Regression analysis of Gapminder data

In [1]:
```
# Get necessary Python packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
import statsmodels.api
```

In [2]:
```
file = "/content/drive/MyDrive/Colab Notebooks/CMSC320/Project3/08_gap-every-fi
data = pd.read_csv(file, sep='\t')
data
```

Out[2]:

	country	continent	year	lifeExp	pop	gdpPercap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
2	Afghanistan	Asia	1962	31.997	10267083	853.100710
3	Afghanistan	Asia	1967	34.020	11537966	836.197138
4	Afghanistan	Asia	1972	36.088	13079460	739.981106
...	...	...	...	...	...	...
1699	Zimbabwe	Africa	1987	62.351	9216418	706.157306
1700	Zimbabwe	Africa	1992	60.377	10704340	693.420786
1701	Zimbabwe	Africa	1997	46.809	11404948	792.449960
1702	Zimbabwe	Africa	2002	39.989	11926563	672.038623
1703	Zimbabwe	Africa	2007	43.487	12311143	469.709298

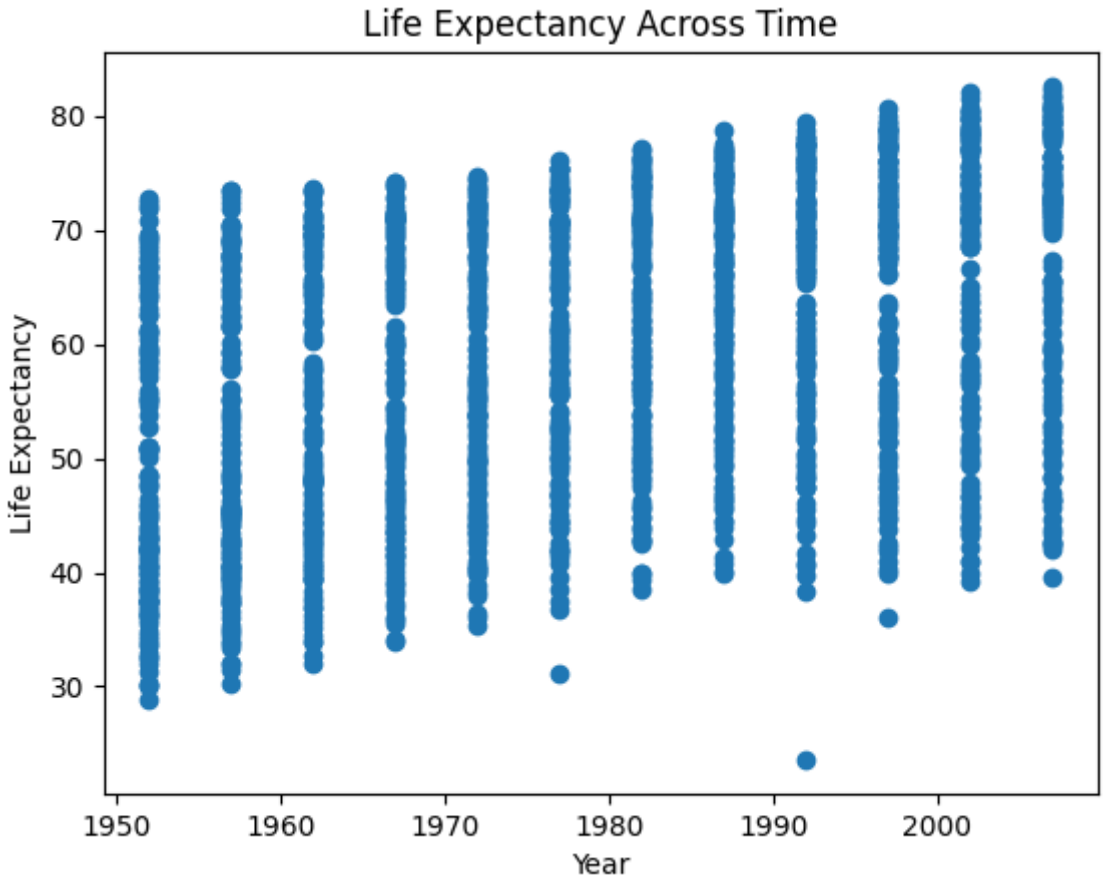
1704 rows × 6 columns

Exercise 1

In [3]:
```
x = data['year']
y = data['lifeExp']

plt.scatter(x, y)
plt.xlabel("Year")
plt.ylabel("Life Expectancy")
plt.title("Life Expectancy Across Time")
```

