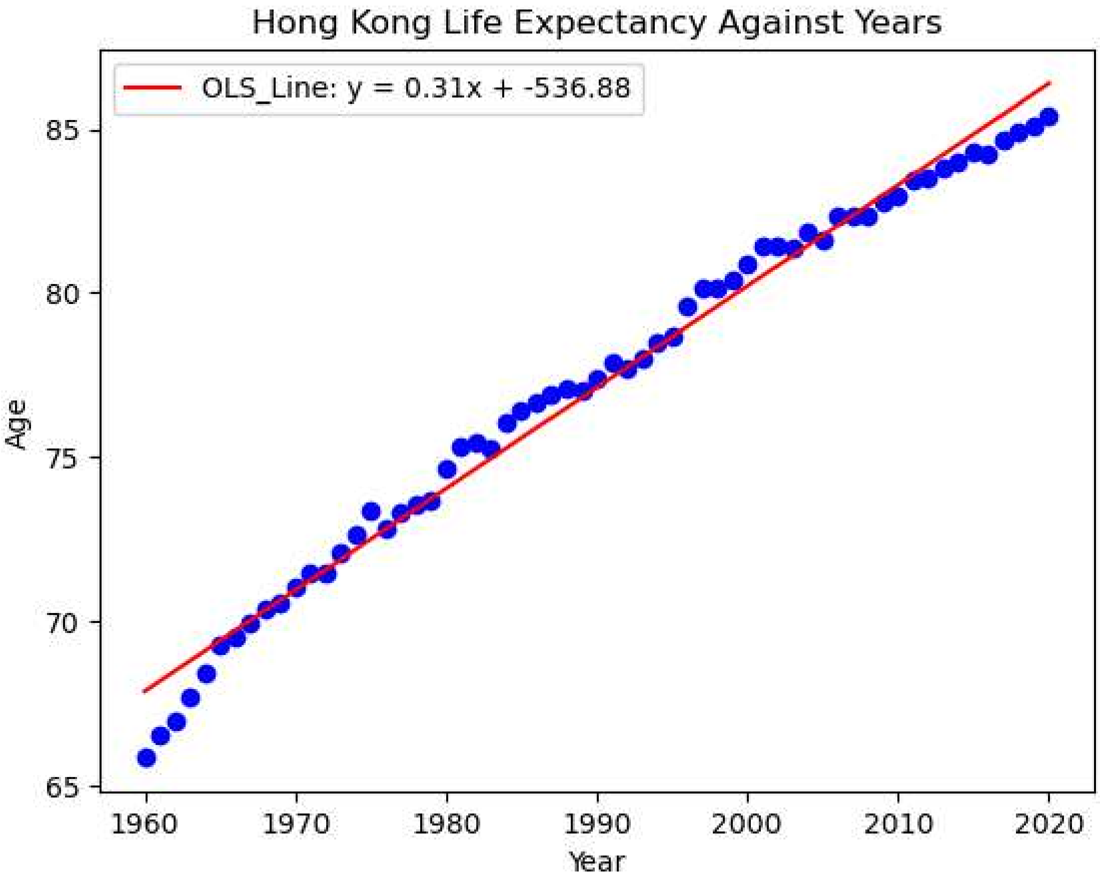
In [201…

```
#(a)

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

# Import the CSV file as a Pandas DataFrame
expectancy_df = pd.read_csv('expectancy.csv')

# Display the first few rows of the DataFrame
print(expectancy_df.head())
print(expectancy_df.tail())
```

```
   Year        Age
0  1960  65.866293
1  1961  66.558878
2  1962  66.977171
3  1963  67.685732
4  1964  68.446098
    Year        Age
56  2016  84.226829
57  2017  84.680488
58  2018  84.934146
59  2019  85.078049
60  2020  85.387805
```

In [202…

```
#(b)

# Fetch the data from the DataFrame expext_df
x = expectancy_df['Year']
y = expectancy_df['Age']

# Create a scatter plot
plt.scatter(x, y, color='blue')

# Calculate the OLS line (linear regression)
slope, intercept = np.polyfit(x, y, 1)
OLS_line = slope * x + intercept

# Plot the OLS line
plt.plot(x, OLS_line, color='red', label=f'OLS_Line: y = {slope:.2f}x + {intercept:

# Add labels and title
plt.xlabel('Year')
plt.ylabel('Age')
plt.title('Hong Kong Life Expectancy Against Years')
plt.legend()

# Show the plot
plt.show()
print (expectancy_df)
```



```
    Year        Age
0   1960  65.866293
1   1961  66.558878
2   1962  66.977171
3   1963  67.685732
4   1964  68.446098
..   ...        ...
56  2016  84.226829
57  2017  84.680488
58  2018  84.934146
59  2019  85.078049
60  2020  85.387805

[61 rows x 2 columns]
```

In [203…

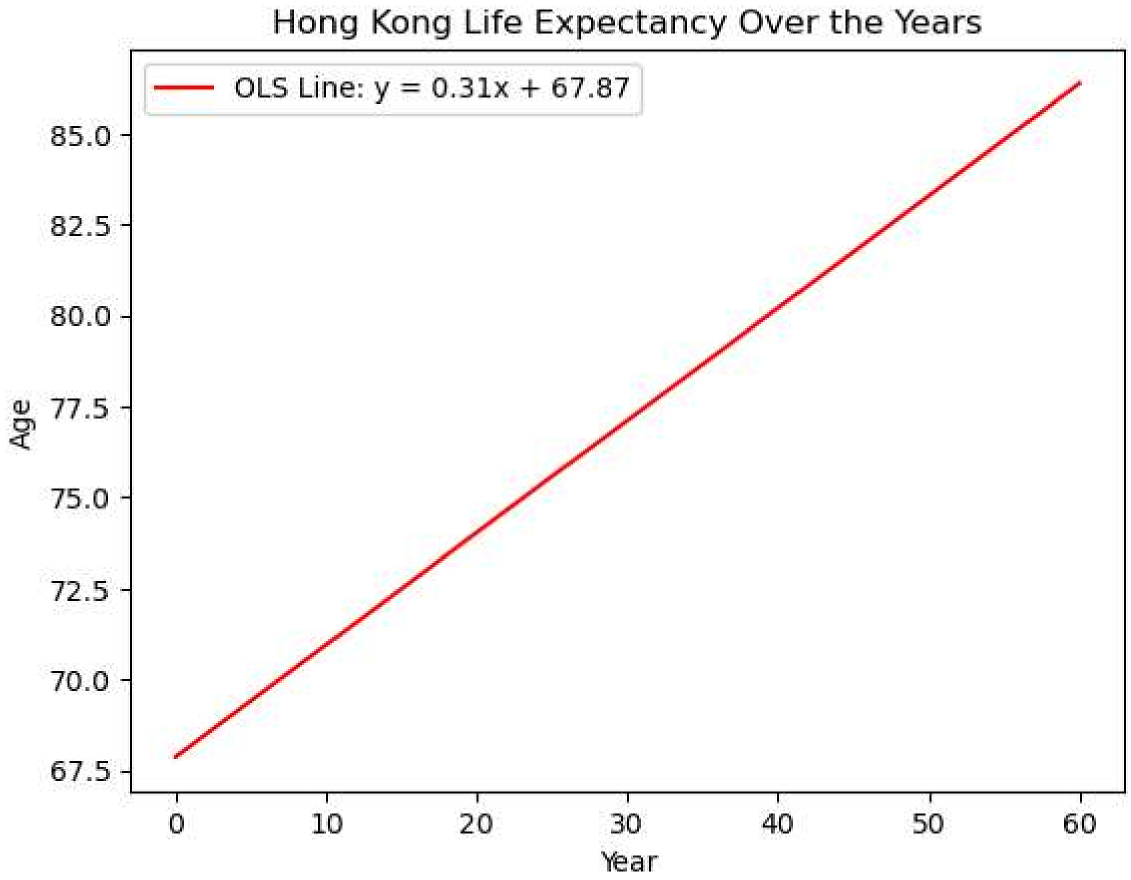
```
#(c)

# Life expectancy in Year 2025 = slope * x + intercept

# Pls see below
```

In [204…

```
print(expectancy_df)
expectancy_df.set_index('Year',inplace=True)
expectancy_df.reset_index(drop=True, inplace=True)
print(expectancy_df)
x = expectancy_df.index
y = expectancy_df['Age']

slope, intercept = np.polyfit(x, y, 1)
```

```
OLS_line = slope * x + intercept
print (f'slope = {slope:.2f}')
print (f'intercept = {intercept:.2f}')

# Plot the OLS line
plt.plot(x, OLS_line, color='red', label=f'OLS Line: y = {slope:.2f}x + {intercept:

# Add labels and title
plt.xlabel('Year')
plt.ylabel('Age')
plt.title('Hong Kong Life Expectancy Over the Years')
plt.legend()

# Show the plot
plt.show()
```

```
    Year        Age
0   1960  65.866293
1   1961  66.558878
2   1962  66.977171
3   1963  67.685732
4   1964  68.446098
..   ...        ...
56  2016  84.226829
57  2017  84.680488
58  2018  84.934146
59  2019  85.078049
60  2020  85.387805

[61 rows x 2 columns]
          Age
0   65.866293
1   66.558878
2   66.977171
3   67.685732
4   68.446098
..        ...
56  84.226829
57  84.680488
58  84.934146
59  85.078049
60  85.387805

[61 rows x 1 columns]
slope = 0.31
intercept = 67.87
```



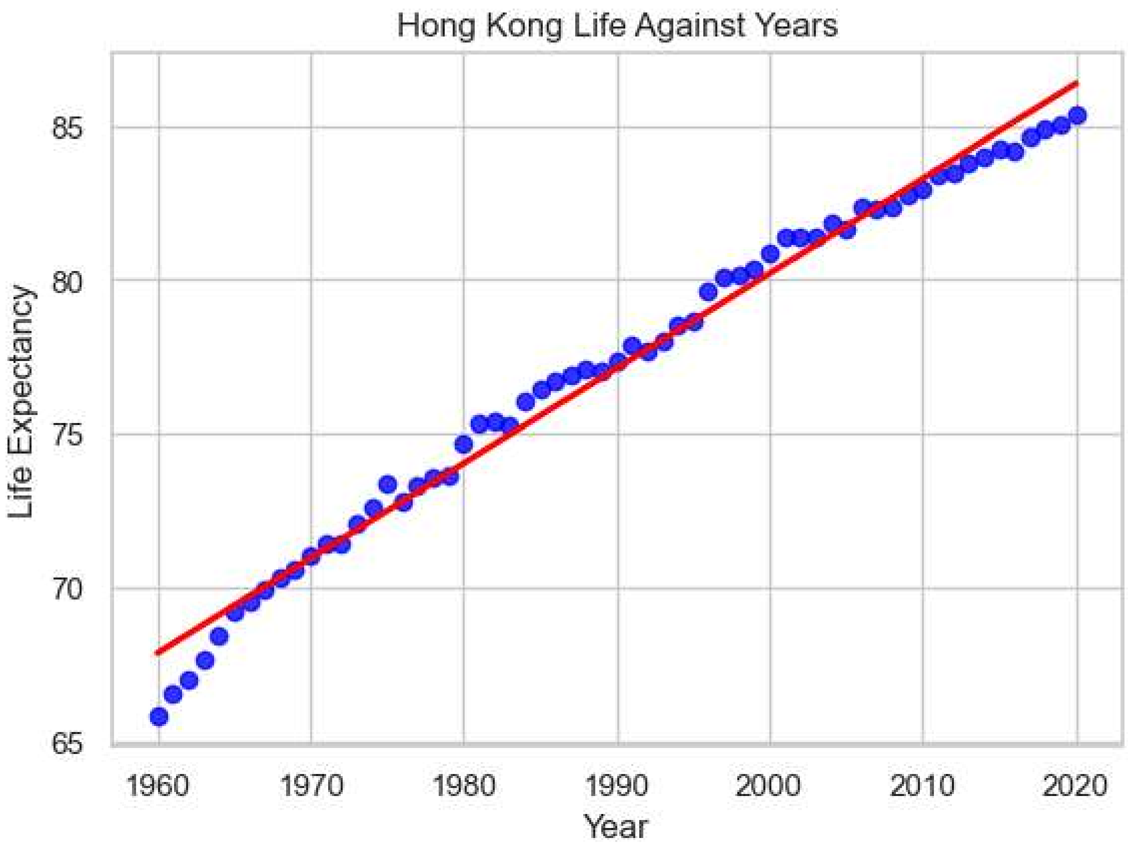
In [ ]:

```
# (c)

# intercept = 67.87 , x = 60 + 5 = 65 (2025 is equivalent to 5 year  by interpolati

# Therefore life expectancy = y = 0.31 x + intercept

#              = 0.31 * 65 + 67.87 = 88.02 (Answer)
```

In [223…

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

# Import the CSV file as a Pandas DataFrame
expectancy_df = pd.read_csv('expectancy.csv')

# Set the style of the plot
sns.set(style='whitegrid')

# Create a scatter plot with an OLS line
plt.figure(figsize=(6.5, 4.5))
sns.regplot(x='Year', y='Age', data=expectancy_df, ci=None, color='blue', line_kws=

# Add labels and title
plt.xlabel('Year')
plt.ylabel('Life Expectancy')
plt.title('Hong Kong Life Against Years')
```

```
# Show the plot
plt.show()
```



In [ ]:

```
# (d)
```

In [2]:

```
import pandas as pd
import plotly.express as px
import numpy as np

# Import the CSV file as a Pandas DataFrame
expectancy_df = pd.read_csv('expectancy.csv')


# Create a scatter plot
fig = px.scatter(expectancy_df, x='Year', y='Age', title='Hong Kong Life Expectancy

# Calculate the OLS line (linear regression)
slope, intercept = np.polyfit(expectancy_df['Year'], expectancy_df['Age'], 1)
OLS_line = slope * expectancy_df['Year'] + intercept

# Add the OLS line to the plot
fig.add_scatter(x=expectancy_df['Year'], y=OLS_line, mode='lines')
# Show the plot
fig.show()
```

In [ ]:
```
# (f)
# 1. Both Plotly and Seaborn give us the intercept  and slope  for the OLS regressi
# 2. Plotly: It offers interactive visualizations, allowing users to get more detai
#   his own format with ease.
# 3. Seaborn: it offer a more straightforward visualization ; it works seemlessly w
#    and can be complex when needed.
# 4. In terms of insight, both will provide the same statistical interpretation, bu
```

In [ ]:
```
# (g)
# R_square tells you how well the regression line fits the data. Higher R2 indicate
# however, if R2 value is too high, it may include some noise (data that does not r
```