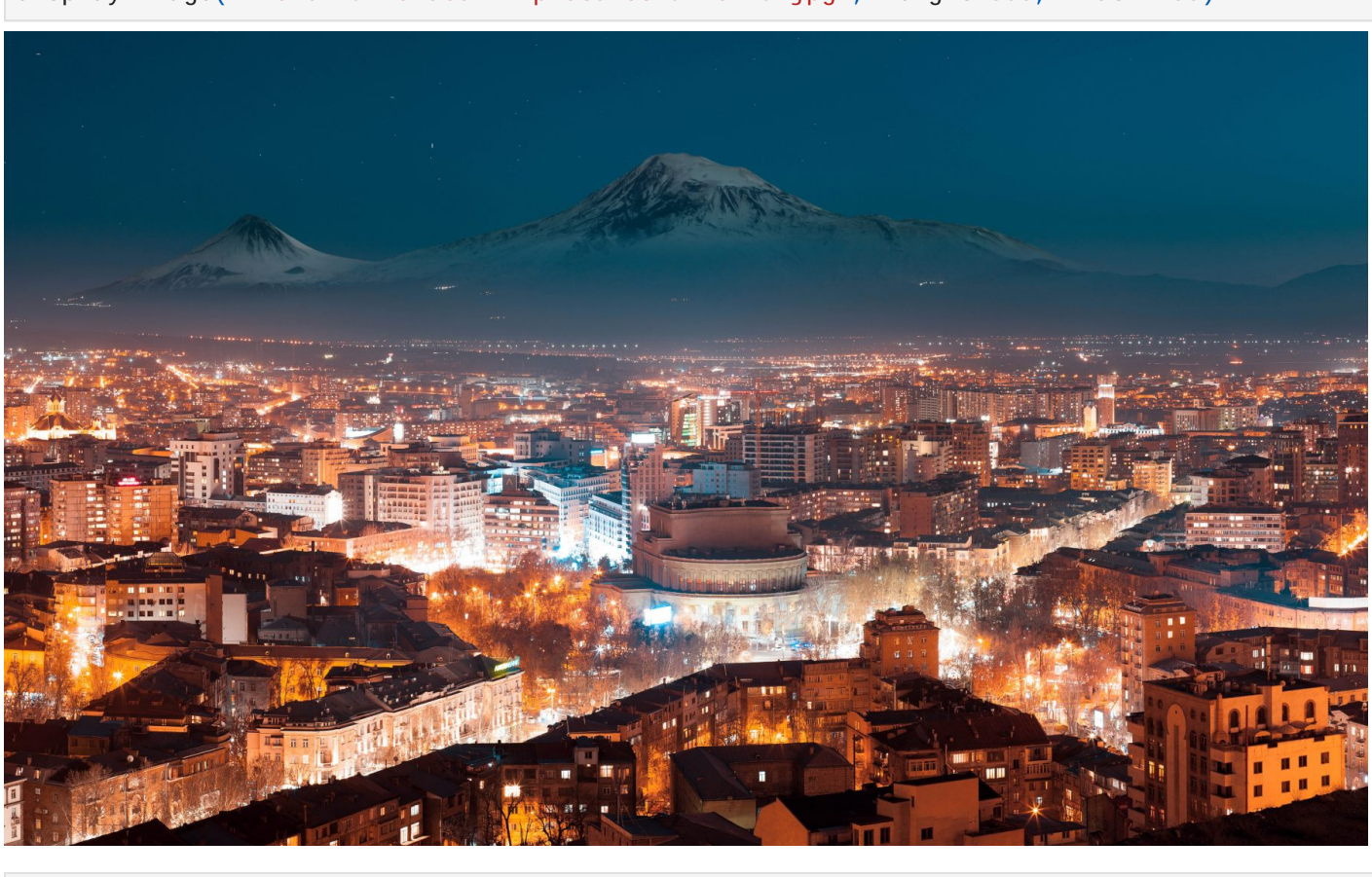
Armenian Market Car Price Prediction. I obtained this dataset from www.kaggle.com