Out[3]:
```
Text(0.5, 1.0, 'Life Expectancy Across Time')
```



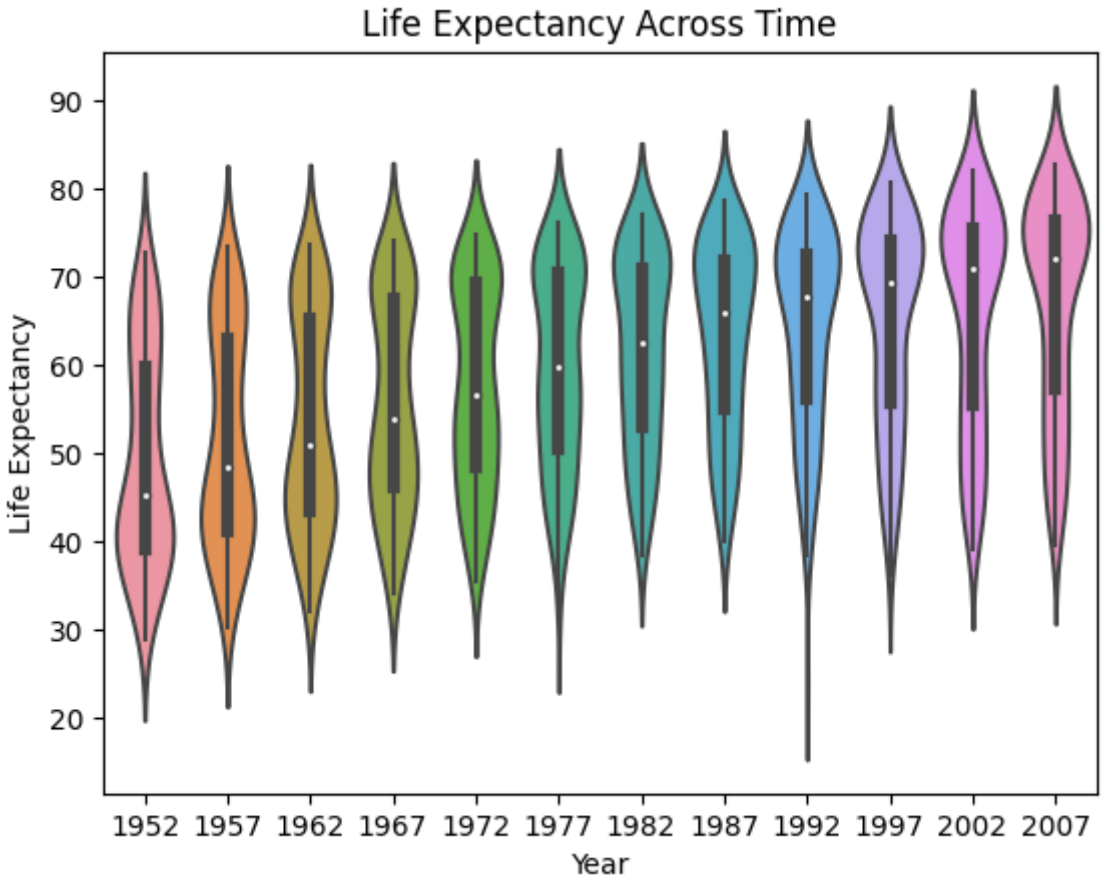
Question 1

We can see that the minimum and maximum of life expectancy is slightly increasing over time, even though there are some noise points, which means the mean data of life expectancy is increasing. So, we can conclude that there is a increasing general trend.

Question 2

In [4]:
```
sns.violinplot(x='year', y='lifeExp', data=data)
plt.xlabel("Year")
plt.ylabel("Life Expectancy")
plt.title("Life Expectancy Across Time")
```

Out[4]:
```
Text(0.5, 1.0, 'Life Expectancy Across Time')
```



From the violin plot, we can see that before year of 1972, the value is skewed to the bottom(less value). Besides, after year of 1972, the value is skewed to the top(more value). So, the distribution is not unimodal nor symmetric as the life expectancy has several peak in some years.

Question 3

Yes, I will reject the null hypothesis of no relationship. We can expect that life expectancy has generally increased over time based on the plot.

Question 4

I will assume it still have violin shape for each year as the residuals are calculated based on the data we have. Besides, the mean of residuals for each year will around zero.

