The book tells us that if we use the lowest position of the pendulums mass we get only Kinetic energy. Thus $E_K = (\frac{1}{2}mv^2)$. At the highest point of the pendulums mass the velocity is zero. Thus the potential energy is $E_P = mgh$". Let θ_m denote a sine function that models the periodic motion of the pendulum.

$$\theta = \theta_m \sin(\tfrac{2\pi t}{\tau} + \tfrac{\pi}{2})$$

When time $t = 0$ the pendulum is now at the highest point thus $\theta = \theta_m$. When $t = \frac{\tau}{4}$ (a quarter of a period) the pendulum is now at its lowest point thus $\theta = 0$. We can now calculate the kinetic energy at the lowest point

$$E_K = (\tfrac{1}{2})mv^2 = (\tfrac{1}{2})m(l\tfrac{d\theta}{dt})^2$$

Taking the derivative of θ and looking at when $t = \frac{\tau}{4}$ we get the following:

$$E_K = \tfrac{2mgl\pi^2\theta_m^2}{\alpha^2}$$

Now calculating the potential energy at the pendulums highest height.

$$E_P = mgh = mg(l - l\cos(\theta_m)) = 2mgl\sin^2(\tfrac{\theta_m}{2})$$

We know that the maximum angle of θ_m is going to be really small so we will take the one that the book suggests which is $0.3 = 17°$. Since $\sin(0.3) = 0.2955202 \approx 0.3$ the relative error of this will be less than 2%. Therefore

$$E_P = \tfrac{mgl\theta_m^2}{2}$$

The energy conservation law tells us that $E_K = E_P$, thus

$$\tfrac{2mgl\pi^2\theta_m^2}{\alpha^2} = \tfrac{mgl\theta_m^2}{2}$$

Simplifying this we find that $\alpha = 2\pi$. Plugging this back into our original equation:

$$\tau = 2\pi\sqrt{\frac{l}{g}}$$

□

(c) [4 points] If the string length is increased to $1.005l$ due to expansion in a higher temperature environment, and its corresponding period is denoted by τ_2. Express τ_2 in terms of τ found in Part (a) of this problem, i.e.,

$$\tau_2 = k\tau$$

Find the numerical value of k and keep six (6) decimal places.

Proof. From part (a) we know that

$$\tau = 2\pi\sqrt{\tfrac{l}{g}}$$

Solving for l

$$l = \left(\tfrac{\tau^2}{4\pi^2}\right) g$$

Now increasing our string length to $1.005l$ and substituting l

$$\tau_2 = 2\pi\sqrt{\frac{1.005\left(\frac{\tau^2}{4\pi^2}\right) g}{g}} \tag{1}$$

Simplifying our equation

$$\begin{aligned}
\tau_2 &= 2\pi\sqrt{1.005\left(\frac{\tau^2}{4\pi^2}\right)} && (2)\\
&= 2\pi\frac{\sqrt{1.005}}{2\pi} && (3)\\
&= \sqrt{1.005}\,\tau && (4)
\end{aligned}$$

Thus $\tau_2 = \sqrt{1.005}\,\tau$

□

(d) [4 points] Given that $l = 60[cm]$ and $g = 9.8[m/s^2]$, calculate τ with unit in second. Write down your formulas and steps. You can use a calculator or Python to do the calculation.

Proof. Python code to solve for τ for the given length l and gravitational acceleration g:

```
#Period of the pendulum
import math

l_cm = 60
g = 9.8

l_m = l_cm / 100

tau = 2 * math.pi * math.sqrt(l_m / g)

print("Period of the pendulum:", tau, "seconds")
```

This tells us that: $\tau = 1.554685169343615 seconds$ □

2. [20 points]
(a) [17 points] The frequency model for a string music instrument: A string music instrument, such as a piano or a guitar, makes music sound from its string vibration. The music frequency is mainly determined by three factors: length of the string l (dimension: L), the linear density of the string ρ (dimension: ML^{-1}), and tension force of the string F (dimension: MLT^{-2}). The first two, l and ρ, are fixed for a given instrument, and the last one F varies when being played. A player applies different tension to make music of different frequencies. Assume that the frequency depends on l, ρ, and T in the following way

$$f = \alpha l^a \rho^b F^c$$

where α, a, b and c are constants. Use dimensional analysis to determine values of a, b and c.

Proof. Using the dimensions given to us:

$$T^{-1} = \alpha L^a (ML^{-1})^b (MLT^{-2})^c$$

This yields the following system of equations:

$$-2c = -1$$
$$b + c = -\tfrac{1}{2}$$
$$a - b + c = 0$$

Its pretty clear to see the answer but I wrote a code to double check my work as well

```
#System of equations for the frequency model for a string music instrument
import sympy as sp

a, b, c = sp.symbols('a b c')

T, L, M = sp.symbols('T L M')
equation = sp.Eq(1/T, * L**a * (M*L**-1)**b * (M*L*T**-2)**c)

eq1 = sp.Eq(-1, -2*c)
eq2 = sp.Eq(0, b + c)
eq3 = sp.Eq(0, a - b + c)

solutions = sp.solve((eq1, eq2, eq3), (a, b, c))

print("Solutions:")
print("a =", solutions[a])
print("b =", solutions[b])
print("c =", solutions[c])
```

This code will tell us that $a = -1, b = -\frac{1}{2},\ and\ c = \frac{1}{2}$. Thus,

$$f = \alpha l^{-1} p^{-\frac{1}{2}} F^{\frac{1}{2}}$$

□

3. [20 *points*]
(a) [11 points] Derive a formula for the monthly mortgage payment x, expressed in terms of the principal amount P, monthly interest rate r, and the total number of months of the loan n. Show your work and the detailed steps. The answer for this step is a formula.

Proof. The book tells us that the mathematical model for this problem will be similar to other ones we have done with energy and momentum. Thus we can assume that the formula for the first months payment will look something like:

$$P_1 = P(1 + r) - x$$

Writing out the first few monthly payments that will be made:

$$P_2 = P_(1 + r)^2 - (1 + r)x - x$$
$$P_3 = P(1 + r)^3 - (1 + r)^2 x - (1 + r)x - x$$

We can see that for the kth month the formula will look something like:

$$P_k = P(1 + r)^k - (1 + r)^{k-1}x - \cdots - (1 + r)^2 x - (1 + r)x - x$$

Using the formula for the geometric series ($\sum_{k=0}^{n} ar^k = a + ar + ar^2 + \ldots + ar^n = \frac{a(1-r^{n+1})}{1-r}$) we can simplify this equation

$$P_k = P(1+r)^k + x\frac{1-(1+r)^k}{r}$$

The payment is over when $k = n$ thus $P_n = 0$

$$P(1+r)^n + x\frac{1-(1+r)^k}{r} = 0$$

Solving for x

$$x = \frac{P(1+r)^n r}{(1+r)^n - 1}$$

□

(b) [3 points] Given the data: The principal amount (i.e., the total loan) is $P = \$500,000$,the annual interest rate (i.e., Annual Percentage Rate APR) is 6.5% (converted into the monthly rate $r = 6.5\% \div 12$), and the loan is to be paid off in 30 years (equivalent to 360 months). Use the above-derived formula and the data to compute the monthly mortgage payment x with a calculator or Python. The answer should be an amount of money per month.

Proof. Python code for mortgage payments:

```
#Monthly mortgage payments
P = 500000
annual_interest_rate = 6.5
monthly_interest_rate = annual_interest_rate / 12 / 100
n = 30 * 12

x = (P * (1 + monthly_interest_rate)**n * monthly_interest_rate) / ((1 +
                                               monthly_interest_rate)**n - 1)

print("The monthly mortgage payment will approximately be: \$", round(x, 2))
```

This code tells me that the monthly mortgage payment will approximately be: $ 3160.34

□

(c) [3 points] If the annual rate is reduced to 6.125% in the above data, what is the monthly mortgage payment?

Proof. If the annual rate is reduced to 6.125% the new monthly mortgage payment will approximately be: $3,038.05

□

(d) [3 points] If the principal is increased to $P = \$550,000$, the annual rate is 6.125%, and the loan period is still 30 years, what is the monthly mortgage payment now?

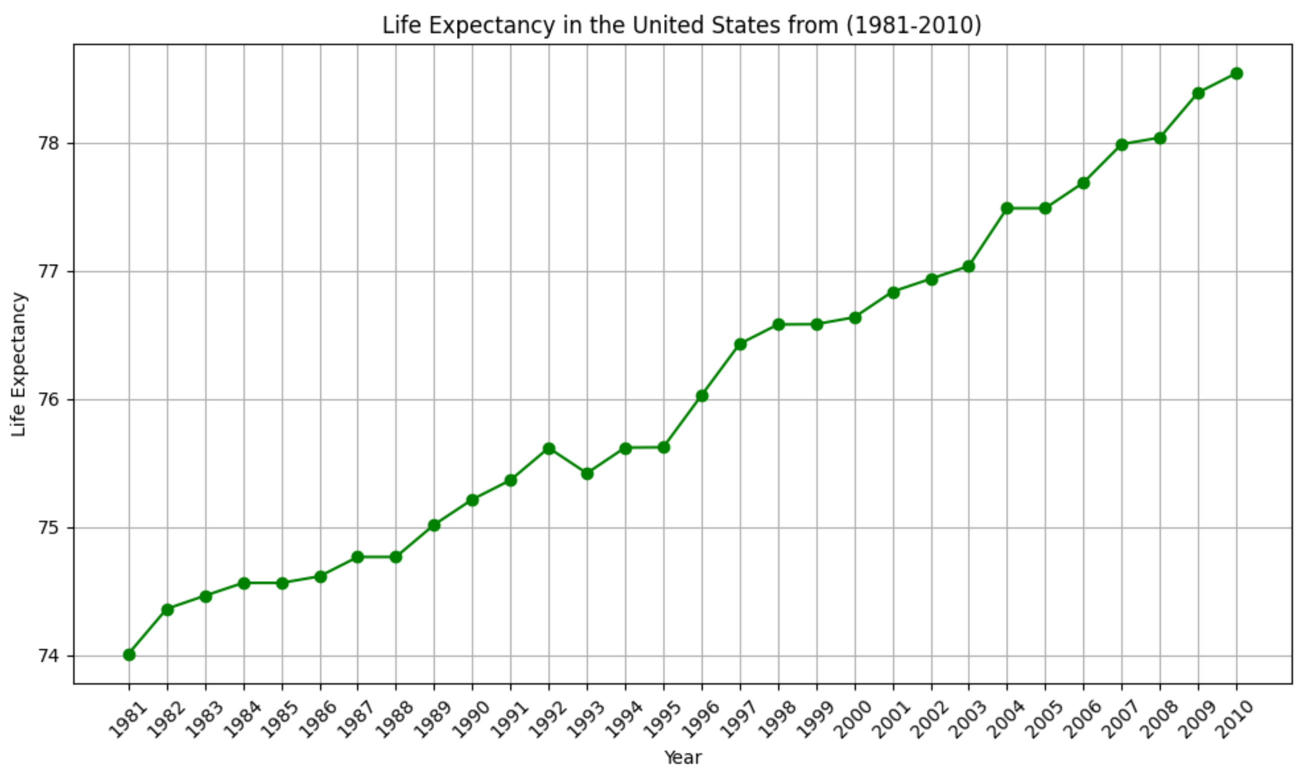
Proof. If the principal is increased to $P = \$550,000$, the annual rate is 6.125%, and the loan period is still 30 years. The new monthly mortgage payment will approximately be: $3341.86

□

4. [35 *points*] **The Python programming part**
(i) [13 points] The life expectancy data file named `LifeExpectancy19602020.csv` is given in this midterm on Canvas.
(a) [8 points] Use Python to plot the life expectancy data of the United States against time from 1981 to 2010, i.e., the vertical axis is life expectancy and the horizontal axis is time in years from 1981 to 2010.

Proof. Here is the Python code that I wrote:

```
#Life Expectrancey in the US
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('LifeExpectancy19602020.csv')

us_data = df[df['Country Name'] == 'United States']
us_data = us_data.set_index('Country Name')
us_data = us_data.loc[:, '1981':'2010']

plt.figure(figsize=(10, 6))
plt.plot(us_data.columns, us_data.values.flatten(), marker='o', linestyle='-'
                                                    , color = 'g')
plt.title('Life Expectancy in the United States from (1981-2010)')
plt.xlabel('Year')
plt.ylabel('Life Expectancy')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

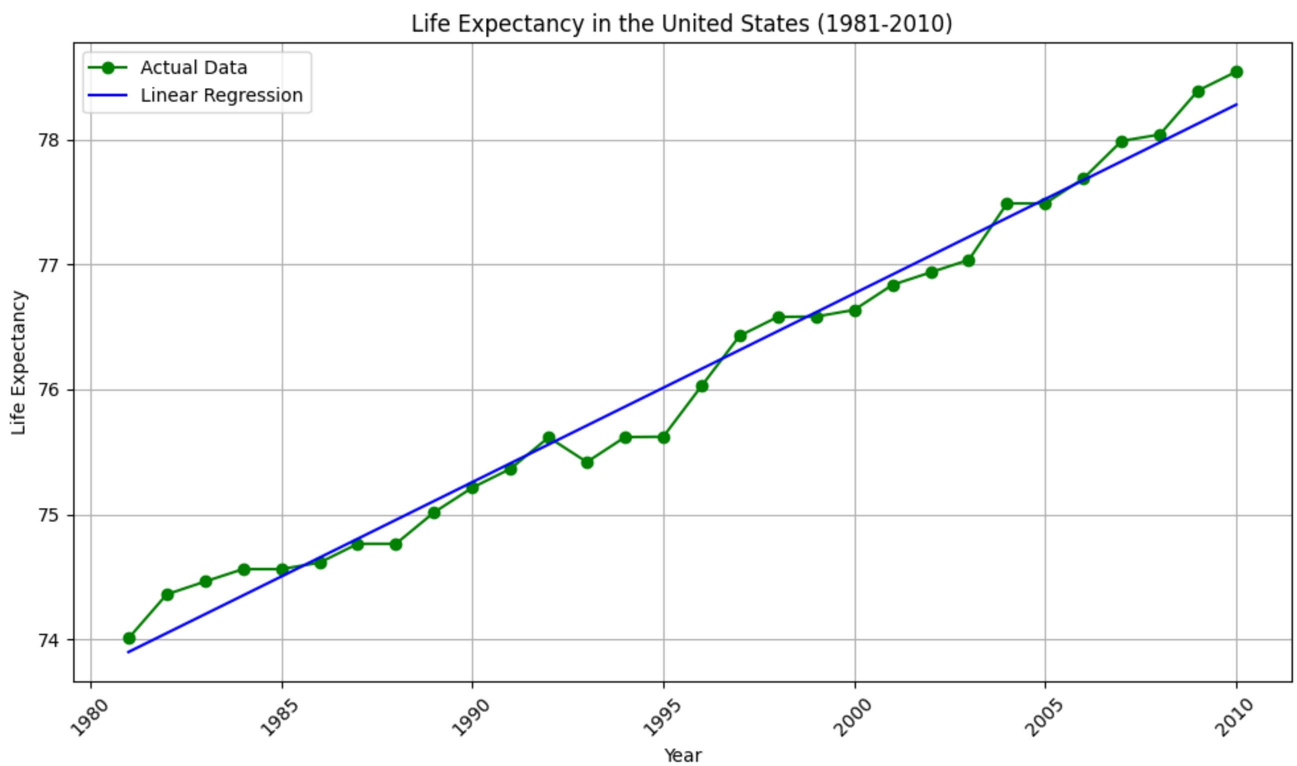


□

(b) [5 points] Use Python and linear regression to find the change rate of life expectancy for the United States during the period of 1981 - 2010. Use the units [years per decade] in your answer. Copy your Python solution results to your Python code as comment lines after #.

Proof. Here is the Python code that I wrote:

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

df = pd.read_csv('LifeExpectancy19602020.csv')

us_data = df[df['Country Name'] == 'United States']
us_data = us_data.set_index('Country Name')
us_data = us_data.loc[:, '1981':'2010']

years = np.array(us_data.columns, dtype=np.float64).reshape(-1, 1)
life_expectancy = us_data.values.flatten()

model = LinearRegression()
model.fit(years, life_expectancy)

rate_per_year = model.coef_[0]

rate_per_decade = rate_per_year * 10

predicted_life_expectancy = model.predict(years)

plt.figure(figsize=(10, 6))
plt.plot(years, life_expectancy, color='g', label='Actual Data', marker='o')
plt.plot(years, predicted_life_expectancy, color='b', label='Linear
                                         Regression')
plt.title('Life Expectancy in the United States (1981-2010)')
plt.xlabel('Year')
plt.ylabel('Life Expectancy')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
print("The change rate of life expectancy in the United States from 1981 to
                                         2010 is: {:.2f} years per decade".
                                         format(rate_per_decade))
```



The change rate of life expectancy in the United States from 1981 to 2010 is: 1.51 years per decade

□

(ii) [22 points]
Use Python and the given dataset `EarthTemperatureData.txt` downloadable from the Canvas' Assignment/Midterm block to plot a figure.

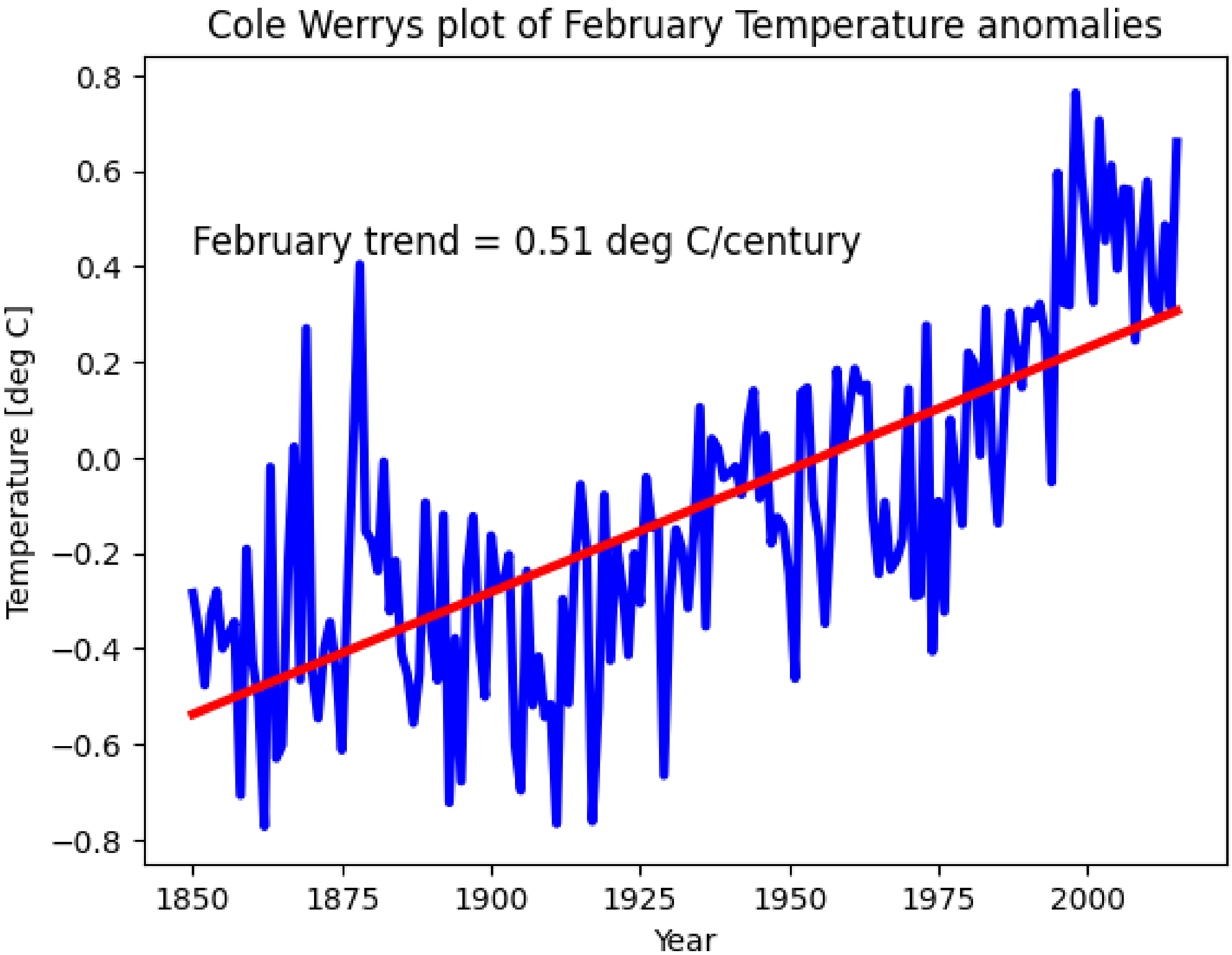
Proof. Python code for changing the txt file to an excel file:

```
#Convert txt file to an Excel file
import pandas as pd

with open("EarthTemperatureData.txt", "r") as file:
    data = file.readlines()

formatted_data = []
for line in data:
    line = line.strip().split()
    formatted_data.append(line)


df = pd.DataFrame(formatted_data, columns=["YEAR", "JAN", "FEB", "MAR", "APR"
                                          , "MAY", "JUN", "JUL", "AUG", "SEP", "
                                          OCT", "NOV", "DEC", "ANNUAL"])

df.to_excel("new_formatted_data.xlsx", index=False)
```

Now its very easy to read the data

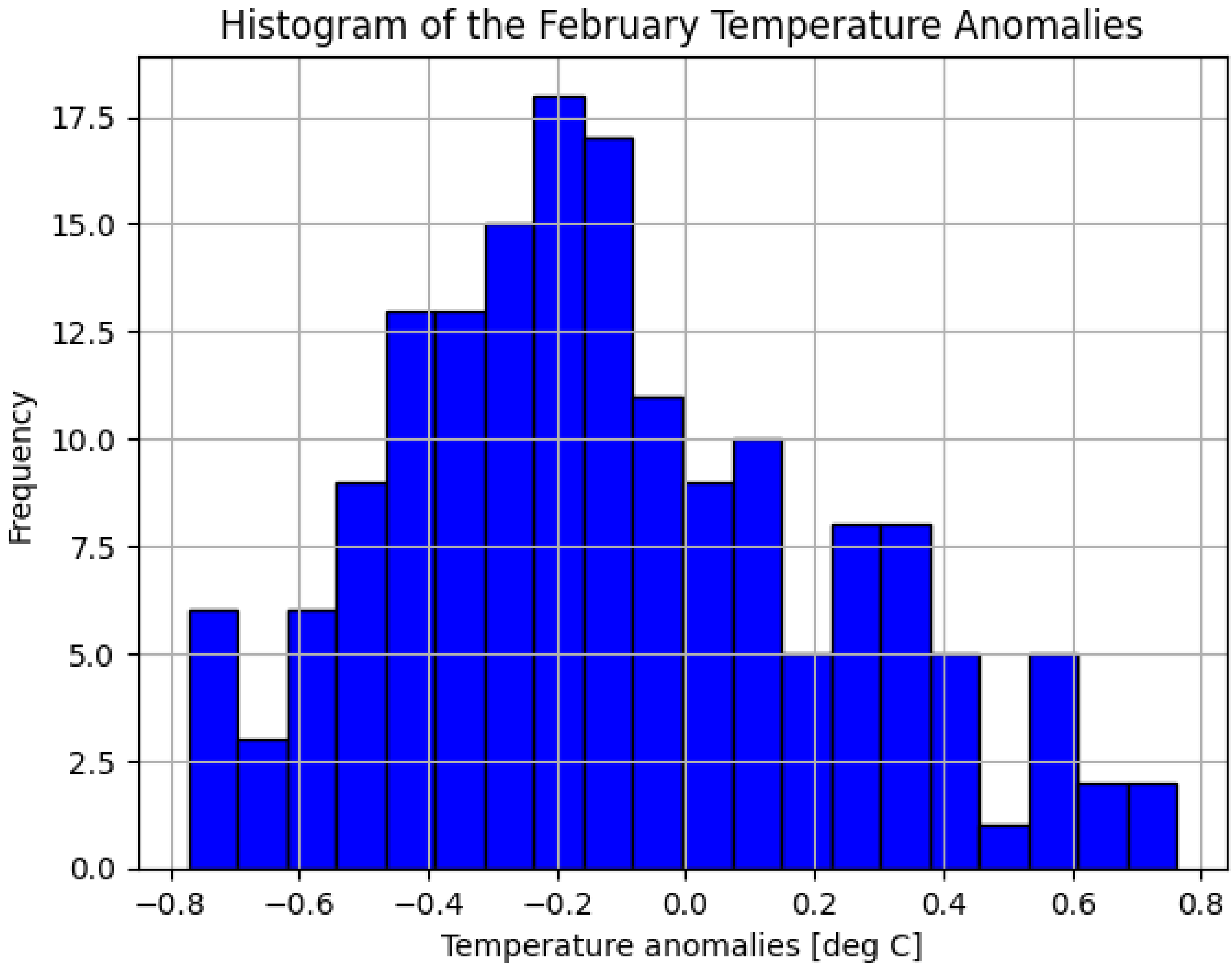
```
#Plot for Feburary temperature anomalies
import pandas as pd
```

```
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

file_path = "new_formatted_data.xlsx"
data = pd.read_excel(file_path)

years = data['YEAR'].values.reshape(-1, 1)
february_temperatures = data['FEB'].values

centuries = (years - 1850) / 100

model = LinearRegression()

model.fit(centuries, february_temperatures)

predicted_february_temperatures = model.predict(centuries)

plt.plot(years, february_temperatures, color='blue', label='February
                                       Temperatures', linewidth=3)
plt.plot(years, predicted_february_temperatures, color='red', linewidth=4)

slope_text = f"February trend = {model.coef_[0]:.2f} deg C/century"
plt.text(1850, min(february_temperatures) + 1.2, slope_text, fontsize=12,
                                       color='black', ha='left')

plt.xlabel('Year')
plt.ylabel('Temperature [deg C]')
plt.title('Cole Werrys plot of February Temperature anomalies')

plt.show()

print("Linear Trend:", model.coef_[0])
```



Python code for the histogram:

```
#Histogram plot
import pandas as pd
import matplotlib.pyplot as plt

file_path = "new_formatted_data.xlsx"
data = pd.read_excel(file_path)

february_temperatures = data['FEB'].values

plt.hist(february_temperatures, bins=20, color='blue', edgecolor='black')
plt.title('Histogram of the February Temperature Anomalies')
plt.xlabel('Temperature anomalies [deg C]')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```


Histogram of the February Temperature Anomalies
Frequency
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
Temperature anomalies [deg C]

□