In [17]:
```python
#Car Name: The name or model of the car.

#Year: The manufacturing year of the car.

#Region: The geographical area where the car is being sold or was registered

#FuelType: The type of fuel the car uses.

#Mileage: The total distance the car has traveled, typically measured in miles or kilomete

#Price: The selling price of the car, usually in the local currency.
```

In [1]:
```python
!date
```
```
Mon 15 Jul 2024 05:49:04 PM EDT
```

In [3]:
```python
from IPython import display
display.Image('/home/harlohutch77/pictures/armenia.jpg', height=300, width=700)
```

Out[3]:



In [4]:
```python
#importing all libraries needed for this prject to work properly
```

In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import cufflinks as cf
%matplotlib inline
```

In [2]:
```python
#using pandas to read the csv file
```

In [3]:
```python
arm = pd.read_csv('/home/harlohutch77/python/python_master/Armenian Market Car Prices.csv')
arm.info()
```
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7995 entries, 0 to 7994
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Car Name  7995 non-null   object
 1   Year      7995 non-null   int64 
 2   Region    7995 non-null   object
 3   FuelType  7995 non-null   object
 4   Mileage   7995 non-null   int64 
 5   Price     7995 non-null   int64 
dtypes: int64(3), object(3)
memory usage: 374.9+ KB
```