Question 5

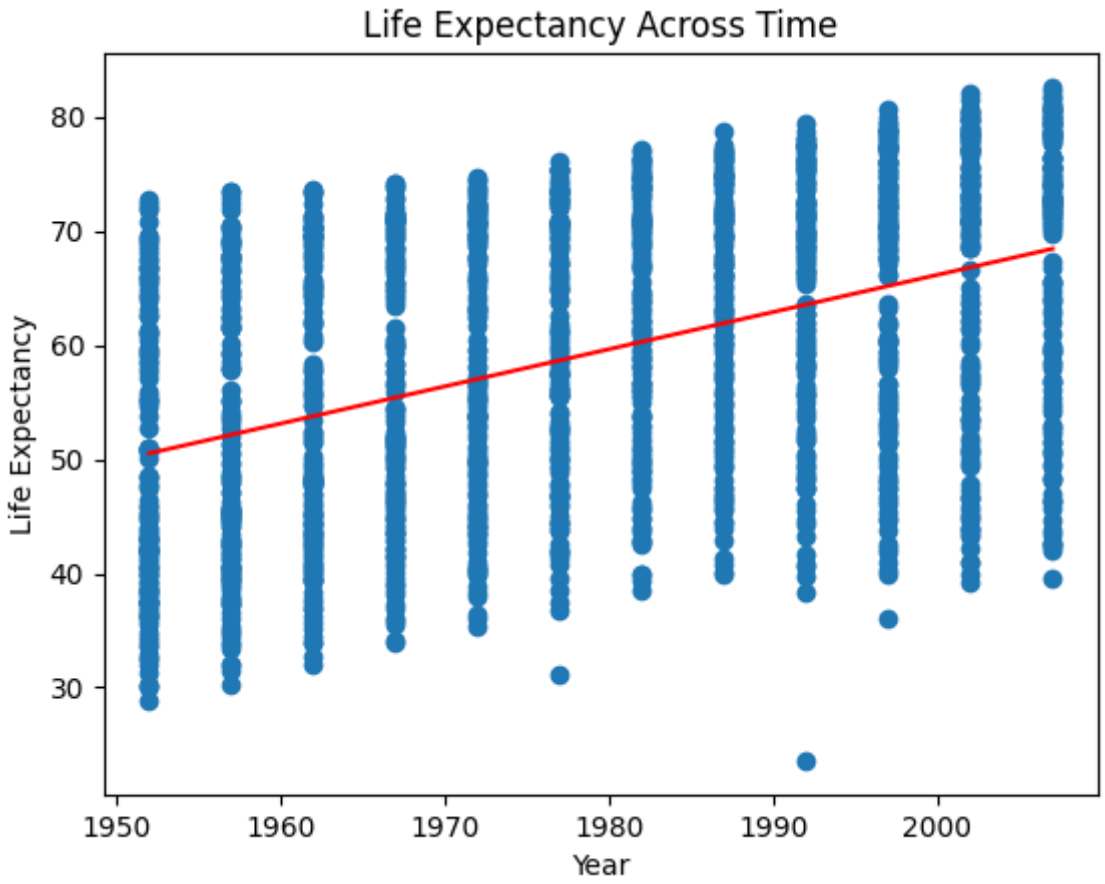
The plot should have a constant variance as the residuals are roughly the same; Also, the residuals should be symmetrically distributed around zero.

Exercise 2

In [5]:
```
# Calculate the model
x = data[['year']]
y = data['lifeExp']
model = LinearRegression().fit(x, y)

# Plot the model on scatter plot
plt.scatter(data['year'], data['lifeExp'])
plt.plot(x, model.predict(x), color='red')
plt.xlabel("Year")
plt.ylabel("Life Expectancy")
plt.title("Life Expectancy Across Time")
```

Out[5]:
```
Text(0.5, 1.0, 'Life Expectancy Across Time')
```



Question 6

In [6]:
```
res1 = statsmodels.formula.api.ols('lifeExp ~ year', data=data).fit()
 print(res1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                lifeExp   R-squared:                       0.190
Model:                            OLS   Adj. R-squared:                  0.189
Method:                 Least Squares   F-statistic:                     398.6
Date:                Sat, 29 Apr 2023   Prob (F-statistic):           7.55e-80
Time:                        03:24:59   Log-Likelihood:                -6597.9
No. Observations:                1704   AIC:                         1.320e+04
Df Residuals:                    1702   BIC:                         1.321e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept   -585.6522     32.314    -18.124      0.000    -649.031    -522.273
year           0.3259      0.016     19.965      0.000       0.294       0.358
==============================================================================
Omnibus:                      386.124   Durbin-Watson:                   0.197
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               90.750
Skew:                          -0.268   Prob(JB):                     1.97e-20
Kurtosis:                       2.004   Cond. No.                     2.27e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct
ly specified.
[2] The condition number is large, 2.27e+05. This might indicate that there ar
e
strong multicollinearity or other numerical problems.
```

The slope coefficient for year gives an estimate of the average increase in life expectancy per year, which the value is 0.3259.

Question 7

Yes, I would reject the null hypothesis of no relationship between year and life expectancy. As the p-value is 0.

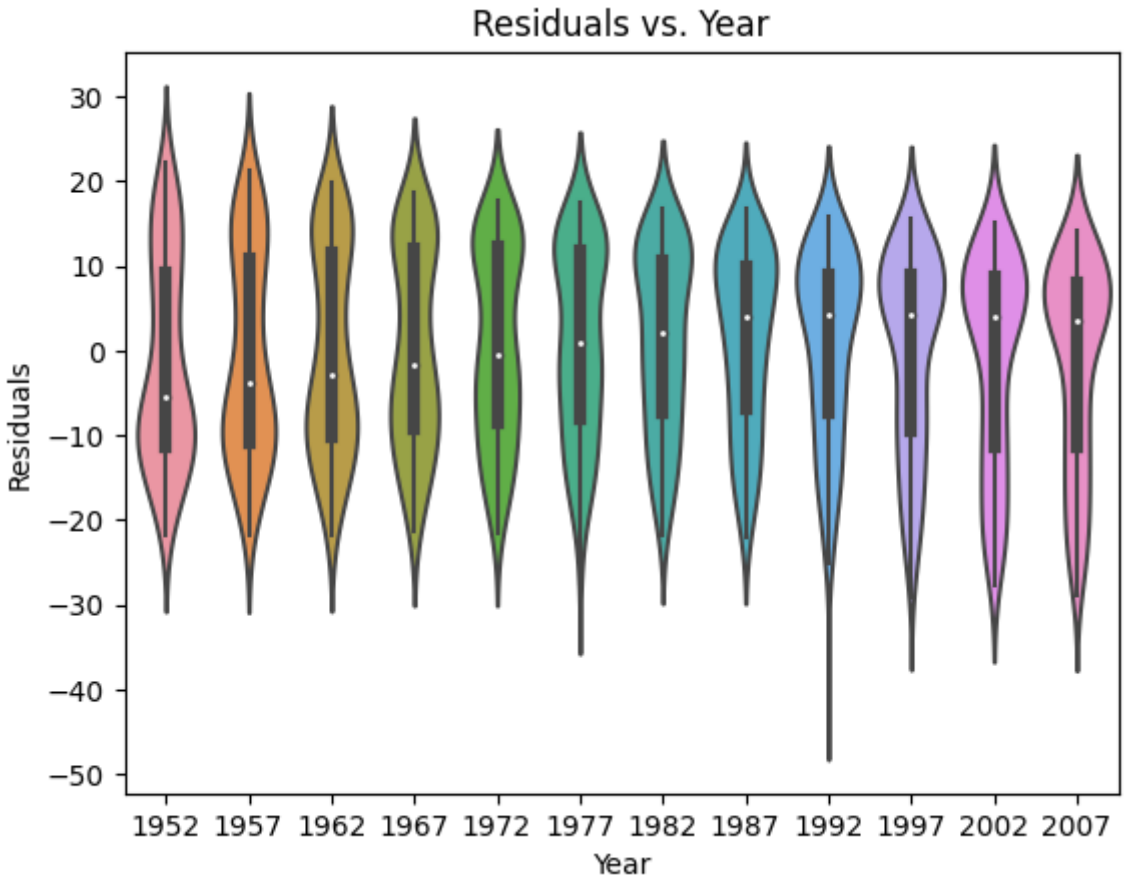
Exercise 3

In [7]:
```
# Obtain residuals
residuals = res1.resid

# Create DataFrame with residuals and year
df = pd.DataFrame({'residuals': residuals, 'year': data['year']})

# Plot
sns.violinplot(x='year', y='residuals', data=df)
plt.xlabel('Year')
plt.ylabel('Residuals')
plt.title('Residuals vs. Year')
```

