Homework #3

CSCI 580, Fall 2023

Your Name Goes Here

Instructions

Answer the following questions based on the Life Expectancy (WHO) dataset available here using any of the Python libraries we have discussed in class:

Below is a code section that loads the dataset for you ...

```
import pandas as pd

df = pd.read_csv('https://www.ecst.csuchico.edu/~bjuliano/csci581/datasets/life_e
df
```

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	perce expend
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.2
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.5
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.2
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.1
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.0
...	...	...	...	...	...	...	...	
2933	Zimbabwe	2004	Developing	44.3	723.0	27	4.36	0.0
2934	Zimbabwe	2003	Developing	44.5	715.0	26	4.06	0.0
2935	Zimbabwe	2002	Developing	44.8	73.0	25	4.43	0.0
2936	Zimbabwe	2001	Developing	45.3	686.0	25	1.72	0.0
2937	Zimbabwe	2000	Developing	46.0	665.0	24	1.68	0.0

2938 rows × 22 columns

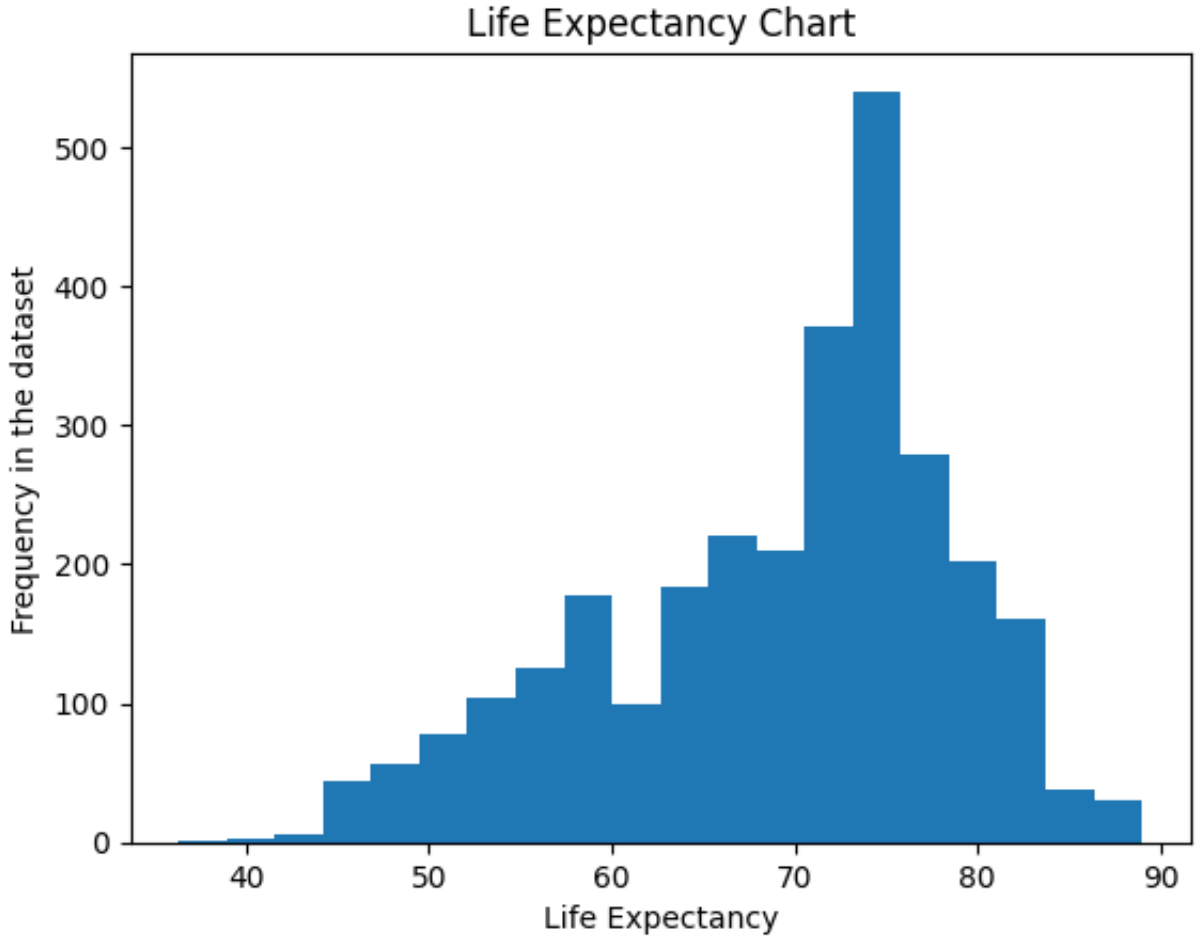
A. Which country has the *shortest* average life expectancy?

1. Use Python code to perform some exploratory data analysis (EDA) to determine your answer. For full credit, you must show all your work by using both text and code sections in this notebook. Use text sections to explain what you are doing to derive the required answer.
2. For full credit, also support your answer by providing an appropriate chart.

Solution(s)

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


df = pd.read_csv('https://www.ecst.csuchico.edu/~bjuliano/csci581/datasets/life_ex
df

min_expec = df['Life expectancy '].idxmin()
print(f'Index of shortest entry: {min_expec}')
country = df.loc[min_expec, 'Country']
# print(f'Country: {country}')
shortest_life = df.loc[min_expec, 'Life expectancy ']

print(f'The country with the shortest life expectancy is {country}, with a life ex

plt.hist(df['Life expectancy '], bins=20)
plt.xlabel('Life Expectancy')
plt.ylabel('Frequency in the dataset')
plt.title('Life Expectancy Chart')
plt.show()

# print(shortest_life)
```

```
Index of shortest entry: 1127
The couyntry with the shortest life expectancy is Haiti, with a life expectan
```



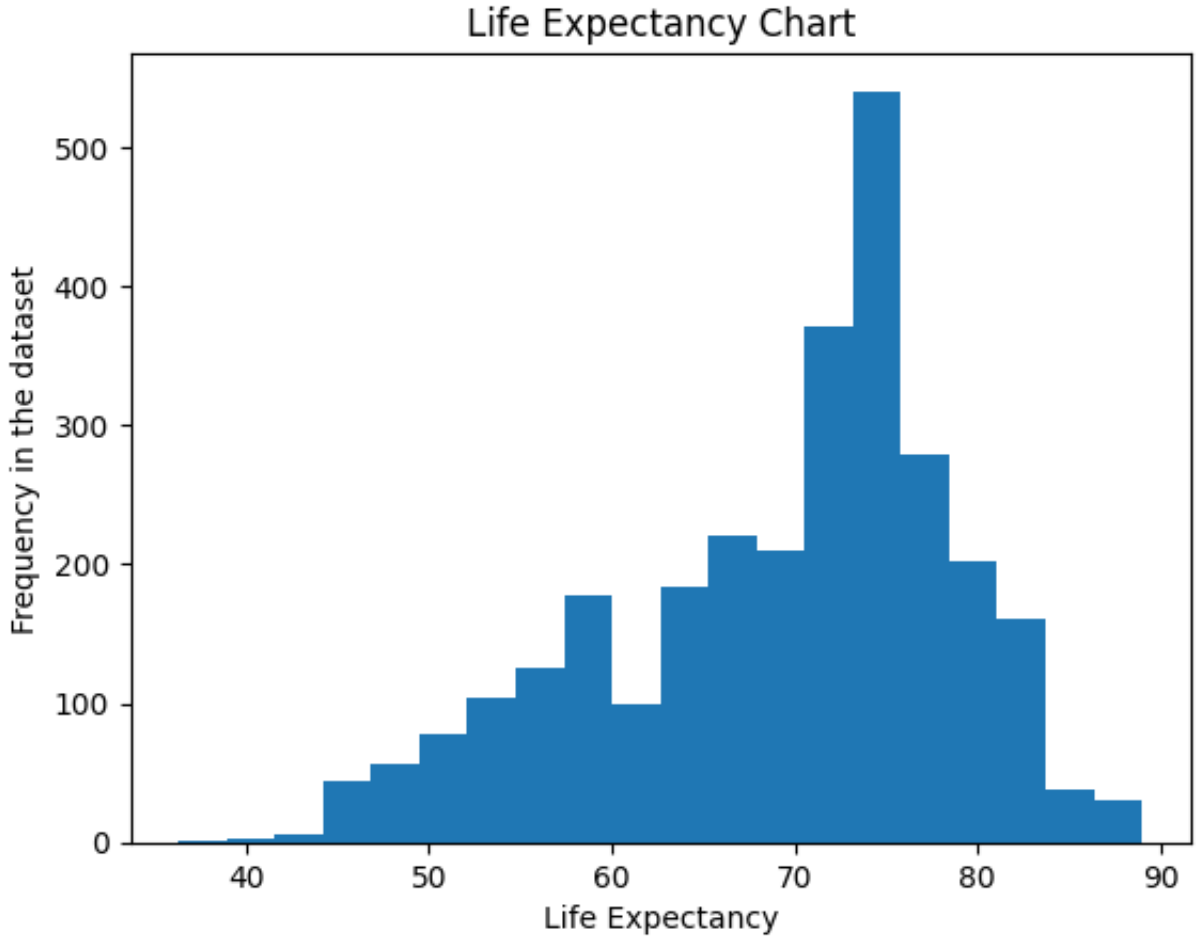
B. Which country has the *longest* average life expectancy?

1. Use Python code to perform some exploratory data analysis (EDA) to determine your answer. For full credit, you must show all your work by using both text and code sections in this notebook. Use text sections to explain what you are doing to derive the required answer.
2. For full credit, also support your answer by providing an appropriate chart.

Solution(s)

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


df = pd.read_csv('https://www.ecst.csuchico.edu/~bjuliano/csci581/datasets/life_e
df

max_expec = df['Life expectancy '].idxmax()
print(f'Index of longest entry: {max_expec}')
country = df.loc[max_expec, 'Country']
# print(f'Country: {country}')
longest_life = df.loc[max_expec, 'Life expectancy ']

print(f'The country with the longest life expectancy is {country}, with an averag

plt.hist(df['Life expectancy '], bins=20)
plt.xlabel('Life Expectancy')
plt.ylabel('Frequency in the dataset')
plt.title('Life Expectancy Chart')
plt.show()

# print(shortest_life)
```

```
Index of shortest entry: 241
The country with the longest life expectancy is Belgium, with an average life
```



C. There is a claim that based on the data from the USA, that there exists a *linear correlation* between a person's *body mass index* (BMI) and their *life expectancy*.

C.1: Use the whole USA dataset (only 16 data points) to generate a *linear regression* model to validate this claim. Call this model `model1`.

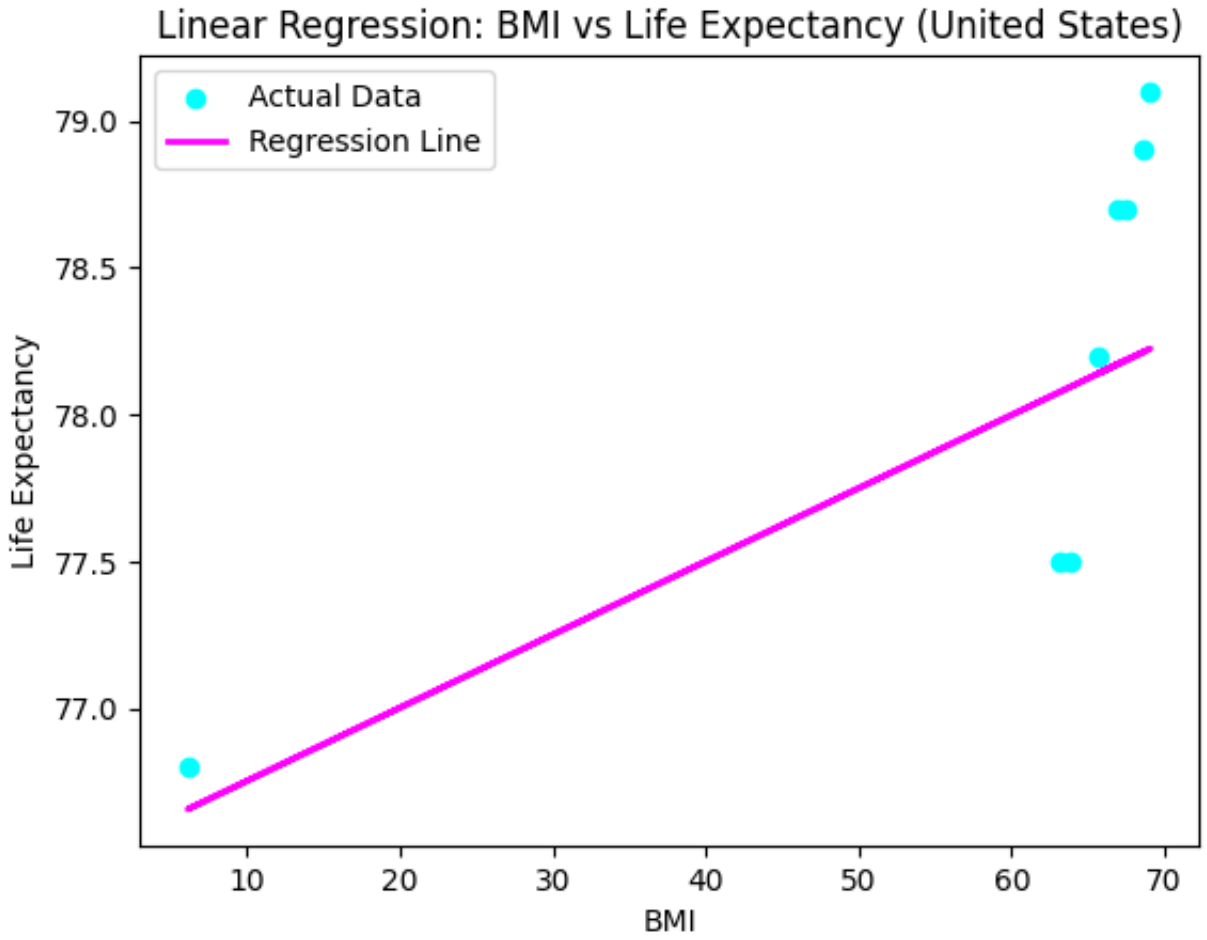
- Provide text and code sections that show your calculations and support your answer.
- Provide an appropriate chart as visualization to support your answer.
- Provide the model equation for your `model1`.

Solution(s)

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
df = pd.read_csv('https://www.ecst.csuchico.edu/~bjuliano/csci581/datasets/life_e
df = df[df['Country'] == 'United States of America']
df
# print(df.columns)
feature = df[' BMI ']
target = df['Life expectancy ']
# data prep
x_train, x_test, y_train, y_test = train_test_split(feature, target, test_size=0.
x_train = x_train.values.reshape(-1,1)
x_test = x_test.values.reshape(-1,1)
y_train = y_train.values.reshape(-1,1)
y_test = y_test.values.reshape(-1,1)


model1 = LinearRegression()

# training model
model1.fit(x_train, y_train)
prediction = model1.predict(x_test)
mse = mean_squared_error(y_test, prediction)
```

```
print(f'Mean Square Error: {mse}')

# display model
plt.scatter(x_test, y_test, color='cyan', label='Actual Data')
plt.plot(x_test, prediction, color='magenta', linewidth=2, label='Regression Line
plt.title('Linear Regression: BMI vs Life Expectancy (United States)')
plt.xlabel('BMI')
plt.ylabel('Life Expectancy')
plt.legend()
plt.show()

print("This model doesn't do a very good job, more datapoints (training data) wou
```

```
Mean Square Error: 0.31133687722066183
```




```
This model doesn't do a very good job, more datapoints (training data) would
```

C.2: Use the data from any ONE of the remaining 192 countries as a *test set* to validate the performance of `model1` that you generated in Part **C.1** above.

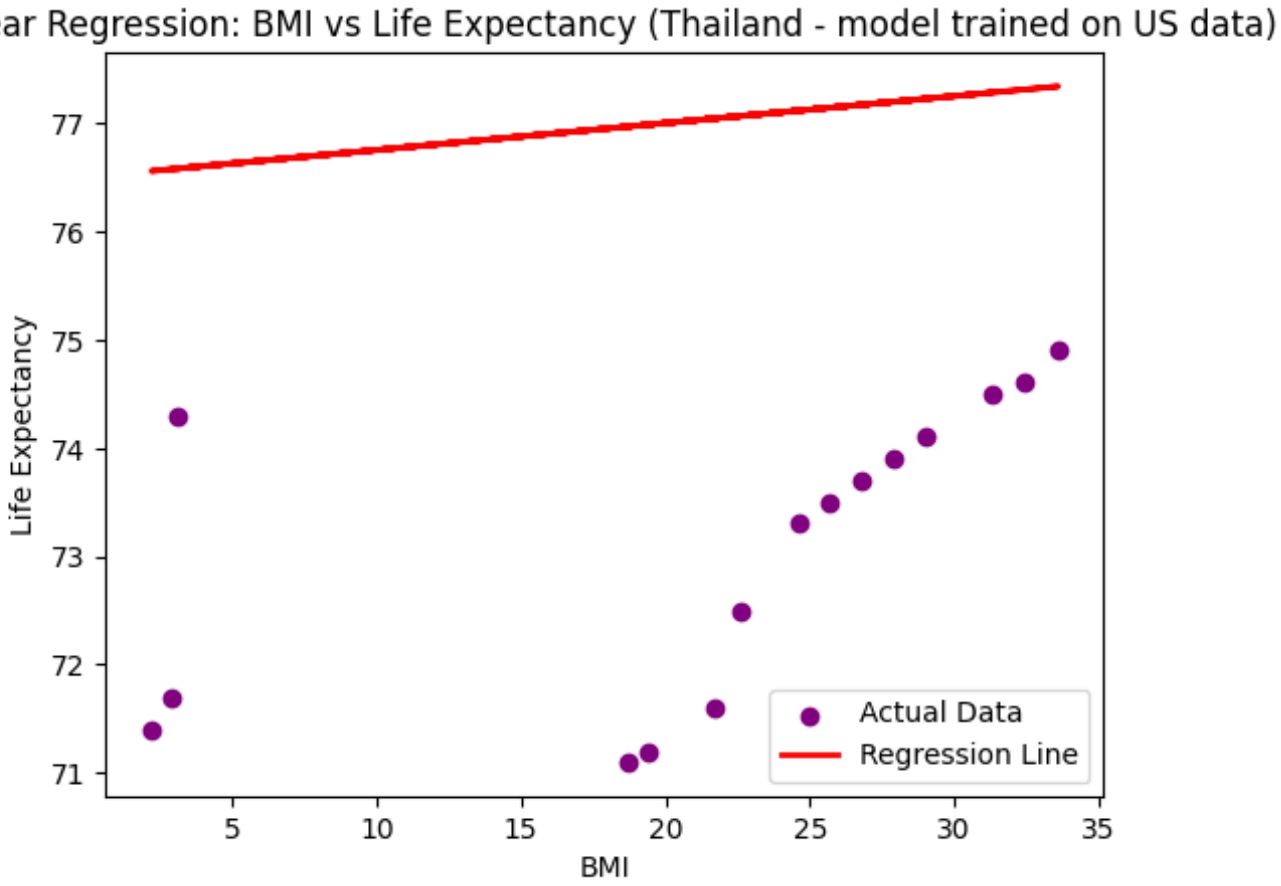
- *Explain* how well (or poorly) your USA data-trained `model1` is predicting the values for the country you selected.
- Provide text and code sections that show your calculations and support your answer.
- Provide an appropriate chart as visualization to support your answer.

The USA trained model `model1` already wasn't the best at predicting life expectancy based on BMI when using data from the united states. It only exacerbated it's subpar preformance when comparing it to a developing country like Thailand where the daily average caloric intake is likely much lower than the average intake for America

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
df = pd.read_csv('https://www.ecst.csuchico.edu/~bjuliano/csci581/datasets/life_e
df = df[df['Country'] == 'Thailand']

feature = df[' BMI ']
target = df['Life expectancy ']
# data prep
x_train, x_test, y_train, y_test = train_test_split(feature, target, test_size=0.
x_test = x_test.values.reshape(-1,1)
y_test = y_test.values.reshape(-1,1)


prediction = model1.predict(x_test)
mse = mean_squared_error(y_test, prediction)
print(f'Mean Square Error: {mse}')

# display model
```

```
plt.scatter(x_test, y_test, color='purple', label='Actual Data')
plt.plot(x_test, prediction, color='red', linewidth=2, label='Regression Line')
plt.title('Linear Regression: BMI vs Life Expectancy (Thailand - model trained on
plt.xlabel('BMI')
plt.ylabel('Life Expectancy')
plt.legend()
plt.show()
```

```
Mean Square Error: 17.046847041760515
```



C.3: Generate a second model using the whole USA dataset (only 16 data points) to generate a second *linear regression* model and this time do a 75%-25% *train-test split* where 75% of the dataset is used for *training* (12 data points) and the remaining 25% for *testing* (4 data points). Call this model `model2`.

1. Set `random_state=42` when running `train_test_split`.
2. Provide the model equation for your `model2`.
3. Explain how well (or poorly) `model2` performs on this smaller test set compared to `model1`.
4. Provide text and code sections that show your calculations and support your answer.
5. Provide an appropriate chart as visualization to support your answer.

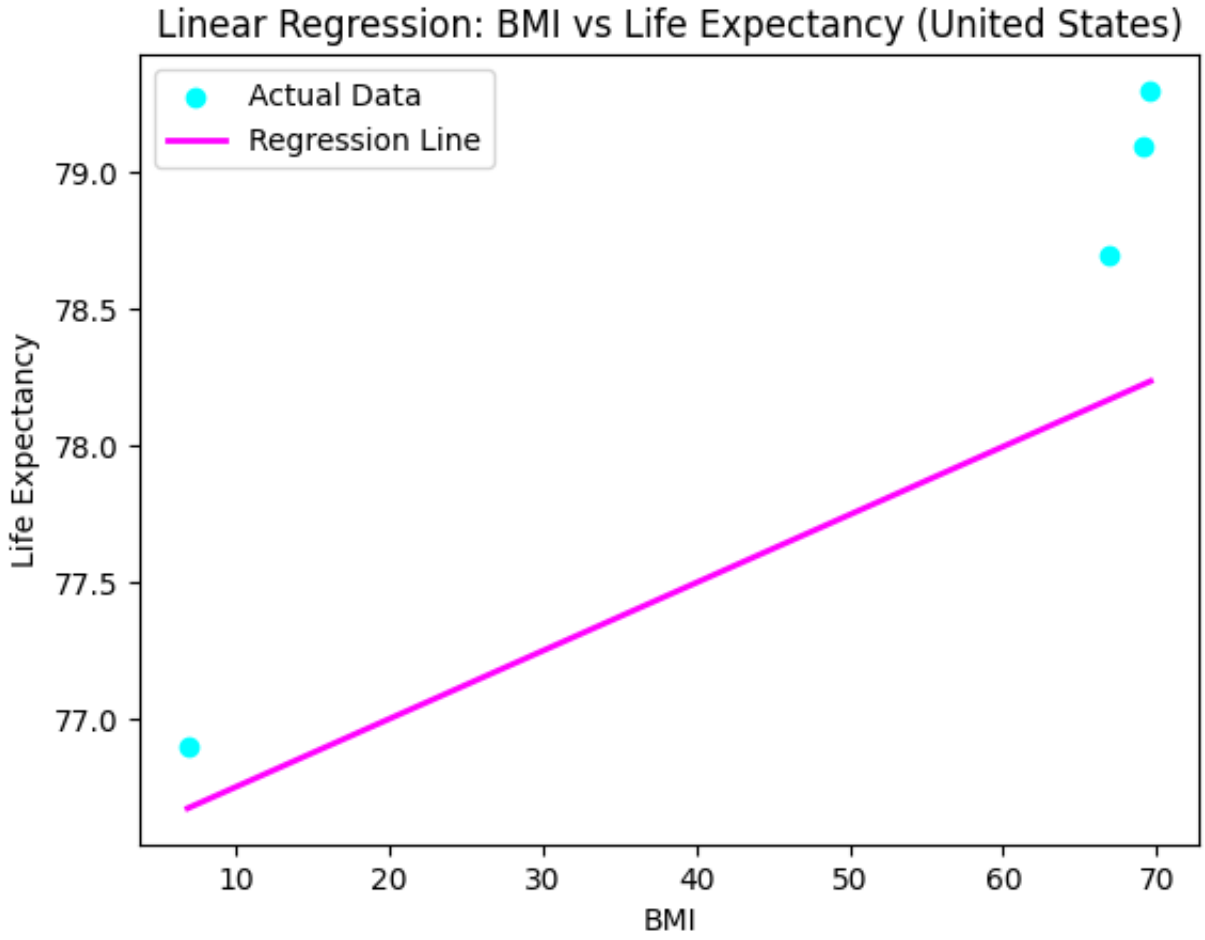
Solution(s)

Overall, there was a slight improvment in performance when compared to `model1`, but honestly, both arent great. This exercise was a great proof of the concept of 75-25 train/test split. At least with this one there aren't so many points that are so off from the line

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
df = pd.read_csv('https://www.ecst.csuchico.edu/~bjuliano/csci581/datasets/life_e
df = df[df['Country'] == 'United States of America']
df
# print(df.columns)
feature = df[' BMI ']
target = df['Life expectancy ']
```

```
# data prep
x_train, x_test, y_train, y_test = train_test_split(feature, target, test_size=0.
x_train = x_train.values.reshape(-1,1)
x_test = x_test.values.reshape(-1,1)
y_train = y_train.values.reshape(-1,1)
y_test = y_test.values.reshape(-1,1)


model2 = LinearRegression()

# training model
model2.fit(x_train, y_train)
prediction = model1.predict(x_test)
mse = mean_squared_error(y_test, prediction)
print(f'Mean Square Error: {mse}')

# display model
plt.scatter(x_test, y_test, color='cyan', label='Actual Data')
plt.plot(x_test, prediction, color='magenta', linewidth=2, label='Regression Line
plt.title('Linear Regression: BMI vs Life Expectancy (United States)')
plt.xlabel('BMI')
plt.ylabel('Life Expectancy')
plt.legend()
plt.show()
```

```
Mean Square Error: 0.5561568570564203
```



Notes

(REQUIRED) Include your overall final thoughts, comments, or observations regarding this exercise here.

Overall, I enjoyed this assignment. I've for the most part enjoyed the class as well. My only complaint about these last 3 programming assignments is that I wish we got more than just 3 hands-on assignments. I feel like our grades on exams would have gone up had we gotten more of these.