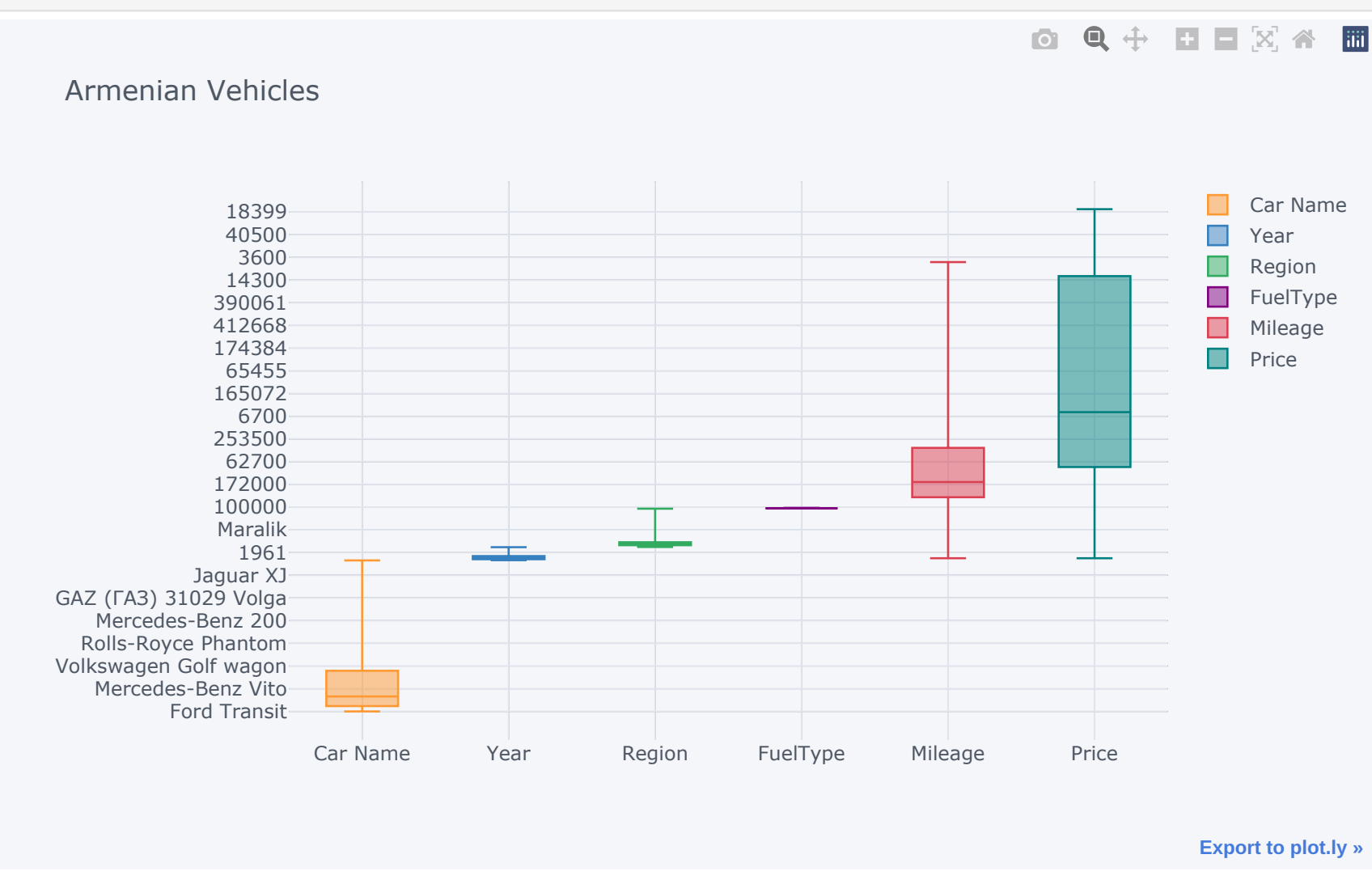
In [4]:
```python
arm.describe()
```

Out[4]:

	Year	Mileage	Price
count	7995.000000	7995.000000	7995.000000
mean	2010.342089	146401.523327	15555.002877
std	9.113848	108371.306576	17731.598604
min	1943.000000	10.000000	200.000000
25%	2004.000000	60000.000000	6000.000000
50%	2011.000000	130000.000000	11500.000000
75%	2018.000000	210000.000000	19000.000000
max	2024.000000	1000000.000000	565000.000000

In [5]:
```python
from plotly import __version__
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot

print(__version__) # requires version >= 1.9.0

import cufflinks as cf

cf.go_offline()
```
```
5.7.0
```

In [6]:
```python
arm.iplot(kind='box', title='Armenian Vehicles')
```



In [18]:
```python
#There are 7995 vehicles sold from 1943 to 2024
print(arm['FuelType'])
```
```
0         Diesel
1       Gasoline
2       Gasoline
3       Gasoline
4       Gasoline
          ...   
7990    Gasoline
7991    Gasoline
7992    Gasoline
7993    Gasoline
7994    Gasoline
Name: FuelType, Length: 7995, dtype: object
```

In [22]:
```python
#226 deisel vehicles out of 7995
arm[arm['FuelType']=='Diesel']['Car Name']
```

Out[22]:
```
0                 Ford Transit
38      Mercedes-Benz Sprinter
43                Ford Transit
47                Ford Transit
92      Mercedes-Benz Sprinter
                 ...          
7760              JAC T8 pickup
7844              Ford Transit
7920              Ford Transit
7922              Ford Transit
7987        Mercedes-Benz Viano
Name: Car Name, Length: 226, dtype: object
```

In [24]:
```python
#Percentage of deisel vehicles sold
print(226/7995)
```
```
0.028267667292057538
```

In [14]:
```python
#7326 gasoline vehicles out of the 7995
arm[arm['FuelType']=='Gasoline']['Car Name']
```

Out[14]:
```
1                  Ford Focus
2                 Opel Vectra
3                     Mazda 6
4                 Opel Vectra
5                     Mazda 6
                 ...         
7990    VAZ (Lada) 2121 (4x4)
7991           Toyota 4Runner
7992     Opel Astra hatchback
7993             Toyota Camry
7994    Mercedes-Benz C-Class
Name: Car Name, Length: 7326, dtype: object
```