Out[7]:
```
Text(0.5, 1.0, 'Residuals vs. Year')
```



Question 8

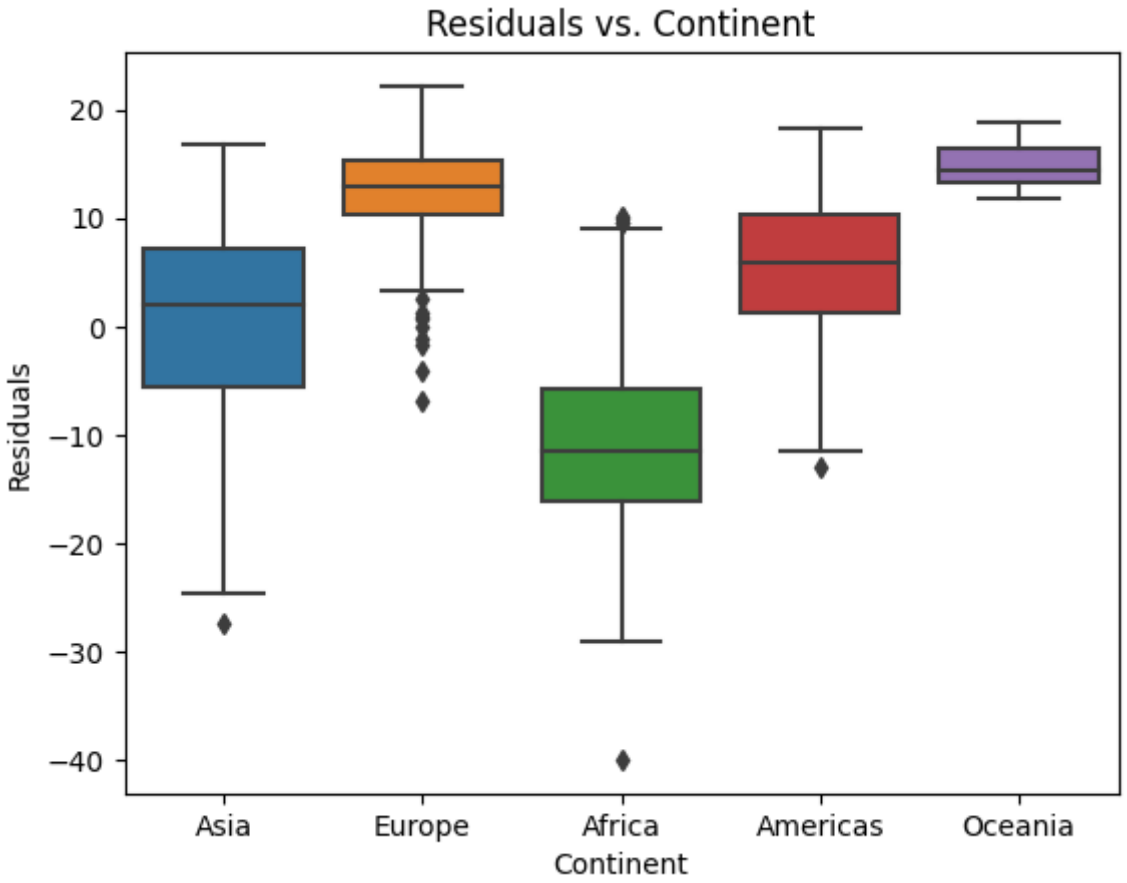
Yes, it matchs the expectancy I made. Still have violin shape for each year and the mean are around zero.

Exercise 4

In [8]:
```
# Create DataFrame with residuals and continent
df = pd.DataFrame({'residuals': residuals, 'continent': data['continent']})

# Plot
sns.boxplot(x='continent', y='residuals', data=df)
plt.xlabel('Continent')
plt.ylabel('Residuals')
plt.title('Residuals vs. Continent')
```

Out[8]:
```
Text(0.5, 1.0, 'Residuals vs. Continent')
```



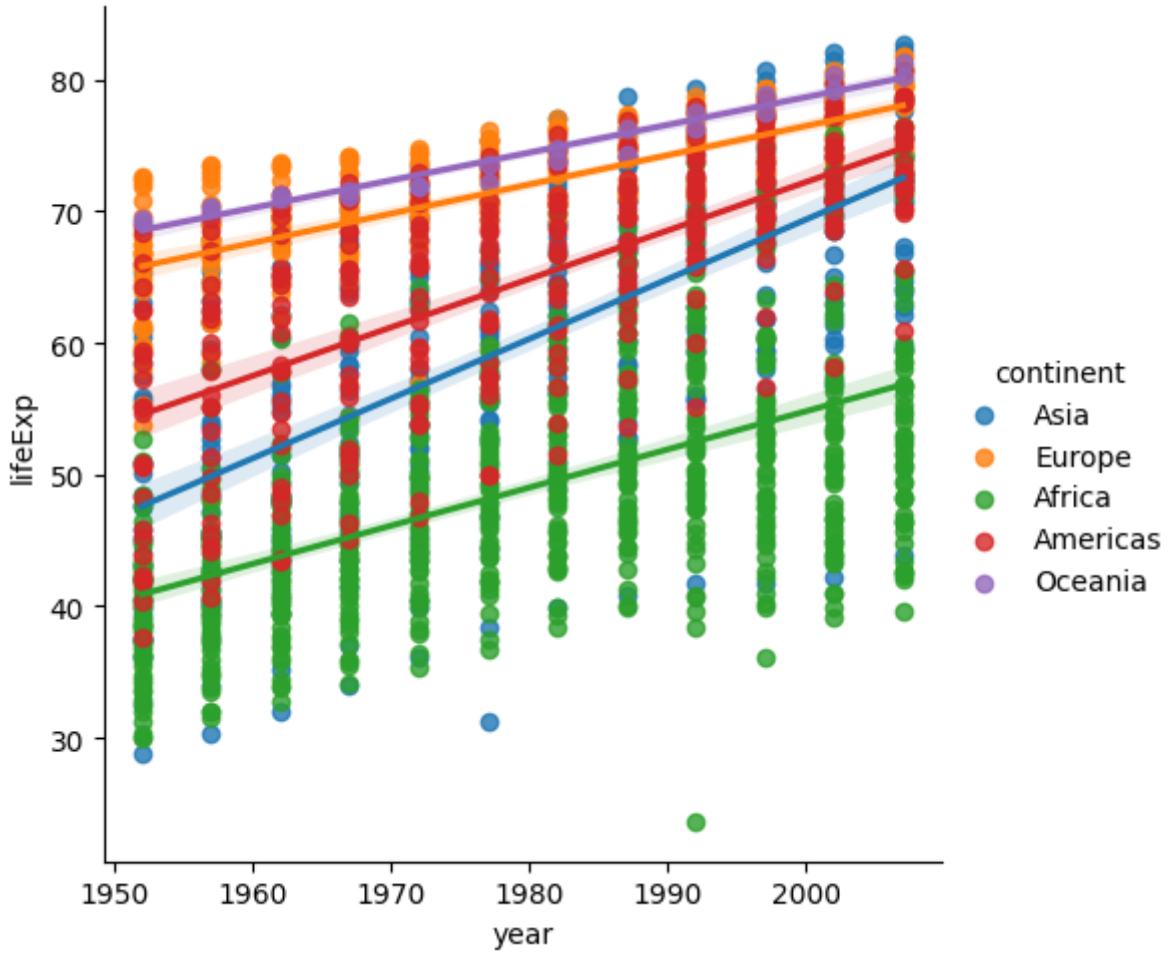
Question 9

Yes, there is a dependence between model residual and continent. As we can see from the plot, the mean and variance of residuais for each continent is different. We can performing separate regression analysis of life expectancy across time for each continent and do the comparsion to see any relatives.

Exercise 5

In [9]:
```
sns.lmplot(x='year', y='lifeExp', hue='continent', data=data)
```

Out[9]:
```
<seaborn.axisgrid.FacetGrid at 0x7fb19c50feb0>
```



Question 10

Yes, we should include an interaction term for continent and year to reduce the effect. As we can see from the plot, the regrssion line for each continent have huge difference, such as the gradient of slopes. So, they are making big effect on the regression model we just created.

Exercise 6

In [10]:
```
res2 = statsmodels.formula.api.ols('lifeExp ~ year*continent', data=data).fit()
print(res2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                lifeExp   R-squared:                       0.693
Model:                            OLS   Adj. R-squared:                  0.691
Method:                 Least Squares   F-statistic:                     424.3
Date:                Sat, 29 Apr 2023   Prob (F-statistic):               0.00
Time:                        03:25:04   Log-Likelihood:                -5771.9
No. Observations:                1704   AIC:                         1.156e+04
Df Residuals:                    1694   BIC:                         1.162e+04
Df Model:                           9
Covariance Type:            nonrobust
================================================================================
================
                                   coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------------------
----------------
Intercept                     -524.2578     32.963    -15.904      0.000    -58
8.911    -459.605
continent[T.Americas]         -138.8484     57.851     -2.400      0.016    -25
2.315     -25.382
continent[T.Asia]             -312.6330     52.904     -5.909      0.000    -41
6.396    -208.870
continent[T.Europe]            156.8469     54.498      2.878      0.004      4
9.957     263.737
continent[T.Oceania]           182.3499    171.283      1.065      0.287    -15
3.599     518.298
year                             0.2895      0.017     17.387      0.000
0.257       0.322
year:continent[T.Americas]       0.0781      0.029      2.673      0.008
0.021       0.135
year:continent[T.Asia]           0.1636      0.027      6.121      0.000
0.111       0.216
year:continent[T.Europe]        -0.0676      0.028     -2.455      0.014      -
0.122      -0.014
year:continent[T.Oceania]       -0.0793      0.087     -0.916      0.360      -
0.249       0.090
==============================================================================
Omnibus:                       27.121   Durbin-Watson:                   0.242
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               44.106
Skew:                          -0.121   Prob(JB):                     2.65e-10
Kurtosis:                       3.750   Cond. No.                     2.09e+06
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct
ly specified.
[2] The condition number is large, 2.09e+06. This might indicate that there ar
e
strong multicollinearity or other numerical problems.
```

Question 11

All parameters in the model are significantly different from zero except for continent Oceania, which have a value of 0.287, far larger than 0.05 (if we assume the significance level is 0.05).

Question 12

In [11]:
```
continent_increase = res2.params.loc['year:continent[T.Americas]':'year:contine
continent_increase.index = ['Americas', 'Asia', 'Europe', 'Oceania']
continent_increase
```

Out[11]:
```
Americas    0.078122
Asia        0.163593
Europe     -0.067597
Oceania    -0.079257
dtype: float64
```

We can know that the life expectancy for Americas increase 0.078122 years on average; the life expectancy for Asia increase 0.163593 years on average; the life expectancy for Europe decrease 0.067597 years on average; the life expectancy for Oceania decrease 0.079257 years on average.

Exercise 7

In [12]:
```
print("F test(value) for model (a):", res1.fvalue)
print("F test(value) for model (b):", res2.fvalue)
print("Result from model (b) better than result from model (a)?", res2.fvalue >
```

```
F test(value) for model (a): 398.6047457117622
F test(value) for model (b): 424.2729023400693
Result from model (b) better than result from model (a)? True
```

Question 13

In [13]:
```
print(res2.compare_f_test(res1))
```

```
(346.5535276625867, 0.0, 8.0)
```

The interactive model is significantly better than the year-only model since it has a higher F-statistic value. Also, from the above output, we can know that the p-value is 0.0, which means the probability of null hypothesis is true.

Exercise 8

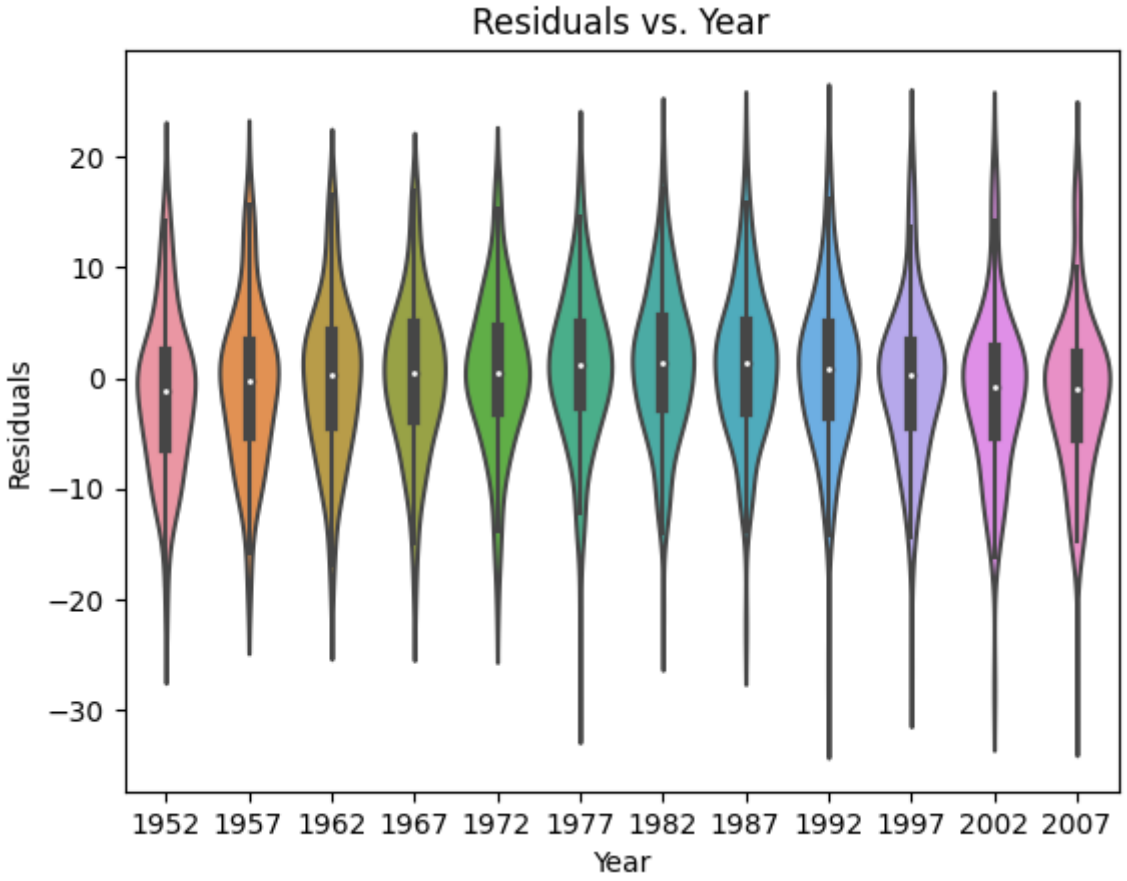
In [14]:
```
# Obtain residuals and fitted values
residuals = res2.resid
predicted = res2.predict()

# Create DataFrame with residuals and year
df = pd.DataFrame({'residuals': residuals, 'predicted': predicted, 'year': data

# Plot
sns.violinplot(x='year', y='residuals', data=df)
```

```
plt.xlabel('Year')
plt.ylabel('Residuals')
plt.title('Residuals vs. Year')
```

