Math 596 Homework 2

Cole Werry

February 25th 2024

EXERCISE 1.1)
Two bodies with masses equal to m_1 and m_2. The distance between their centers of mass is r. The law of universal gravitation states that the two bodies attract each other with an attraction force equal to:

$$F_g = G\frac{m_1 m_2}{r^2}$$

Find the dimension of the universal gravitational constant G.

Proof. We have force on the left (F_g), which has the dimensions mass times acceleration. Since F_g has dimensions $[M][L][T]^{-2}$, and $\frac{m_1 m_2}{r^2}$ has dimensions $\frac{[M]^2}{[L]^2}$

$$[G] \times \left(\frac{[M]^2}{[L]^2}\right) = [M][L][T]^{-2}$$

Solving for G:

$$[G] = \frac{[M][L][T]^{-2}}{\frac{[M]^2}{[L]^2}}$$

$$[G] = \frac{[M][L][T]^{-2} \times [L]^2}{[M]^2}$$

$$[G] = [L]^3[M]^{-1}[T]^{-2}$$

$\square$

EXERCISE 1.4) Dimensional analysis with experimental data for the free-fall of a body.
a) Make a dimensional analysis for falling distance h as a functions of mass m, gravitational acceleration g, and time t.

The relationship between the distance h as it relates to $m, g,$ *and* t is given by the following formula:

$$h = \alpha m^a g^b t^c$$

taking the dimensional analysis of the formula gives us:

$$[h] = [\alpha][m]^a[g]^b[t]^c$$

Thus,

$$L = 1 \cdot M^a(LT^{-2})^bT^c$$
$$= M^aL^bT^{-2b+c}$$

Solving for the unknowns we find that $a = 0, b = 1,\ and\ c = 2$. Therefore our equation can be written in the following way:

$$h = \alpha gt^2$$

Taking the derivative with respect to time t:

$$v = \frac{dh}{dt} = 2\alpha gt$$

Then the change from potential energy to kinetic energy happens and we get:

$$mg\alpha gt^2 = \frac{1}{2}m(2\alpha gt)^2$$

Reducing this:

$$\alpha = 2\alpha^2$$

Therefore,

$$\alpha = \frac{1}{2},\ or\ \alpha = 0$$

However if $\alpha = 0$ then we have no motion. Thus $\alpha = \frac{1}{2}$ and our universal model equation is the following:

$$h = \frac{1}{2}gt^2$$

EXERCISES 1.6) 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

$$f = \alpha l^a\rho^bF^c$$

where α, a, b and c are constants. Use dimensional analysis to determine values of a, b and c. The constant α is still to be determined by another condition or experiment. It actually turns out to be 2.

If $\alpha = 2$ then

$$f = 2l^a \rho^b F^c$$

Using the dimension given to us:

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

Thus

$$f = (M)^{a+b+c} (LT^{-2})^c$$

Solving for our exponents (I got kind of lost here)

EXERCISE FROM BOOK: Example 1.5
The universal formula for the speed of a free falling body may be interpreted as the following:

$$v = \alpha m^a t^b g^c h^d$$

Taking the dimensional analysis of this:

$$[v] = [\alpha][m]^a [t]^b [g]^c [h]^d$$

Thus,

$$LT^{-1} = 1 \cdot M^a T^b (LT^{-2})^c L^d$$

Simplifying:

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

Solving for our exponents tells us that $a = 0, b - 2c = -1,\ and\ c + d = 1$. However we have too many variables two of which we can say have a direct relation, the distance h and the time t. Assume we left out the distance h and that $d = 0$ then $c = 1,\ and\ b = 1$. Therefore our equation can be written as:

$$v = \alpha g t$$

Now assume that we left out the time t and that $b = 0$ instead then we have $c = d = \frac{1}{2}$. Therefore,

$$v = \alpha \sqrt{gh}$$

The change from potential energy to kinetic energy can be seen as:

$$mgh = \tfrac{1}{2} m v^2$$

Substituting $v = \alpha \sqrt{gh}$

$$mgh = \tfrac{1}{2} \alpha^2 mgh$$

Simplifying

$$\alpha = \sqrt{2}$$

Thus

$$v = \sqrt{2gh}$$

Which is another formula for a free fall body.
EXERCISE 1.13) From Fig. 1.6 and Eq. (1.61), derive the period formula Eq. (1.60) of the harmonic oscillation using an assumed model.

From the book we can find the equation for our model which is:

$$\tau = \alpha m^a l^b g^c$$

Taking the dimensional analysis:

$$[\tau] = [\alpha][m]^a[l]^b[g]^c$$

Thus,

$$= M^a L^b (LT^{-2})^c = M^a L^{b+c} T^{-2c}$$

This gives us three linear equations we must solve $a = 0, b+c = 0,\ and\ -2c = 1$. These equations have the soltions $c = -\frac{1}{2}, b = \frac{1}{2}, a = 0$ Therefore,

$$\tau = \alpha\sqrt{\frac{l}{g}}$$

1 Python Programming and linear models

Please see bottom of document for all graphs I couldn't figure out how to format them correctly sorry.
EXERCISE 3.1) Use Python to define a data sequence t=seq(2015,2018, length=100),and then plot the following two functions on the same figure: y = sin(2π(t - 0.1)) and y = cos2(2πt).

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

t = np.linspace(2015, 2018, 100)

y1 = np.sin(2 * np.pi * (t - 0.1))
y2 = np.cos(2 * 2 * np.pi * t)

plt.plot(t, y1, label='y = sin(2  (t - 0.1))')
plt.plot(t, y2, label='y = cos2(2  t)')

plt.xlabel('t')
plt.ylabel('y')
plt.title('Plot of two functions')
plt.legend()

plt.grid(True)
plt.show()
```

EXERCISE 3.4) Use Python to solve the following linear equations:

$$
\begin{gathered}
-3x + 2y + z = 1 \\
-2x - y + z = 2 \\
2x + y - 4z = 0
\end{gathered}
$$

```
import numpy as np

A = np.array([[-3, 2, 1],
              [-2, -1, 1],
              [2, 1, -4]])

b = np.array([1, 2, 0])

solution = np.linalg.solve(A, b)

print("Solution:")
print("x =", solution[0])
print("y =", solution[1])
print("z =", solution[2])
```

Solution: x = -1.0, y = -0.6666666666666669, z = -0.6666666666666666

EXERCISE 3.5) (a) Use Python to arrange the monthly Cuyamaca Tmax sequence data from January 1961 to December 1990 as a matrix with each row as year and each column as month.

I had a difficult time writing a code to read the csv file so I kind of brute forced it by adding all of the data myself sorry in advance for the sloppy work I will do my best to clean it up for next time.

```
import pandas as pd

tmax_data = [
[53.8, 55.7, 53, 64.3, 64.2, 81.3, 87.6, 85, 76.3, 69.3, 54, 48.9],
    [48.2, 47, 46.7, 68.3, 63.6, 74.9, 83.8, 88.1, 82.5, 70.6, 64.3, 54.9
                                    ],
    [46.1, 61.8, 53.7, 55, 68.3, 70.6, 85.7, 83.3, 80.1, 70.1, 58.5, 54.4
                                    ],
    [48.7, 52.2, 49.7, 55.6, 62.3, 74.9, 86.8, 85.3, 80.4, 75.3, 51.9, 50
                                    .8],
    [50.7, 52.4, 49.9, 56.7, 66.6, 69.8, 84.4, 84.9, 73.3, 76.8, 59.1, 46
                                    .9],
    [46.9, 46.2, 58.7, 66.5, 72.5, 78.4, 84.7, 86.6, 80.5, 69.6, 58.9, 50
                                    .2],
    [51, 55, 54.2, 47.8, 65, 70.9, 86.6, 86.5, 74.8, 74.4, 61.4, 43.3],
    [47.6, 57, 55.7, 59.5, 67.6, 76, 83.8, 78.9, 78.9, 69.4, 58.6, 47],
    [50.8, 44, 52.6, 61.7, 69.5, 73.8, 83.4, 89.3, 81.4, 65.6, 57, 58.1],
    [53.1, 55.5, 55.1, 56.5, 69.4, 77.2, 86.6, 86.6, 78.6, 67.3, 58.5, 47
                                    .9],
    [51.8, 52.7, 57.5, 56.8, 60.6, 73.7, 85.1, 85.6, 79.5, 64.7, 55.9, 44
                                    .3],
```

```
    [51.6, 56.8, 66.8, 63.1, 71.2, 77.3, 86, 81.6, 77.7, 62.8, 54, 47.8],
    [45.8, 47.9, 44.3, 59, 70.3, 78.4, 84.1, 82.3, 76.7, 69.7, 55, 54.9],
    [47.6, 53.3, 55.3, 59.6, 67.3, 81.3, 81.7, 82.2, 81.9, 67, 58.3, 50.6],
    [53.2, 50.4, 49.7, 47.1, 65.8, 75.6, 83, 82.7, 79.7, 66.7, 59.5, 53.5],
    [54.7, 52.7, 52.8, 56.2, 69.1, 76.8, 83.1, 78.5, 72, 67.1, 60.5, 54.6],
    [48.2, 59.2, 50, 63, 57.3, 77.3, 84, 82.3, 76.3, 71.6, 63.7, 57.6],
    [50.5, 50.3, 54.6, 55, 66.3, 78.6, 84.7, 82.7, 75.2, 74.1, 54.1, 47.5],
    [43.2, 47.9, 51.2, 60.8, 66.1, 78.5, 82.7, 79.4, 83.9, 69.9, 55.6, 56.2],
    [49.8, 54.4, 50.2, 59.3, 57.6, 74.3, 85.1, 81.5, 80, 70.4, 62.9, 57.8],
    [54.6, 56.8, 52, 63.3, 67.4, 83.6, 85.3, 87.2, 80.5, 63.7, 62.7, 58.1],
    [47.2, 53.1, 51.3, 59.5, 66.5, 71.9, 81.7, 85.6, 76.1, 67.8, 54.2, 50.4],
    [52.4, 51.4, 51.2, 52.1, 67.1, 73.4, 82, 81.2, 80, 68.4, 56.5, 52.6],
    [53.8, 58.4, 60.5, 60, 76.2, 76.3, 82.3, 81.2, 79.3, 63.3, 55.2, 45.4],
    [47.7, 51.7, 53.5, 66.2, 68.2, 79.5, 84.7, 84.1, 70.7, 67.4, 55, 53.5],
    [58.7, 53.5, 65.0461376, 65.04553072, 65.04492384, 65.04431697, 65.04371009, 65.04310321, 65.04249634, 65.04188946, 65.04128258, 65.04067571],
    [65.04006883, 65.03946195, 65.03885507, 65.0382482, 65.03764132, 65.03703444, 65.03642757, 65.03582069, 65.03521381, 65.03460693, 65.03400006, 65.03339318],
    [65.0327863, 65.03217943, 65.03157255, 65.03096567, 65.0303588, 65.02975192, 65.02914504, 65.02853816, 65.02793129, 65.02732441, 65.02671753, 65.02611066],
    [65.02550378, 65.0248969, 65.02429003, 65.02368315, 65.02307627, 65.02246939, 65.02186252, 65.02125564, 65.02064876, 65.02004189, 65.01943501, 65.01882813],
    [51.3, 48.8, 57.5, 62, 63.2, 77.5, 83.5, 81.9, 79.6, 73.7, 58.9, 47.9]
]
years = range(1961, 1991)

months = [
    'January', 'February', 'March', 'April', 'May', 'June',
    'July', 'August', 'September', 'October', 'November', 'December'
]

df = pd.DataFrame(tmax_data, columns=months, index=years)
```

```
print(df)
```

(b) Do the same for Tmin.

```
import pandas as pd

tmin_data = [
[26.7, 28.8, 31.7, 35, 37.4, 52.4, 54.6, 53.2, 41.4, 36.7, 28.7, 27.3],
    [27.1, 30.4, 27.1, 39, 38.3, 44.3, 50.1, 53.5, 45.8, 35.6, 28.5, 25.5
                                         ],
    [24.9, 33.7, 28.4, 29.4, 39.8, 44.2, 53.2, 52.8, 46.7, 40.1, 32.5, 24
                                         .8],
    [25.8, 24.4, 25.3, 31, 37.3, 43.1, 52.4, 52.4, 43.1, 42.9, 27.8, 32],
    [30, 28.4, 28.6, 33.3, 37.8, 39.6, 50.8, 52.1, 40.4, 38.6, 33.3, 29],
    [25.4, 25.9, 31.6, 37, 39.9, 45.2, 52, 53.8, 45.9, 37.5, 32.6, 31.7],
    [30.2, 29.9, 32.9, 28.2, 39.4, 42.2, 54.3, 55.1, 48.7, 36.1, 32.6, 23
                                         .3],
    [27.1, 34.8, 31.3, 31.9, 38.6, 47.6, 54.2, 49.8, 45.7, 37.4, 31.4, 23
                                         .4],
    [31.6, 28.4, 29.2, 38, 37, 44.5, 52.7, 55, 49.5, 35.1, 36.2, 26.6],
    [28.6, 28.2, 30, 28.5, 39.6, 45.8, 55.5, 56.4, 45, 37.1, 30.9, 26.1],
    [25.4, 28.4, 31.3, 30, 35.8, 44.1, 54.3, 53.7, 45.5, 33.4, 28.7, 24.3
                                         ],
    [24.2, 30.1, 33.2, 34.6, 37.7, 45.8, 56.4, 51.9, 44.5, 38.5, 28.7, 26
                                         .6],
    [24.9, 28, 26.7, 31.6, 40.5, 49.2, 51.7, 51.4, 43.2, 34.6, 31.4, 30.7
                                         ],
    [27.2, 27.4, 31.4, 32.8, 38.9, 50.4, 55.5, 50.9, 49.6, 39, 32.4, 26],
    [27.9, 27.1, 28.1, 29, 38.8, 46.4, 54.1, 51.8, 48.7, 34.8, 30.8, 28],
    [27, 30.5, 28.7, 30, 41, 47.8, 52.2, 47.7, 47.8, 39.8, 34.8, 25.1],
    [29.6, 27.8, 26, 34, 35, 49.4, 53.1, 55, 47.9, 39.5, 33.5, 33.8],
    [31.6, 32.6, 35.7, 33.7, 40.9, 51.9, 54.7, 53, 45.8, 42, 30.3, 24.9],
    [26.6, 24.2, 31, 35.4, 39.5, 50.9, 53.1, 50.8, 50.2, 40, 31.5, 29.6],
    [34.6, 34.8, 31.5, 35.5, 36.8, 47.9, 56.6, 52, 47.6, 42.5, 34.6, 33.2
                                         ],
    [31.6, 30.7, 31.7, 36.2, 40.9, 52.3, 55.7, 55.9, 49.9, 36.2, 34.3, 31
                                         .7],
    [28.3, 31.2, 31.9, 34.9, 39.8, 43.9, 52.7, 55.2, 46, 35, 31.6, 28.4],
    [30.6, 31.8, 34.5, 32.5, 41, 45.2, 52.3, 54.4, 53.3, 43.1, 35.8, 33.9
                                         ],
    [32.9, 29, 34.8, 34.8, 45.1, 46.9, 56.6, 54.7, 50.4, 36.4, 30.7, 29],
    [29.6, 29, 29.3, 39.7, 40.2, 48.3, 55.7, 52.5, 42.7, 37.9, 31.5, 30.7
                                         ],
    [33.4, 31.2, 39.75560494, 39.76581613, 39.77602732, 39.78623851, 39.
                                         79644971, 39.8066609, 39.
                                         81687209, 39.82708328, 39.
                                         83729448, 39.84750567],
    [39.85771686, 39.86792805, 39.87813925, 39.88835044, 39.89856163, 39.
                                         90877282, 39.91898402, 39.
                                         92919521, 39.9394064, 39.
                                         94961759, 39.95982879, 39.
                                         97003998],
    [39.98025117, 39.99046236, 40.00067356, 40.01088475, 40.02109594, 40.
                                         03130714, 40.04151833, 40.
```

```
                                        05172952, 40.06194071, 40.
                                        07215191, 40.0823631, 40.
                                        09257429],
    [40.10278548, 40.11299668, 40.12320787, 40.13341906, 40.14363025, 40.
                                        15384145, 40.16405264, 40.
                                        17426383, 40.18447502, 40.
                                        19468622, 40.20489741, 40.
                                        2151086],
    [28, 25.7, 32.2, 38.2, 40.4, 48.6, 55.9, 52, 48.9, 38.7, 32.8, 24.7]
]
years = range(1961, 1991)

months = [
    'January', 'February', 'March', 'April', 'May', 'June',
    'July', 'August', 'September', 'October', 'November', 'December'
]

df = pd.DataFrame(tmin_data, columns=months, index=years)

print(df)
```

(c) Do the same for Tmean.

```
    import pandas as pd

tmean_data = [
[40.2, 42.2, 42.4, 49.7, 50.8, 66.8, 71.1, 69.1, 58.9, 53, 41.4, 38.1],
    [37.6, 38.7, 36.9, 53.7, 51, 59.6, 67, 70.8, 64.1, 53.1, 46.4, 40.2],
    [35.5, 47.8, 41, 42.2, 54.1, 57.4, 69.5, 68.1, 63.4, 55.1, 45.5, 39.6
                                        ],
    [37.2, 38.3, 37.5, 43.3, 49.8, 59, 69.6, 68.8, 61.8, 59.1, 39.8, 41.4
                                        ],
    [40.4, 40.4, 39.2, 45, 52.2, 54.7, 67.6, 68.5, 56.9, 57.7, 46.2, 38],
    [36.2, 36.1, 45.1, 51.8, 56.2, 61.8, 68.4, 70.2, 63.2, 53.5, 45.8, 40
                                        .9],
    [40.6, 42.4, 43.6, 38, 52.2, 56.6, 70.5, 70.8, 61.7, 55.3, 47, 33.3],
    [37.3, 45.9, 43.5, 45.7, 53.1, 61.8, 69, 64.3, 62.3, 53.4, 45, 35.2],
    [41.2, 36.2, 40.9, 49.8, 53.3, 59.2, 68.1, 72.2, 65.5, 50.4, 46.6, 42
                                        .3],
    [40.9, 41.9, 42.5, 42.5, 54.5, 61.5, 71.1, 71.5, 61.8, 52.2, 44.7, 37
                                        ],
    [38.6, 40.6, 44.4, 43.4, 48.2, 58.9, 69.7, 69.7, 62.5, 49, 42.4, 34.3
                                        ],
    [37.9, 43.4, 50, 48.8, 54.4, 61.6, 71.2, 66.7, 61.1, 50.6, 41.4, 37.2
                                        ],
    [35.4, 38, 35.5, 45.3, 55.4, 63.8, 67.9, 66.8, 59.9, 52.2, 43.2, 42.8
                                        ],
    [37.4, 40.3, 43.3, 46.2, 53.1, 65.8, 68.6, 66.6, 65.7, 53, 45.4, 38.3
                                        ],
    [40.5, 38.8, 38.9, 38, 52.3, 61, 68.6, 67.2, 64.2, 50.7, 45.2, 40.7],
    [40.8, 41.6, 40.7, 43.1, 55.1, 62.3, 67.6, 63.1, 59.9, 53.5, 47.6, 39
                                        .9],
    [38.9, 43.5, 38, 48.5, 46.1, 63.4, 68.6, 68.6, 62.1, 55.5, 48.7, 45.7
                                        ],
    [41, 41.5, 45.2, 44.4, 53.6, 65.3, 69.7, 67.9, 60.5, 58, 42.2, 36.2],
```

```
    [34.9, 36, 41.1, 48.1, 52.8, 64.7, 67.9, 65.1, 67, 54.9, 43.5, 42.9],
    [42.2, 44.6, 40.8, 47.4, 47.2, 61.1, 70.8, 66.8, 63.8, 56.5, 48.7, 45
                                        .5],
    [43.1, 43.7, 41.9, 49.8, 54.1, 67.9, 70.5, 71.5, 65.2, 49.9, 48.5, 44
                                        .9],
    [37.8, 42.2, 41.6, 47.2, 53.2, 57.9, 67.2, 70.4, 61.1, 51.4, 42.9, 39
                                        .4],
    [41.5, 41.6, 42.9, 42.3, 54, 59.3, 67.2, 67.8, 66.6, 55.8, 46.2, 43.2
                                        ],
    [43.3, 43.7, 47.7, 47.4, 60.7, 61.6, 69.4, 68, 64.9, 49.9, 42.9, 37.2
                                        ],
    [38.6, 40.4, 41.4, 53, 54.2, 64, 70.2, 68.3, 56.7, 52.6, 43.3, 42.1],
    [46.1, 42.3, 52.40203296, 52.40682639, 52.41161983, 52.41641326, 52.
                                        42120669, 52.42600013, 52.
                                        43079356, 52.43558699, 52.
                                        44038042, 52.44517386],
    [52.44996729, 52.45476072, 52.45955416, 52.46434759, 52.46914102, 52.
                                        47393446, 52.47872789, 52.
                                        48352132, 52.48831476, 52.
                                        49310819, 52.49790162, 52.
                                        50269506],
    [52.50748849, 52.51228192, 52.51707536, 52.52186879, 52.52666222, 52.
                                        53145566, 52.53624909, 52.
                                        54104252, 52.54583596, 52.
                                        55062939, 52.55542282, 52.
                                        56021626],
    [52.56500969, 52.56980312, 52.57459656, 52.57938999, 52.58418342, 52.
                                        58897686, 52.59377029, 52.
                                        59856372, 52.60335716, 52.
                                        60815059, 52.61294402, 52.
                                        61773746],
    [39.7, 37.3, 44.9, 50.1, 51.8, 63.1, 69.7, 67, 64.3, 56.2, 45.9, 36.3
                                        ]
]
years = range(1961, 1991)

months = [
    'January', 'February', 'March', 'April', 'May', 'June',
    'July', 'August', 'September', 'October', 'November', 'December'
]

df = pd.DataFrame(tmean_data, columns=months, index=years)

print(df)
```

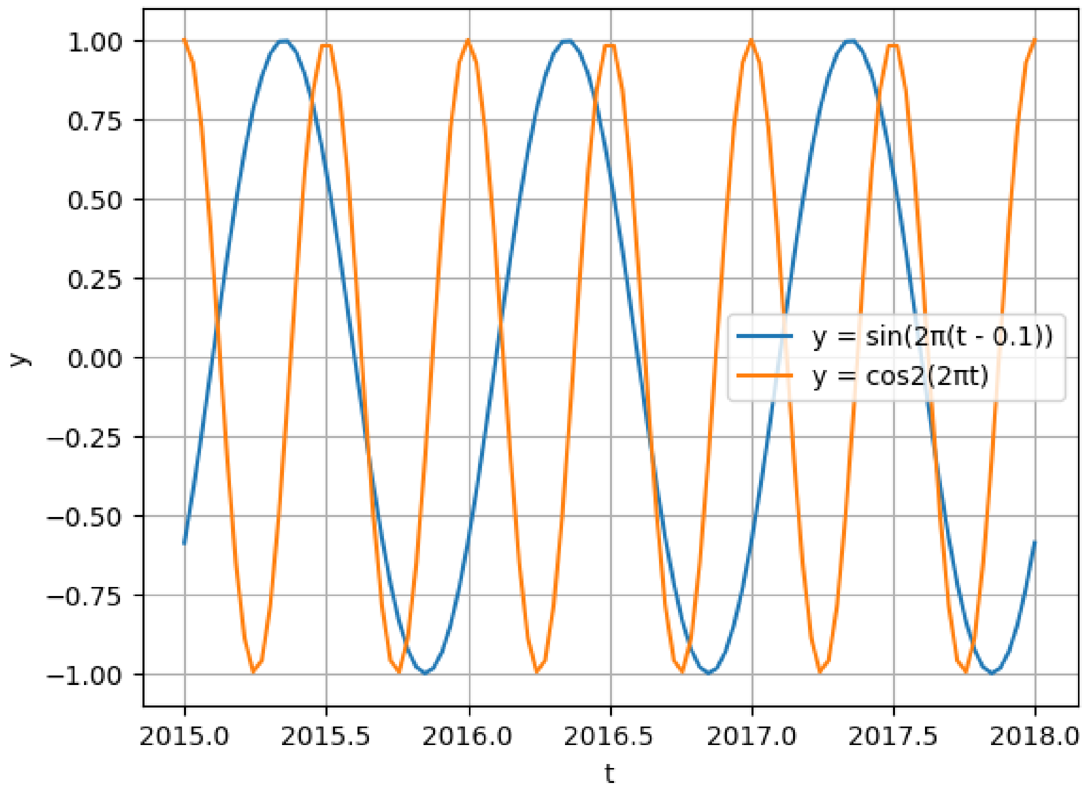
3.7) a) Use Python to plot the the Cuyamaca January Tmin time series from 1951 to 2010 with a continuous curve.

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("CA042239T.csv")
```

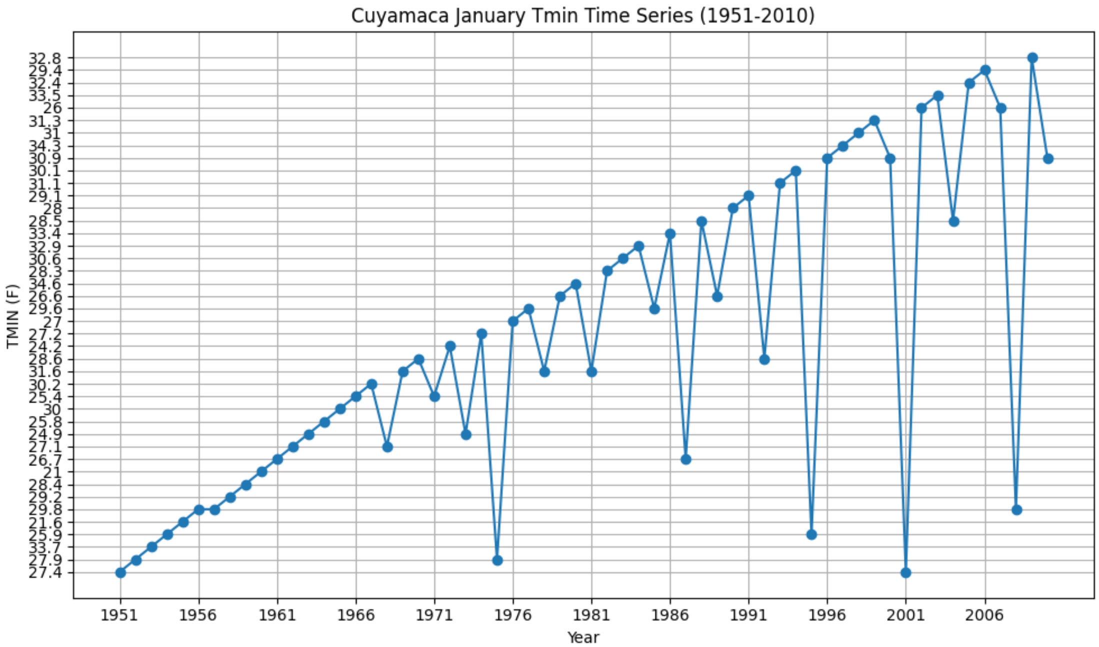
```
df.columns = df.columns.str.strip()

january_data = df[(df['Month'] == 1) & (df['YEAR'] >= 1951) & (df['YEAR']
                                          <= 2010)]

plt.figure(figsize=(10, 6))
plt.plot(january_data['YEAR'], january_data['TMIN (F)'], marker='o',
                                          linestyle='-')
plt.title('Cuyamaca January Tmin Time Series (1951-2010)')
plt.xlabel('Year')
plt.ylabel('TMIN (F)')
plt.grid(True)
plt.xticks(range(1951, 2011, 5))
plt.tight_layout()
plt.show()
```

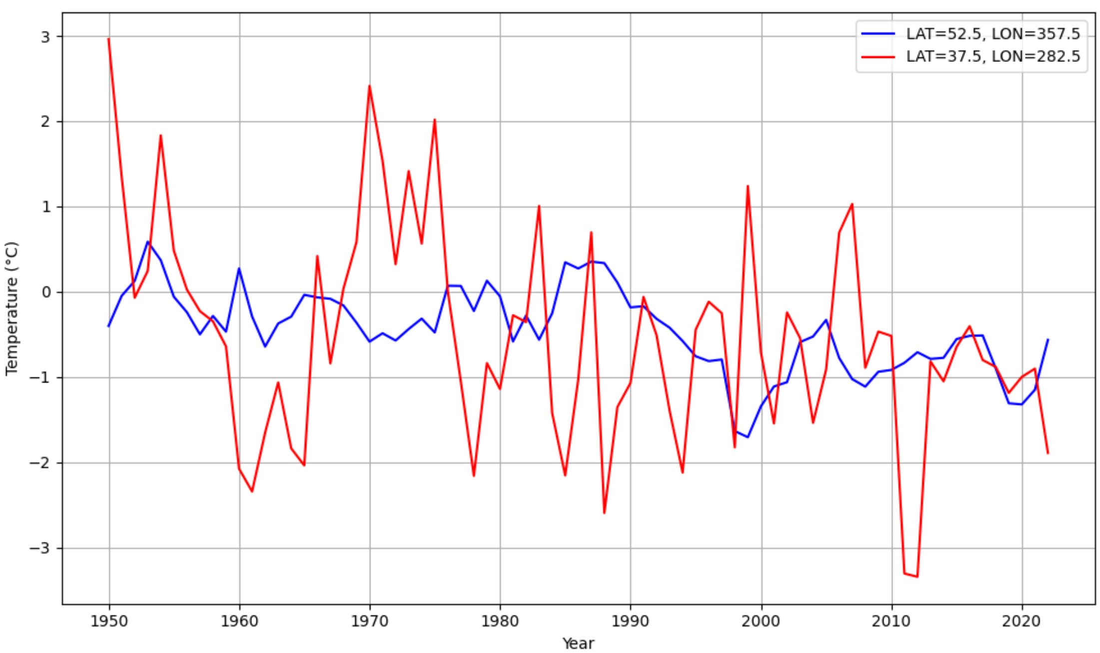
EXERCISE 3.9) Use the gridded NOAA global monthly temperature anomaly data NOAAGlobalTemp from the following website or another data source. Or use the NOAAGlobalT.csv data file from the book's data.zip file downloaded from the book website. Choose two 5-by-5 degrees lat-lon grid boxes of your interest. Plot the temperature anomaly time series of the two boxes on the same figure using two different colors.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


file = "NOAAGlobalT.csv"
noaa = pd.read_csv(file)


j_lon = np.where((noaa['LAT'] == -42.5) & (noaa['LON'] == 172.5))
j_was = np.where((noaa['LAT'] == 37.5) & (noaa['LON'] == 282.5))


lon_temp = noaa.iloc[j_lon[0], 3:].values
was_temp = noaa.iloc[j_was[0], 3:].values

years = np.arange(1950, 2023)

lon_temp = lon_temp.reshape(-1)[:len(years)]
was_temp = was_temp.reshape(-1)[:len(years)]

plt.figure(figsize=(10, 6))
plt.plot(years, lon_temp, linestyle='-', color='blue', label='LAT=52.5,
                                          LON=357.5')
plt.plot(years, was_temp, linestyle='-', color='red', label='LAT=37.5,
                                          LON=282.5')
plt.xlabel('Year')
plt.ylabel('Temperature ( C )')
plt.legend()
plt.grid(True)
```




```
plt.tight_layout()
plt.show()
```


Cuyamaca January Tmin Time Series (1951-2010)
TMIN (F)
32.8 29.4 32.4 33.5 26 31.3 31 34.3 30.9 30.1 31.1 29.1 28 28.5 33.4 32.9 30.6 28.3 34.6 26.6 29.6 27 27.2 24.2 28.6 31.6 30.2 25.4 30 25.8 24.9 27.1 26.7 21 28.4 29.2 29.8 21.6 25.9 33.7 27.9 27.4
1951 1956 1961 1966 1971 1976 1981 1986 1991 1996 2001 2006
Year


LAT=52.5, LON=357.5
LAT=37.5, LON=282.5
Temperature (°C)
3 2 1 0 −1 −2 −3
1950 1960 1970 1980 1990 2000 2010 2020
Year