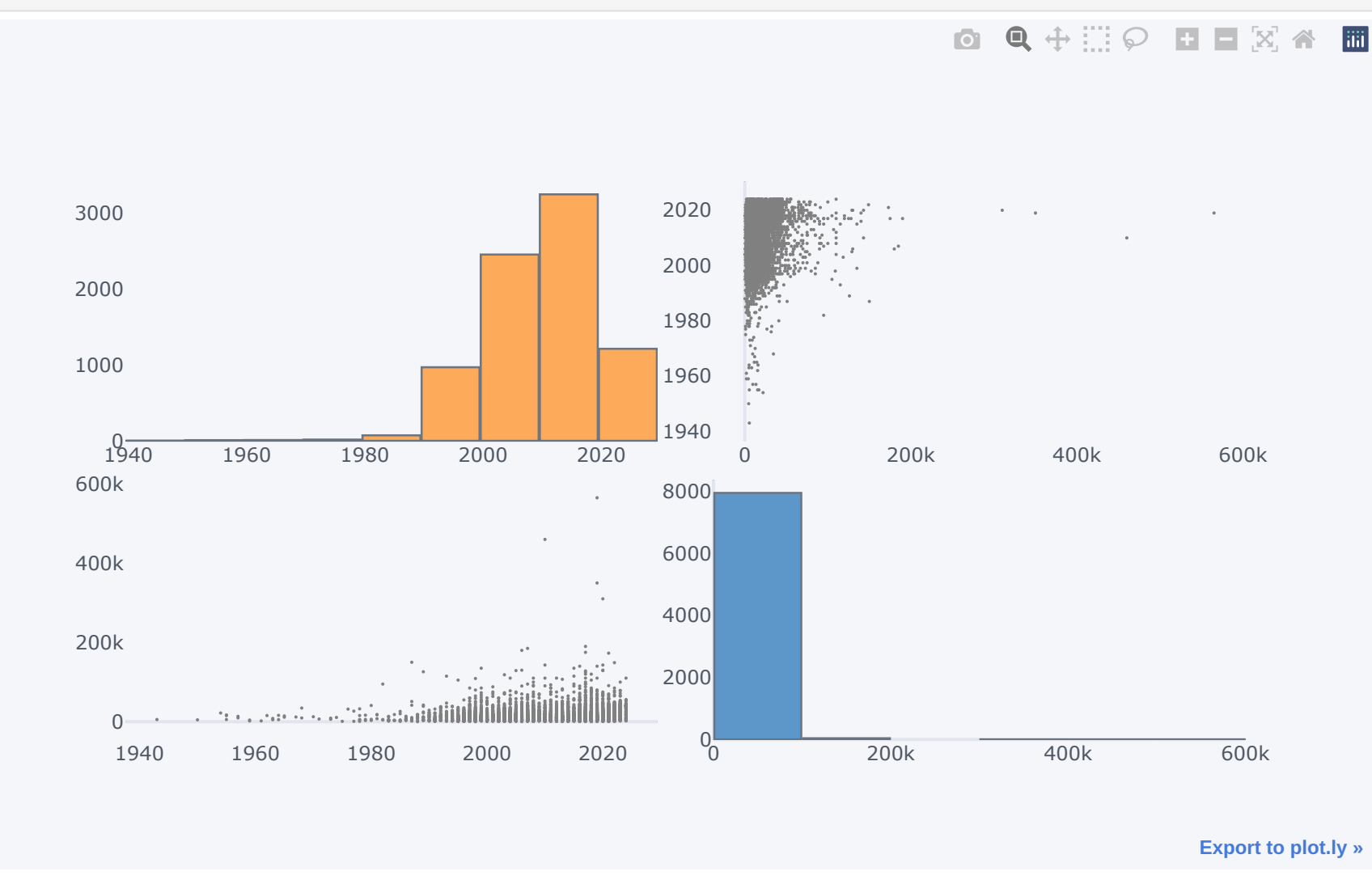
In [26]:
```python
#Percentage of gasoline vehicles sold
print(7326/7995)
```
```
0.9163227016885553
```

In [16]:
```python
#229 electric vehicles out of the 7995
arm[arm['FuelType']=='Electric']['Car Name']
```