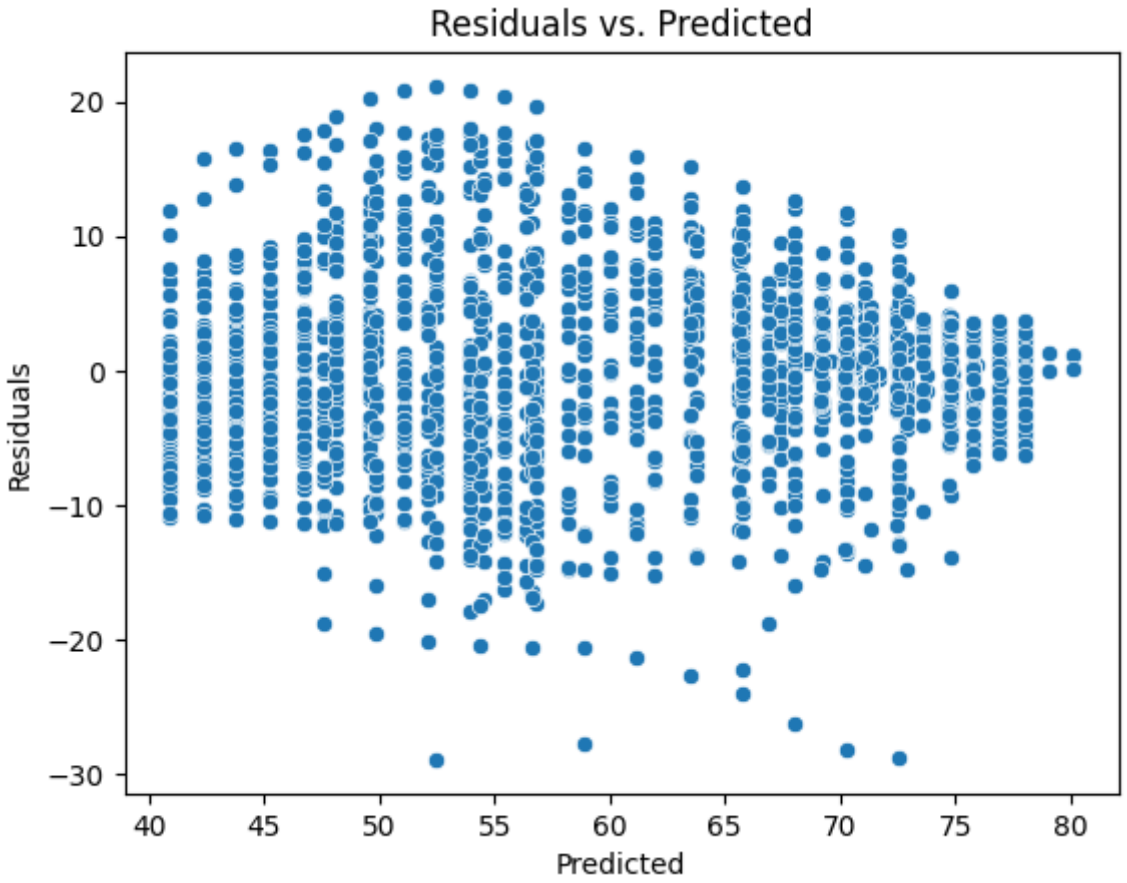
Out[14]:
```
Text(0.5, 1.0, 'Residuals vs. Year')
```



We can see that the residuals are symmetrically distributed around zero and have similar variance across years, which are assumptions of the linear regression model.

In [15]:
```
sns.scatterplot(x='predicted', y='residuals', data=df)
plt.xlabel('Predicted')
plt.ylabel('Residuals')
plt.title('Residuals vs. Predicted')
```

Out[15]:
```
Text(0.5, 1.0, 'Residuals vs. Predicted')
```



We can see that those points are randomly scattered around zero, so the linear regression model is appropriate for the data.

In [17]:
```
%%shell
jupyter nbconvert --to html /content/A3.ipynb
```

```
[NbConvertApp] Converting notebook /content/A3.ipynb to html
[NbConvertApp] Writing 1241974 bytes to /content/A3.html
```

Out[17]: