Day 2:

Name: Daniel Delgado Acosta

Code:

You run an online clothing store called Panda’s Wardrobe. You need a DataFrame containing information about your products.

Create a DataFrame with the following data that your inventory manager sent you:

In:

```
import pandas as pd

df1 = pd.DataFrame({
  'Product ID': [1, 2, 3, 4],
  'Product Name': ['t-shirt', 't-shirt', 'skirt', 'skirt'],
  'Color': ['blue', 'green', 'red', 'black']
})

print(df1
```

Out:

```
Product ID Product Name  Color
0           1      t-shirt   blue
1           2      t-shirt  green
2           3        skirt    red
3           4        skirt  black
```

You're running a chain of pita shops called Pita Power. You want to create a DataFrame with information on your different store locations.

Use a list of lists to create a DataFrame with the following data:

In:

```
import pandas as pd

df2 = pd.DataFrame([
  [1, 'San Diego', 100],
  [2, 'Los Angeles', 120],
        [1, 'San Francisco', 90],
  [2, 'Los Angeles', 115]
],
  columns = ['Store ID', 'Location', 'Number of Employees'])

print(df2)
```

Out:

```
Store ID       Location  Number of Employees
0         1      San Diego                  100
1         2    Los Angeles                  120
2         1  San Francisco                   90
3         2    Los Angeles                  115
```

1. You're working for the County of Whoville and you just received a CSV of data about the different cities in your county. Read the CSV 'sample.csv' into a variable called df, so that you can learn more about the cities.
2. Let's inspect the CSV. Type print(df) on the next line and then run your code. What sort of data were you sent?

In:

```
import pandas as pd

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


print(df)
```

Out:

```
City  Population  Median Age
0      Maplewood      100000          40
1          Wayne      350000          33
2  Forrest Hills      300000          35
3        Paramus      400000          55
4     Hackensack      290000          39
```

1. You're working for a Hollywood studio, trying to use data to predict the next big hit. Load the CSV imdb.csv into a variable called df, so that you can learn about popular movies from the past 90 years.
2. Let's learn about these movies.

In:

```
import pandas as pd
#load the CSV below:
print(df.head())
print(df.info())
```

Out:

```
City  Population  Median Age
0      Maplewood      100000          40
1          Wayne      350000          33
2  Forrest Hills      300000          35
3        Paramus      400000          55
4     Hackensack      290000          39
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   City        5 non-null      object
 1   Population  5 non-null      int64
 2   Median Age  5 non-null      int64
dtypes: int64(2), object(1)
memory usage: 248.0+ bytes
None
```

You're going to staff the clinic for January of this year. You want to know how many visits took place in January of last year, to help you prepare.

1. Create variable january using a logical statement that selects the row of df where the 'month' column is 'January'.
2. Inspect january using print.

In:

```
import pandas as pd

df = pd.DataFrame([
  ['January', 100, 100, 23, 100],
  ['February', 51, 45, 145, 45],
  ['March', 81, 96, 65, 96],
  ['April', 80, 80, 54, 180],
  ['May', 51, 54, 54, 154],
  ['June', 112, 109, 79, 129]],
  columns=['month', 'clinic_east',
           'clinic_north', 'clinic_south',
           'clinic_west'])
Jan = df[df.month == 'January']
print(Jan)

print(Jan)
```

Out:

```
month  clinic_east  clinic_north  clinic_south  clinic_west
0  January          100           100            23          100
```

You want to see how the number of clinic visits changed between March and April.

1. Create the variable march_april, which contains the data from March and April. Do this using two logical statements combined using |, which means “or”.
2. Inspect march_april using print.

In:

```
import pandas as pd

df = pd.DataFrame([
  ['January', 100, 100, 23, 100],
  ['February', 51, 45, 145, 45],
  ['March', 81, 96, 65, 96],
  ['April', 80, 80, 54, 180],
  ['May', 51, 54, 54, 154],
  ['June', 112, 109, 79, 129]],
  columns=['month', 'clinic_east',
           'clinic_north', 'clinic_south',
           'clinic_west'])
march_april = df[(df.month == 'March') | (df.month == 'April')]
print(march_april)
```

Out:

```
month  clinic_east  clinic_north  clinic_south  clinic_west
2  March           81            96            65           96
3  April           80            80            54          180
```

Another doctor thinks that you have a lot of clinic visits in the late Winter.

1. Create the variable january_february_march, containing the data from January, February, and March. Do this using a single logical statement with the isin command.
2. Inspect january_february_march using print.

In:

```
import pandas as pd

df = pd.DataFrame([
  ['January', 100, 100, 23, 100],
  ['February', 51, 45, 145, 45],
  ['March', 81, 96, 65, 96],
  ['April', 80, 80, 54, 180],
  ['May', 51, 54, 54, 154],
  ['June', 112, 109, 79, 129]],
  columns=['month', 'clinic_east',
           'clinic_north', 'clinic_south',
           'clinic_west'])
january_february_march = df[df.month.isin(['January',
'February','March'])]
print(january_february_march)
```

Out:

```
month  clinic_east  clinic_north  clinic_south  clinic_west
0   January          100           100            23          100
1  February           51            45           145           45
2     March           81            96            65           96
```

1. Examine the code in the workspace. Note that df2 is a subset of rows from df. Type the following and press “Run”:
   print(df2)

Note that the indices on df2 are not consecutive.

2. Create a new DataFrame called df3 by resetting the indices on df2 (don’t use inplace or drop). Did df2 change after you ran this command?
3. Reset the indices of df2 by using the keyword inplace=True and drop=True. Did the indices of df2 change? How is df2 different from df3

In:

```
import pandas as pd

df = pd.DataFrame([
  ['January', 100, 100, 23, 100],
  ['February', 51, 45, 145, 45],
  ['March', 81, 96, 65, 96],
  ['April', 80, 80, 54, 180],
  ['May', 51, 54, 54, 154],
  ['June', 112, 109, 79, 129]],
  columns=['month', 'clinic_east',
           'clinic_north', 'clinic_south',
           'clinic_west']
)

df2 = df.loc[[1, 3, 5]]
df3 = df2.reset_index()
print(df3)
```

Out:

```
index     month  clinic_east  clinic_north  clinic_south  clinic_west
0      1  February           51            45           145           45
1      3     April           80            80            54          180
2      5      June          112           109            79          129
```

Complete each step below, running the cell between each step.

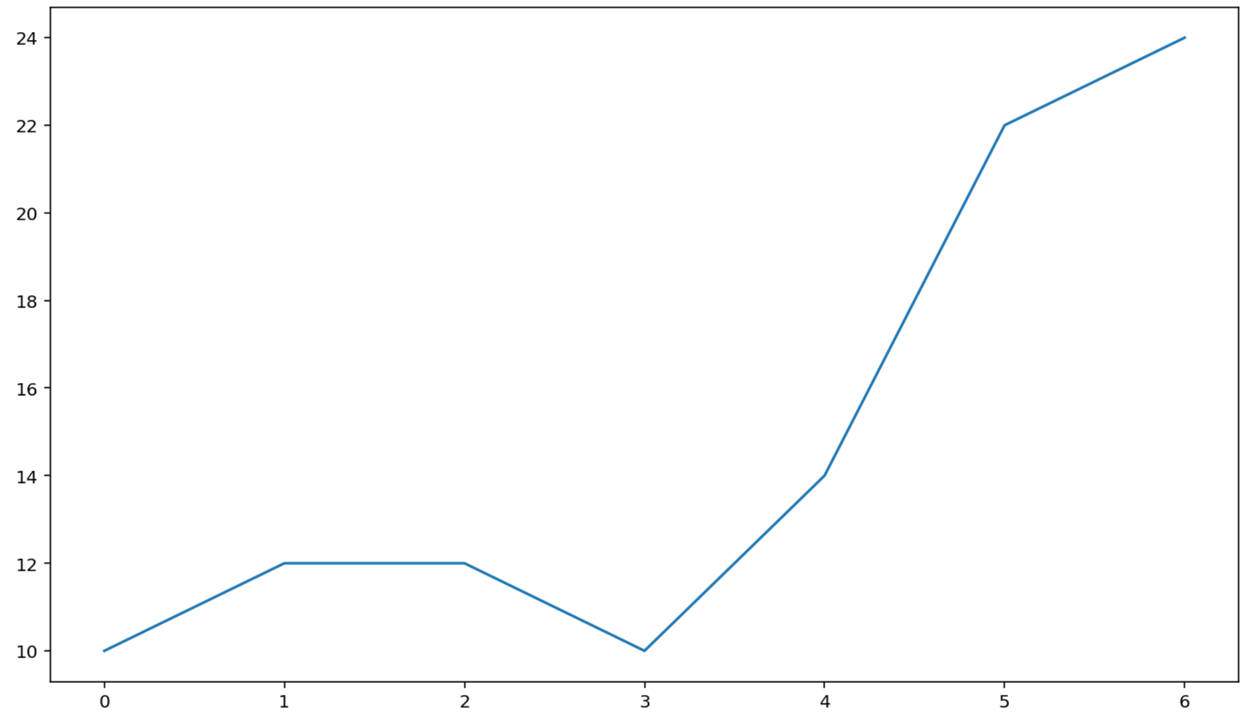
1. We are going to make a simple graph representing someone's spending on lunch over the past week. First, define two lists, days and money_spent, that contain the following integers:
2. Plot days on the x-axis and money_spent on the y-axis using plt.plot()
3. Show the plot using plt.show()

In:

```
from matplotlib import pyplot as plt

Days = [0, 1, 2, 3, 4, 5, 6]
Money_Spent = [10, 12, 12, 10, 14, 22, 24]
plt.plot(Days, Money_Spent)
plt.show()
```

Out:



1. We have defined lists called time, revenue, and costs. Plot revenue vs time.
2. Plot costs vs time on the same plot as the last line.
3. Show the plot using plt.show().

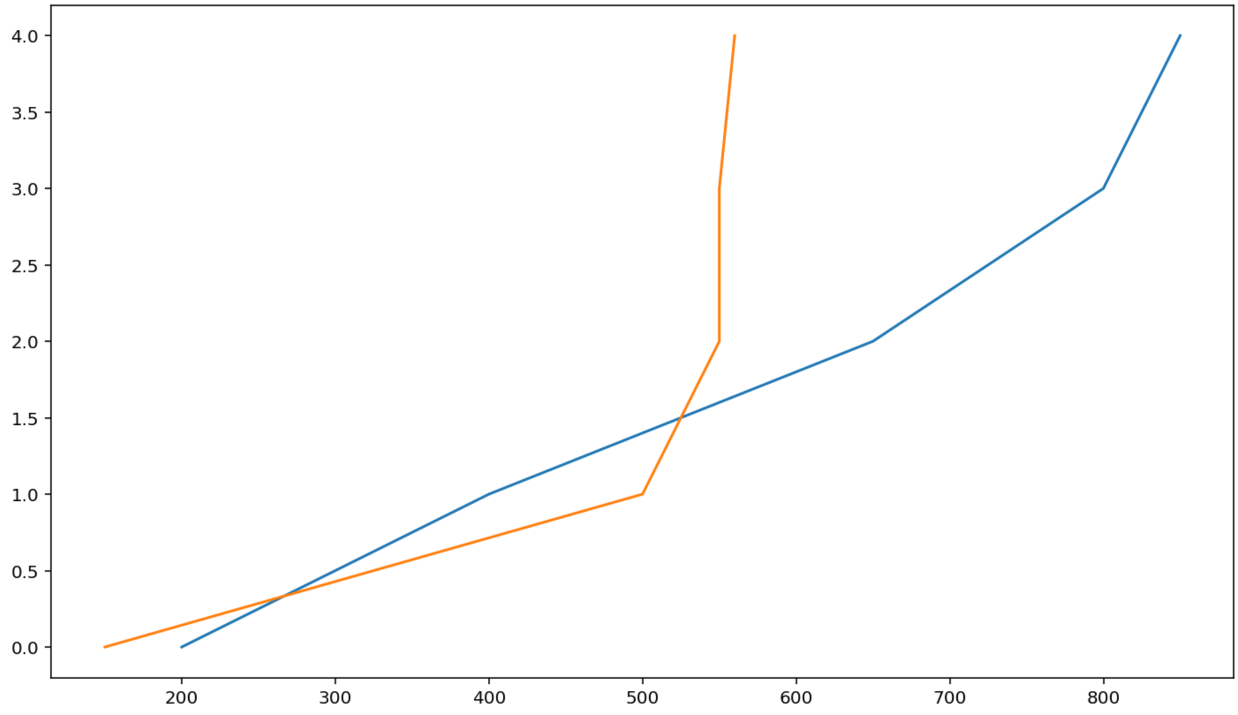
In:

```
from matplotlib import pyplot as plt

time = [0, 1, 2, 3, 4]
revenue = [200, 400, 650, 800, 850]
costs = [150, 500, 550, 550, 560]

# Your code goes here
plt.plot(revenue, time)
plt.plot(costs, time)
plt.show()
```

Out:



1. Plot revenue vs. time as a purple ('purple'), dashed ('--') line.
2. Plot costs vs. time as a line with the HEX color #82edc9 and square ('s') markers.

In:

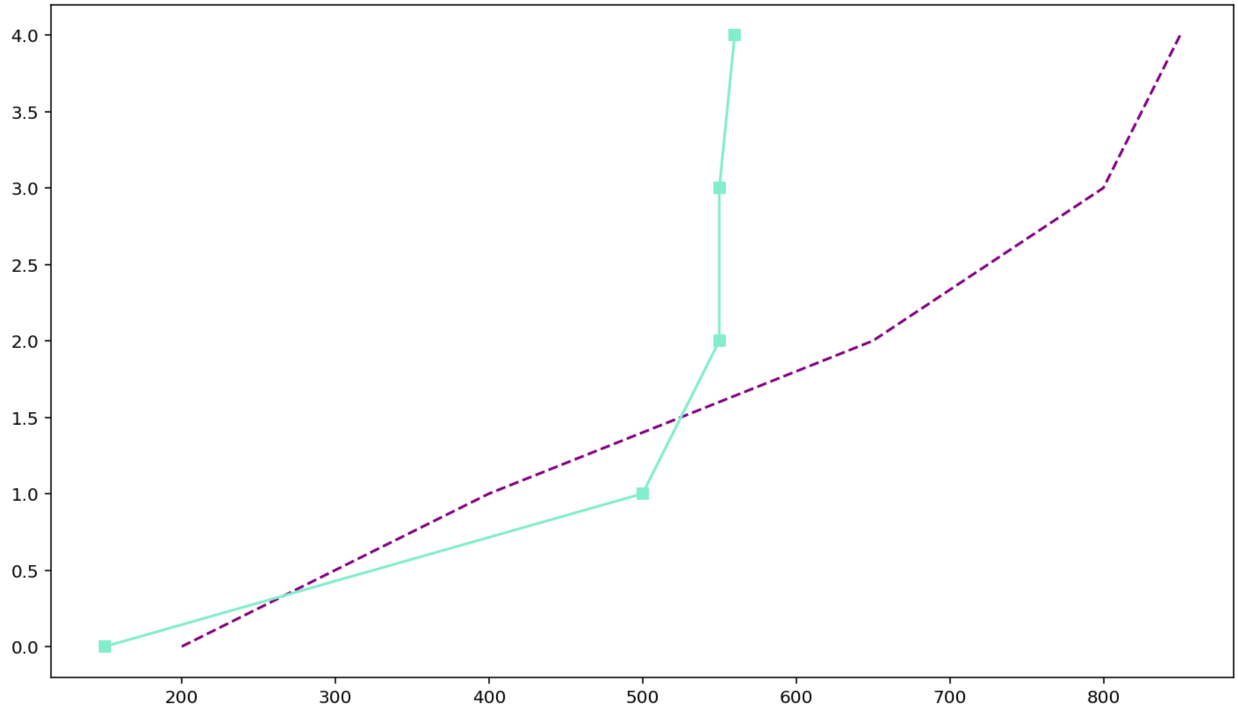
```
from matplotlib import pyplot as plt

time = [0, 1, 2, 3, 4]
revenue = [200, 400, 650, 800, 850]
costs = [150, 500, 550, 550, 560]

# Your code goes here
plt.plot(revenue, time, color='purple', linestyle='--')
plt.plot(costs, time, color='#82edc9', marker='s')

plt.show()
```

Out:



1. We have plotted a line representing someone’s spending on coffee over the past 12 years. Run the code to see the resulting plot.
2. Let’s modify the axes to zoom in a bit more on our line chart. Use plt.axis() to modify the axes so that the x-axis goes from 0 to 12, and the y-axis goes from 2900 to 3100.

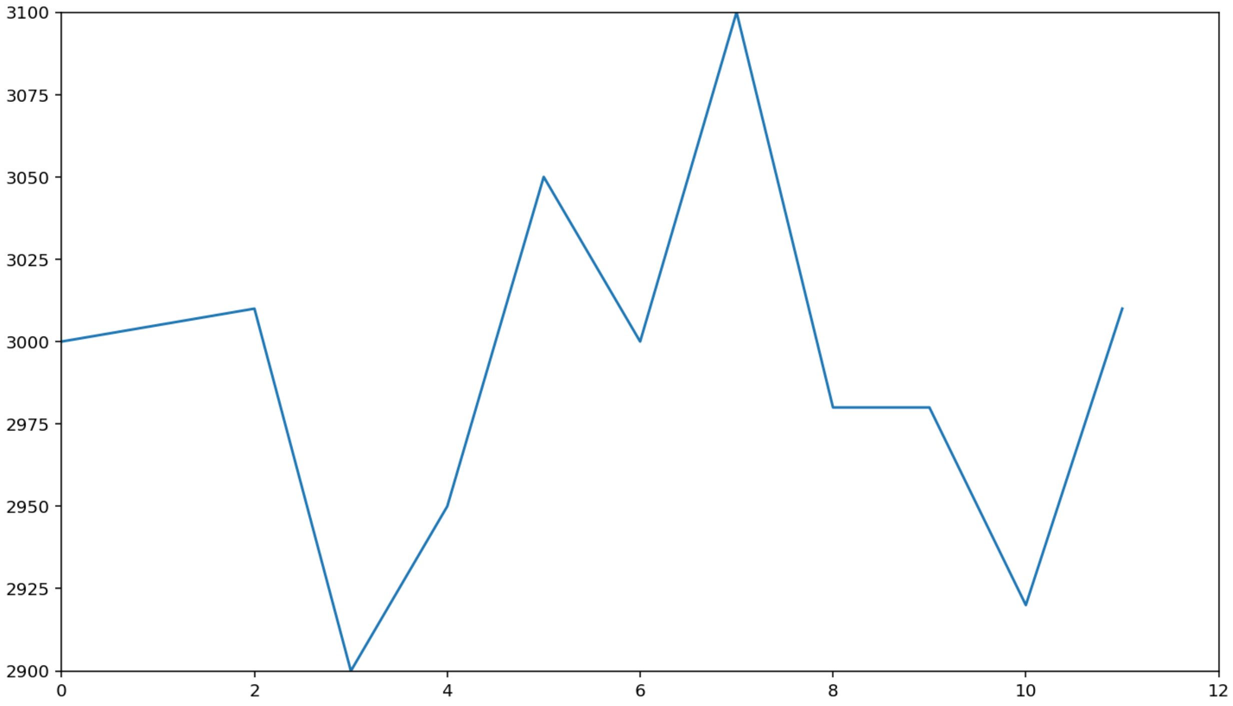
In:

```
from matplotlib import pyplot as plt

x = range(12)
y = [3000, 3005, 3010, 2900, 2950, 3050, 3000, 3100, 2980, 2980,
2920, 3010]
plt.plot(x, y)

# Your code goes here
plt.axis([0, 12, 2900, 3100])
plt.show()
```

Out:



1. Label the x-axis 'Time'.
2. Label the y-axis 'Dollars spent on coffee'.
3. Add the title 'My Last Twelve Years of Coffee Drinking'.

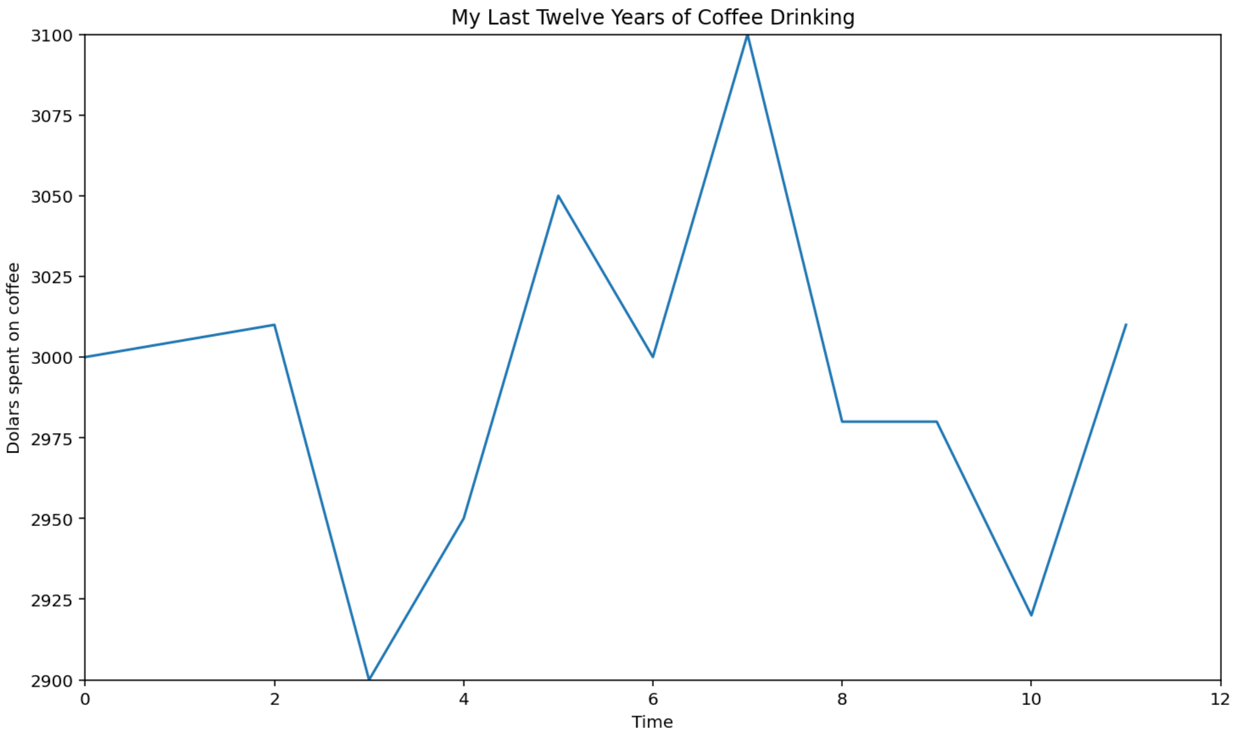
In:

```
from matplotlib import pyplot as plt

x = range(12)
y = [3000, 3005, 3010, 2900, 2950, 3050, 3000, 3100, 2980, 2980,
2920, 3010]
plt.plot(x, y)
plt.axis([0, 12, 2900, 3100])

# Insert your code here
plt.xlabel('Time')
plt.ylabel('Dolars spent on coffee')
plt.title('My Last Twelve Years of Coffee Drinking')
plt.show()
```

Out:


My Last Twelve Years of Coffee Drinking
Dolars spent on coffee
Time