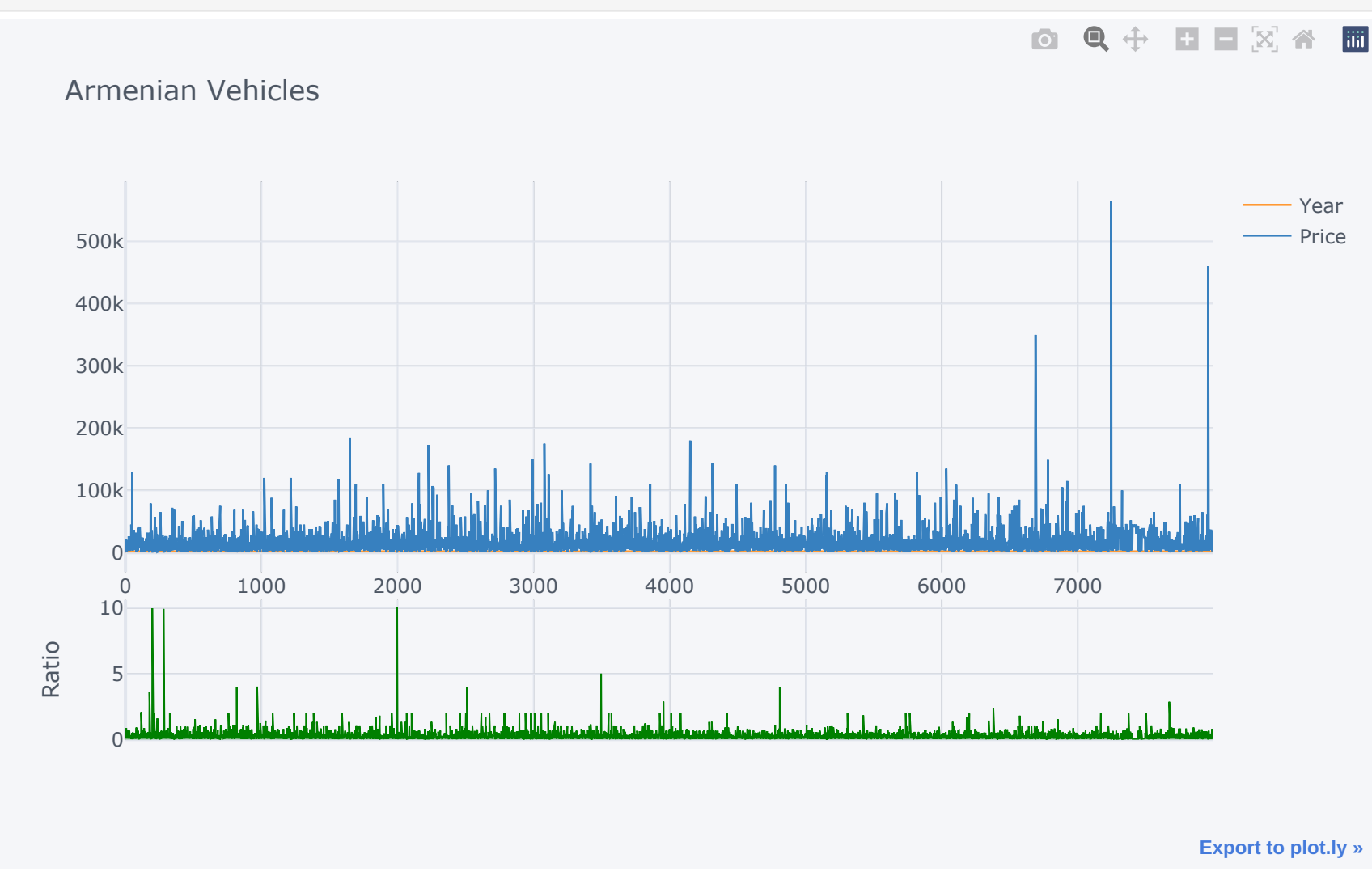
Out[16]:
```
182            Tesla Model Y
183            Tesla Model S
192            Tesla Model S
228            Tesla Model 3
345            Tesla Model 3
                 ...       
7546    Mercedes-Benz EQE AMG
7547        Mercedes-Benz EQE
7577            Tesla Model X
7684    Nissan Leaf hatchback
7798    Nissan Leaf hatchback
Name: Car Name, Length: 229, dtype: object
```

In [27]:
```python
#Percentage of electric vehicles sold
print(229/7995)
```
```
0.02864290181363352
```

In [7]:
```python
arm[["Year","Price"]].scatter_matrix()
```



In [8]:
```python
arm[['Year','Price']].iplot(kind="ratio", title='Armenian Vehicles')
```



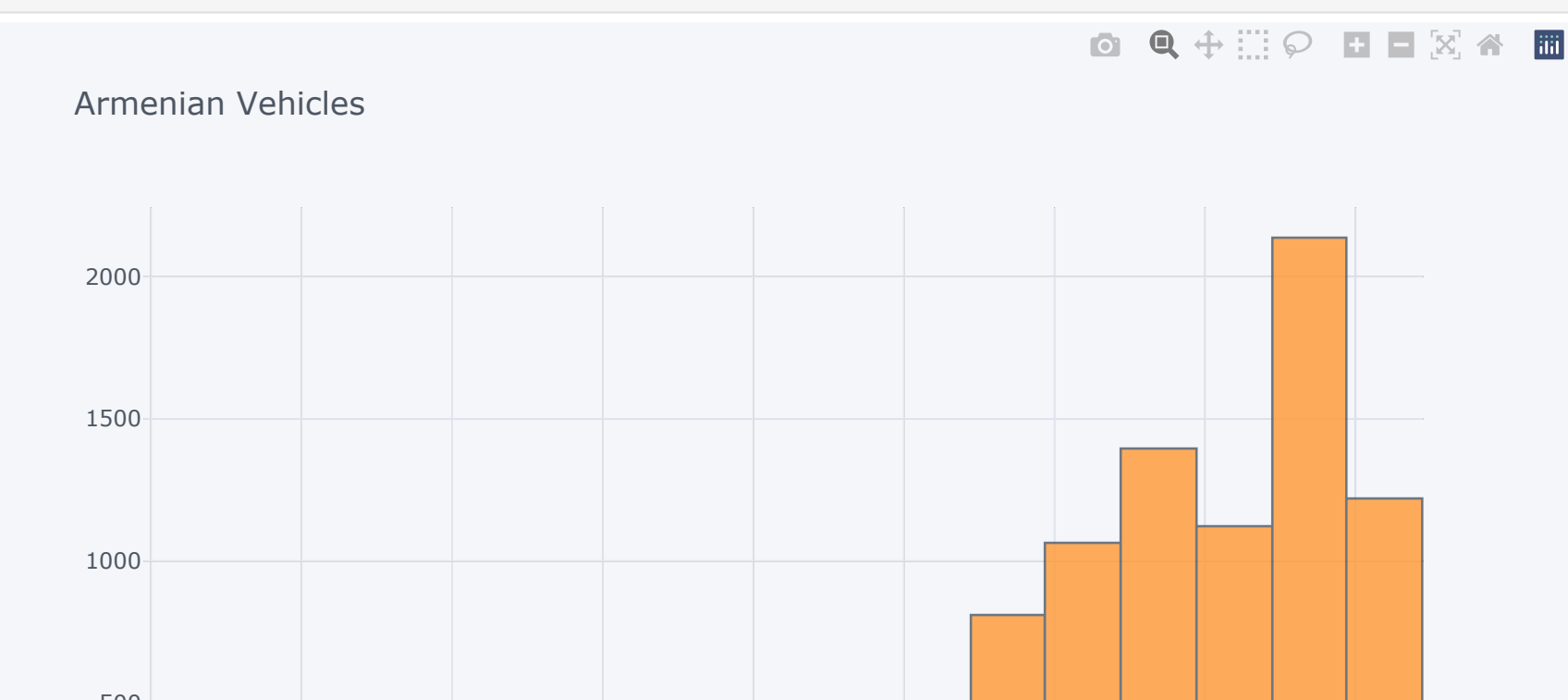
In [41]:
```python
arm[arm['Year']==1950]['Price'].count()
```

Out[41]:
```
1
```

In [42]:
```python
arm[arm['Year']==2024]['Price'].count()
```

Out[42]:
```
92
```

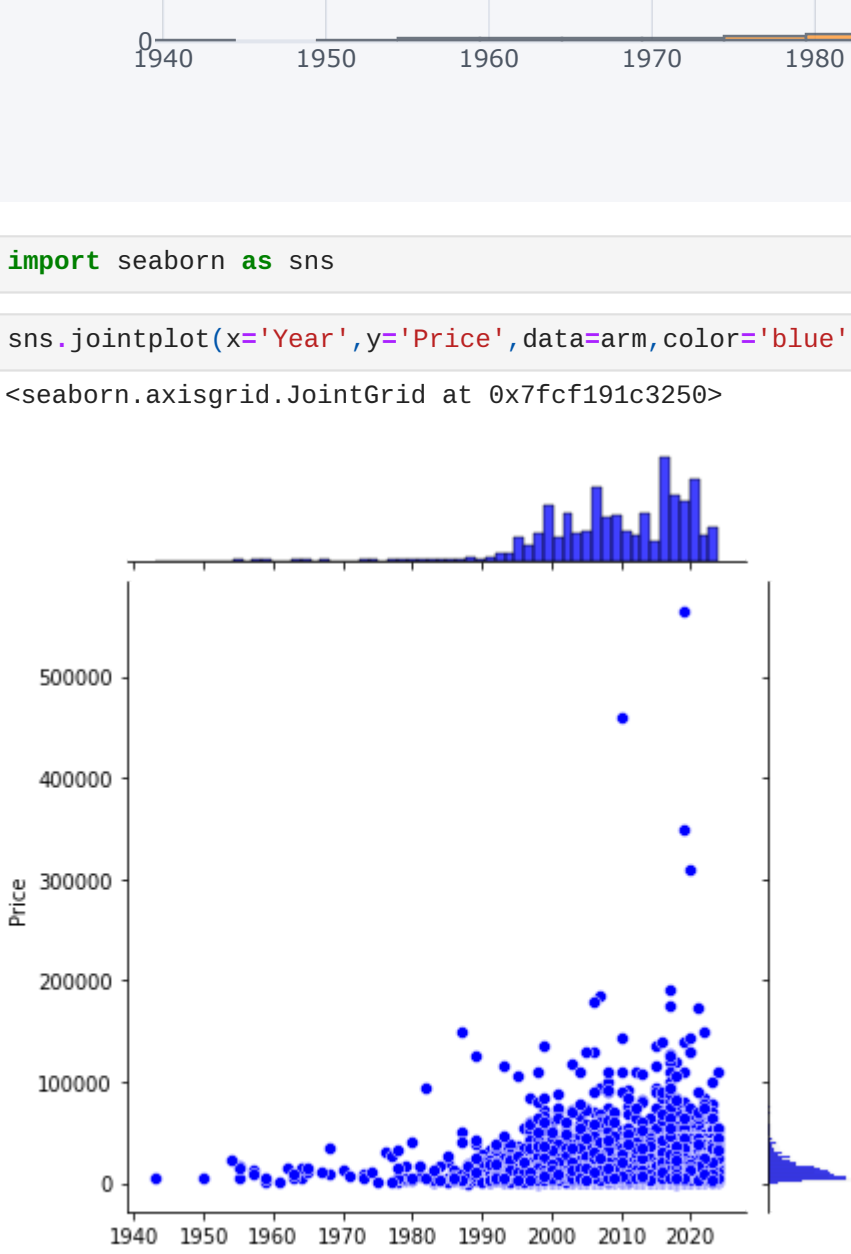
In [10]:
```python
arm['Year'].iplot(kind='hist',title='Armenian Vehicles',bins=35)
#df['A'].iplot(kind='hist', bins=35)
```



In [11]:
```python
import seaborn as sns
```

In [13]:
```python
sns.jointplot(x='Year',y='Price',data=arm,color='blue')
```

Out[13]:
```
<seaborn.axisgrid.JointGrid at 0x7fcf191c3250>
```



In [21]:
```python
#train_test_split linear regression model
```

In [22]:
```python
X = arm[['Year','Price']]
y = arm['Mileage']
```

In [23]:
```python
from sklearn.model_selection import train_test_split
```

In [24]:
```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=101)
```

In [25]:
```python
from sklearn.linear_model import LinearRegression
```

In [26]:
```python
lm = LinearRegression()
```

In [27]:
```python
lm.fit(X_train,y_train)
```

Out[27]:
```
LinearRegression()
```

In [47]:
```python
#model evalutation
```

In [28]:
```python
# print the intercept
print(lm.intercept_)
```
```
15618433.178325195
```

In [29]:
```python
coeff_arm = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
coeff_arm
```

Out[29]:

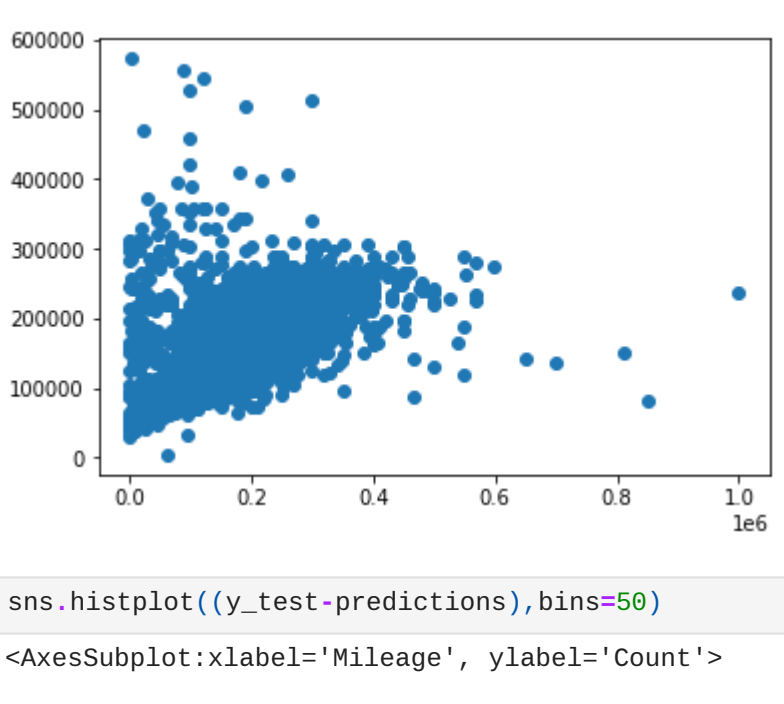
	Coefficient
Year	-7694.870663
Price	-0.138003

In [50]:
```python
#predictions
```

In [30]:
```python
predictions = lm.predict(X_test)
```

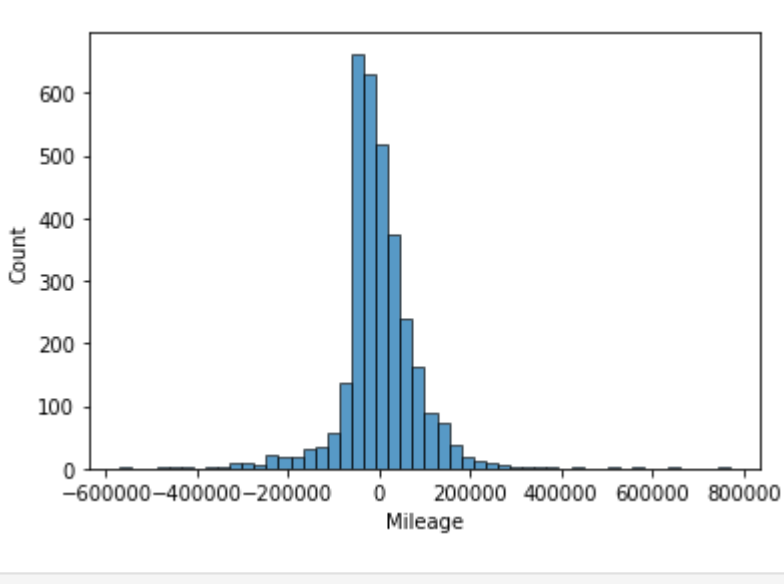
In [31]:
```python
plt.scatter(y_test,predictions)
```

Out[31]:
```
<matplotlib.collections.PathCollection at 0x7f8496d2a880>
```



In [32]:
```python
sns.histplot((y_test-predictions),bins=50)
```

Out[32]:
```
<AxesSubplot:xlabel='Mileage', ylabel='Count'>
```



In [33]:
```python
#Regression Evaluation Metrix
```

In [34]:
```python
from sklearn import metrics
```

In [35]:
```python
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```
```
MAE: 55760.82569821228
MSE: 6957955504.647916
RMSE: 83414.36030233593
```

In [36]:
```python
from IPython import display
display.Image('/home/harlohutch77/pictures/armenia.jpg', height=300, width=700)
```

Out[36]: