In [1]:
```python
import numpy as np
import pandas as pd
import statsmodels.api as sm
import seaborn as sns
from scipy.stats import linregress
import matplotlib.pyplot as plt
```

## Loading set

In [ ]:
```python
df = pd.read_excel(r"C:\Users\224140745\Documents\CAr_Sales_2022_4.xlsx",
                   usecols=['Price', 'Car_Model', 'Year', 'Mileage','Year_Da
```

## Feature engineering

In [3]:
```python
#Age Normoilization
df['Age'] = df['Year_Data_Collected'] - df['Year']
# Data Preview
print('Size of data is ',len(df),'rows and ',len(df.columns), 'columns :','\
df.head()
```

```
Size of data is  33137 rows and  7 columns :
 ['Price', 'Car_Model', 'Year', 'Mileage', 'Transmission', 'Year_Data_Collec
ted', 'Age']
```

Out[3]:

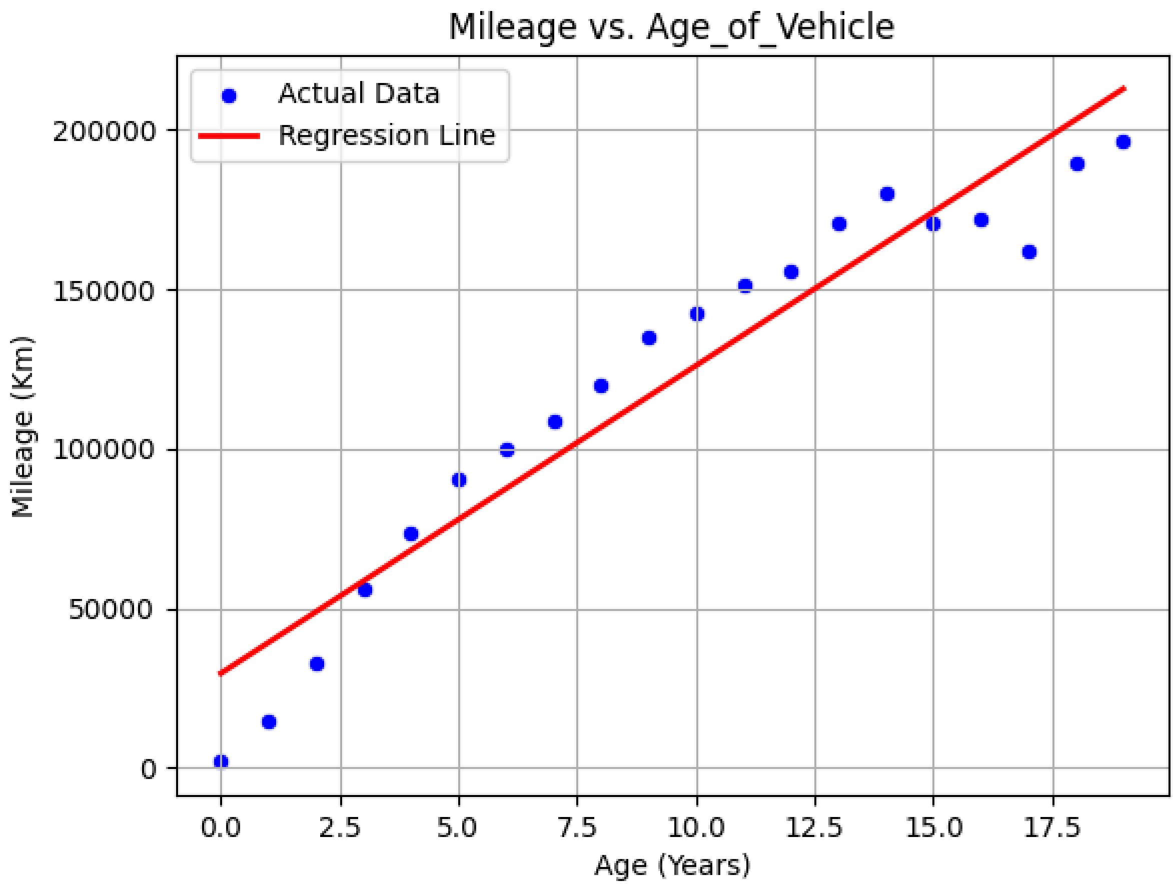
| | Price | Car_Model | Year | Mileage | Transmission | Year_Data_Collected | A |
|---|---|---|---|---|---|---|---|
| **0** | 1999900.0 | Jaguar F-Pace SVR | 2022 | 0.0 | Automatic | 2022 | |
| **1** | 1999900.0 | Jaguar F-Type R AWD Convertible | 2022 | 0.0 | Automatic | 2022 | |
| **2** | 1989276.0 | Jaguar F-Pace SVR | 2022 | 0.0 | Automatic | 2022 | |
| **3** | 1908634.0 | Land Rover Range Rover Sport HSE TDV6 | 2022 | 0.0 | Automatic | 2022 | |
| **4** | 1899995.0 | Audi Q8 55TFSI Quattro | 2022 | 0.0 | Automatic | 2022 | |

The dataset is an aggregated ecommerce sales of cars collected over a span of 3 years

## Mileage Regression

In [93]:
```python
# Extracting Age and Mileage for regression analysis
age_miles = df[df['Age']<20].groupby("Age")['Mileage'].aggregate(['mean','st
```

```python
# Regression analysis for Average Miles vs Age
X = age_miles['Age']
y = age_miles['Average_Miles']

# Add constant for intercept
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()

# Compute regression variables
result = linregress(age_miles['Age'], age_miles['Average_Miles'])
slope = result.slope
intercept = result.intercept
r_value = result.rvalue
p_value = result.pvalue
std_err = result.stderr

# Scatter plot of actual data & regression line
sns.scatterplot(age_miles, x = age_miles['Age'], y = age_miles['Average_Mile

sns.lineplot(age_miles,x = age_miles['Age'],y = (slope*age_miles['Age']+inte

# Labels and title
plt.xlabel("Age (Years)")
plt.ylabel("Mileage (Km)")
plt.title("Mileage vs. Age_of_Vehicle")
plt.legend()
plt.grid()
plt.show()

# Print regression results
print(f"Mileage Regression equation: y = {slope:.2f}x + {intercept:.2f}")
print(f"R² = {r_value**2:.3f}",'|', f"P-value = {p_value:.3f}",'|', f"Standa
print(model.summary())
```


Mileage vs. Age_of_Vehicle
Actual Data
Regression Line
Mileage (Km)
200000
150000
100000
50000
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Age (Years)

```
Mileage Regression equation: y = 9640.15x + 29549.09
R² = 0.921 | P-value = 0.000 | Standard Error = 666.029 | Intercept = 29549.
091 | Slope = 9640.152 |


                            OLS Regression Results
==============================================================================
==
Dep. Variable:          Average_Miles   R-squared:                       0.9
21
Model:                            OLS   Adj. R-squared:                  0.9
16
Method:                 Least Squares   F-statistic:                     20
9.5
Date:                Mon, 16 Jun 2025   Prob (F-statistic):           2.34e-
11
Time:                        16:54:31   Log-Likelihood:              -222.
35
No. Observations:                  20   AIC:                             44
8.7
Df Residuals:                      18   BIC:                             45
0.7
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
==
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
--
const       2.955e+04   7401.619      3.992      0.001     1.4e+04    4.51e+
04
Age         9640.1524    666.029     14.474      0.000    8240.877     1.1e+
04
==============================================================================
==
Omnibus:                        4.154   Durbin-Watson:                   0.3
00
Prob(Omnibus):                  0.125   Jarque-Bera (JB):                2.1
99
Skew:                          -0.556   Prob(JB):                        0.3
33
Kurtosis:                       1.816   Cond. No.                         2
1.5
==============================================================================
==

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is corre
ctly specified.
```

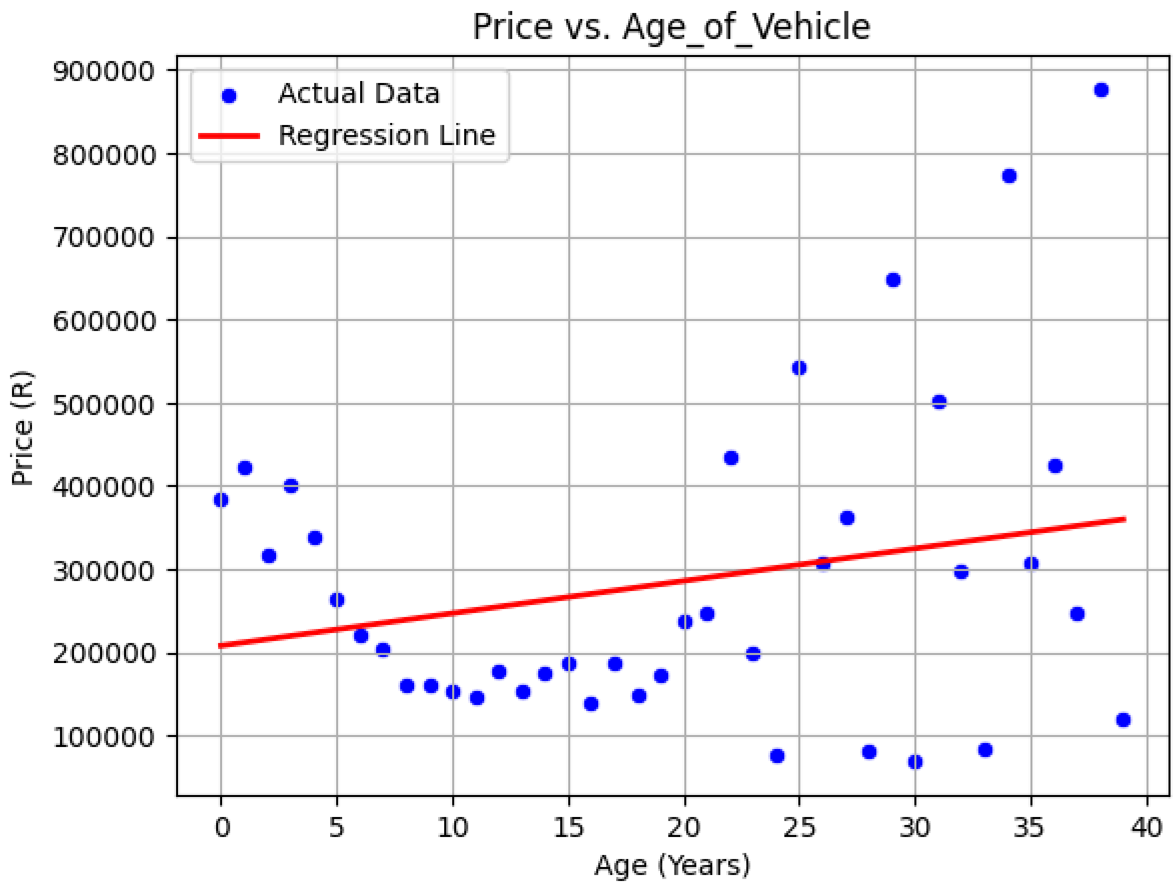
In [36]:
```
age_price.keys()
```

Out[36]:
```
Index(['Age', 'Average_Price', 'St_Dev_Price', 'Count'], dtype='object')
```

## Price Regression

In [ ]:

```
# Extracting Age and Price for regression analysis
age_price = df[df['Age']<40].groupby("Age")['Price'].aggregate(['mean','std'

# Regression analysis for Average Miles vs Age
X = age_price['Age']
y = age_price['Average_Price']

# Add constant for intercept
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()

# Compute regression variables
result = linregress(age_price['Age'], age_price['Average_Price'])
slope = result.slope
intercept = result.intercept
r_value = result.rvalue
p_value = result.pvalue
std_err = result.stderr

# Scatter plot of actual data & regression line
sns.scatterplot(age_price, x = age_price['Age'], y = age_price['Average_Pric

sns.lineplot(age_price,x = age_price['Age'],y = (slope*age_price['Age']+inte

# Labels and title
plt.xlabel("Age (Years)")
plt.ylabel("Price (R)")
plt.title("Price vs. Age_of_Vehicle")
plt.legend()
plt.grid()
plt.show()

# Print regression results
print(f"Price Regression Eqaution: y = {slope:.2f}x + {intercept:.2f}")
print(f"R² = {r_value**2:.3f}",'|', f"P-value = {p_value:.3f}",'|', f"Standa
print(model.summary())
```


Price vs. Age_of_Vehicle
Actual Data
Regression Line
Price (R)
900000
800000
700000
600000
500000
400000
300000
200000
100000
0
5
10
15
20
25
30
35
40
Age (Years)

```
Price Regression Eqaution: y = 3888.22x + 208212.32
R² = 0.061 | P-value = 0.126 | Standard Error = 2482.859 | Intercept = 20821
2.325 | Slope = 3888.223 |


                            OLS Regression Results
==============================================================================
==
Dep. Variable:          Average_Price   R-squared:                       0.0
61
Model:                            OLS   Adj. R-squared:                  0.0
36
Method:                 Least Squares   F-statistic:                     2.4
52
Date:                Mon, 16 Jun 2025   Prob (F-statistic):              0.1
26
Time:                        16:50:44   Log-Likelihood:                -540.
04
No. Observations:                  40   AIC:                             108
4.
Df Residuals:                      38   BIC:                             108
7.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
==
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
--
const       2.082e+05   5.63e+04      3.701      0.001    9.43e+04    3.22e+
05
Age         3888.2227   2482.859      1.566      0.126   -1138.062    8914.5
08
==============================================================================
==
Omnibus:                        8.604   Durbin-Watson:                   2.5
12
Prob(Omnibus):                  0.014   Jarque-Bera (JB):                7.5
00
Skew:                           0.986   Prob(JB):                       0.02
35
Kurtosis:                       3.783   Cond. No.                          4
4.5
==============================================================================
==

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is corre
ctly specified.
```

## References

https://ploomber.io/blog/jupyter-notebook-convert/

https://www.statology.org/how-to-perform-simple-linear-regression-with-statsmodels/

https://ukzn.ci.hr/applicant/index.php?controller=Listings&method=view&listingid=4e1bd2bb-8e74-4c29-9408-bc5fe448c11d