In [48]:
```
%matplotlib inline

from numpy import arange
from matplotlib import pyplot as plt
from scipy.stats import norm
import pandas as pd

plt.rcParams['figure.figsize'] = [16,7]
```

In [49]:
```
df = pd.read_csv("Life_Expectancy.csv")
df.head()
```

Out[49]:

	Year	Life_Expectancy
0	1950	35.812
1	1951	36.103
2	1952	36.687
3	1953	37.272
4	1954	37.859

In [50]:
```
df.dtypes
```

Out[50]:
```
Year                int64
Life_Expectancy    float64
dtype: object
```

In [55]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70 entries, 0 to 69
Data columns (total 2 columns):
Year               70 non-null int64
Life_Expectancy    70 non-null float64
dtypes: float64(1), int64(1)
memory usage: 1.2 KB
```

In [86]:
```
df.describe()
```

Out[86]:

	Year	Life_Expectancy
count	70.000000	70.000000
mean	1984.500000	54.477986
std	20.351085	10.224507
min	1950.000000	35.812000
25%	1967.250000	45.941250
50%	1984.500000	55.621000
75%	2001.750000	63.204750
max	2019.000000	69.656000

In [87]:
```
df['Life_Expectancy'].mean()
```

Out[87]:
```
54.47798571428573
```

In [56]:
```
df.skew()
```

Out[56]:
```
Year               0.000000
Life_Expectancy   -0.255111
dtype: float64
```

In [88]:
```
df.kurtosis()
```

Out[88]:
```
Year              -1.200000
Life_Expectancy   -1.141016
dtype: float64
```

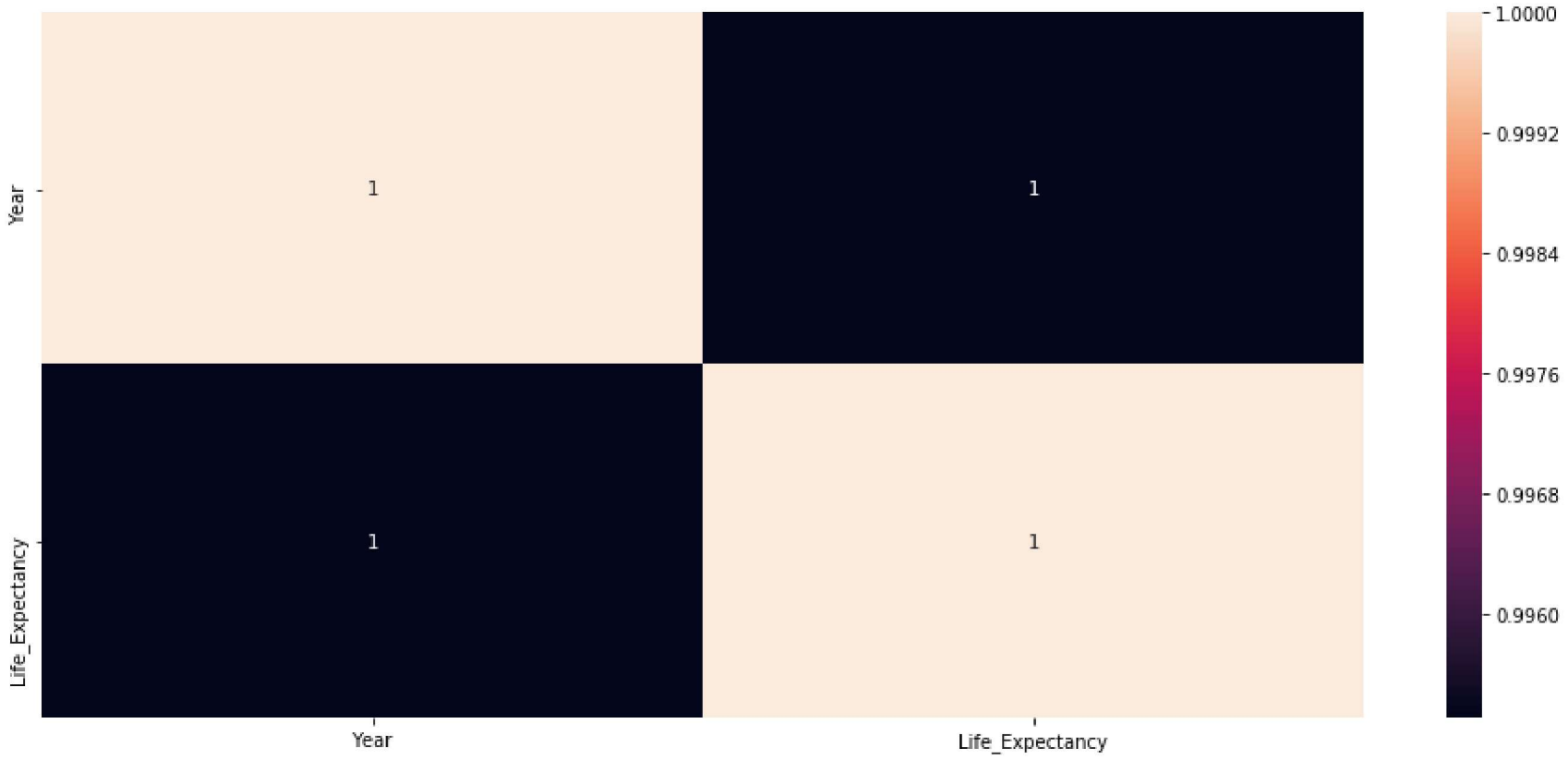
In [53]:
```
df.corr()
```

Out[53]:

	Year	Life_Expectancy
Year	1.000000	0.995317
Life_Expectancy	0.995317	1.000000

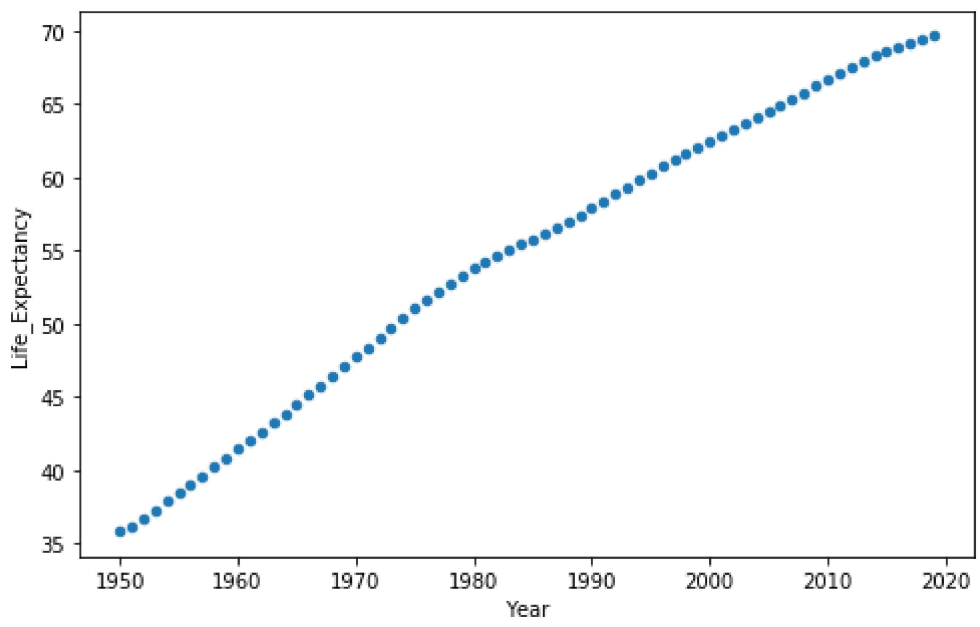
In [89]:
```
import seaborn as sns
sns.heatmap(df.corr(),annot=True)
```

Out[89]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e4e0e749e8>
```



In [54]:
```
df.plot.scatter('Year','Life_Expectancy', figsize=(8,5))
```

Out[54]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e4dffd3240>
```



In [68]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
X = df.iloc[:,0]
Y = df.iloc[:,1]
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=1/3.0,random_state=0)
X_train = X_train.values.reshape(-1,1)
```

In [69]:
```
from sklearn.linear_model import LinearRegression

regressor = LinearRegression()
regressor.fit(X_train,Y_train)
```

Out[69]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

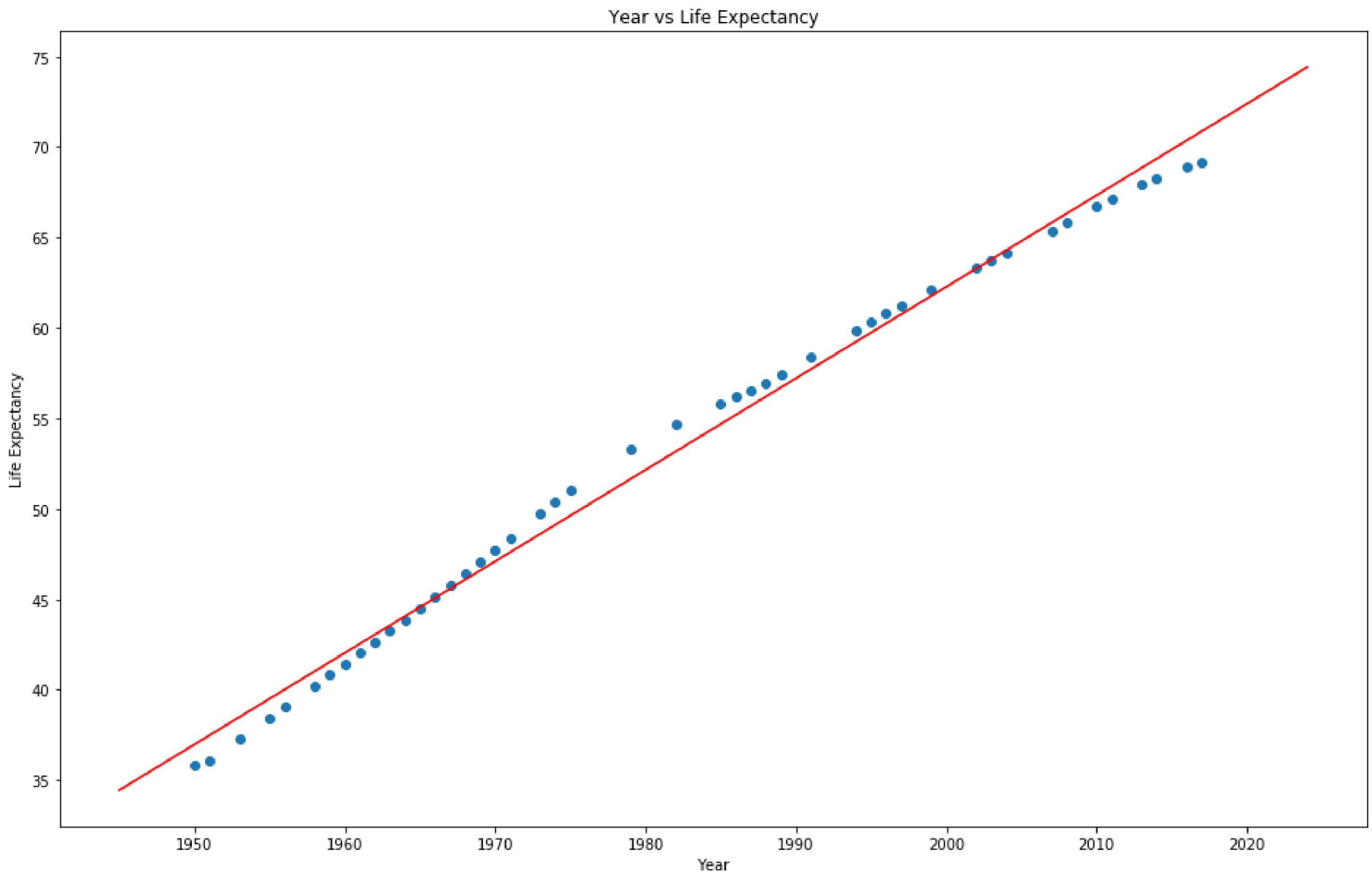
In [70]:
```
Y_pred = regressor.predict(X_test.values.reshape(-1,1))
Y_pred
```

Out[70]:
```
array([50.12754407, 50.63349153, 61.25838827, 48.10375421, 52.15133392,
       62.77623067, 40.51454225, 66.82381038, 54.17512378, 71.88328502,
       65.30596799, 51.139439  , 52.65728139, 58.22270349, 53.66917632,
       64.80002052, 71.37733755, 68.34165277, 58.72865096, 38.99669986,
       69.85949516, 62.2702832 , 37.98480493, 57.21080856])
```

In [71]:
```
Y_test
```

Out[71]:
```
26    51.630
27    52.222
48    61.669
22    49.061
30    53.814
51    62.907
7     39.630
59    66.244
34    55.441
69    69.656
56    64.918
28    52.786
31    54.268
42    58.851
33    55.074
55    64.500
68    69.416
62    67.545
43    59.349
4     37.859
65    68.607
50    62.505
2     36.687
40    57.865
Name: Life_Expectancy, dtype: float64
```

In [72]:
```
fig = plt.figure(figsize=(16, 10))
plt.scatter(X_train, Y_train)
plt.plot(np.arange(1945, 2025, 1), regressor.predict(np.arange(1945, 2025, 1).reshape(-1, 1)), color='red')
plt.title('Year vs Life Expectancy')
plt.xlabel('Year')
plt.ylabel('Life Expectancy')
plt.show()
```



In [73]:
```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
```

In [85]:
```
k = 5
kf = KFold(n_splits=k, random_state=None)

result = cross_val_score(regressor , X_train, Y_train, cv = kf)

print("Avg accuracy: {}".format(result.mean()))
```

```
Avg accuracy: 0.9914322103900443
```